Nonlinear Kalman Filter architecture for integrated GPS and accelerometer based vehicle navigation

Andrew Soundy, Daniel Schumayer, Timothy Molteno
Department of Physics
University of Otago
tim@elec.ac.nz

***Abstract*—By combining GPS and accelerometer measurements with a nonlinear Kalman filter we provide a method to infer vehicle dynamics including position, velocity, acceleration and heading. This will allow inference of driver behaviour such as sharp accelerations or aggressive cornering. We also discuss the independent Gaussian noise model and propose a new noise model for GPS measurements.**

I. Introduction

GPS represents a cheap, accurate tool for determining position with an accuracy as low as 3m (standalone unit) [1]. GPS units also output velocity mesurements which typically have an accuracy of better than $1ms^{-1}$ [2]. However most GPS devices only receive data at a rate of 1Hz, although it is possible to reconfigure the device to sample at a rate of 5Hz or (with some difficulty) 10Hz [3]. Also the short-term noise for a standalone GPS unit is significant and GPS signals can be subject to obstruction [1].

By contrast accelerometers can easily have a much higher sampling frequency, in the order of hundreds of Hz [1]. Inertial navigation also has a low short-term noise. Though for a purely inertial based navigation system errors accumulate over time, leading to poor long-term position estimates unless using expensive, highly specialised equipment [1][3]. Furthermore, by directly measuring the acceleration of a vehicle we can measure any change in the heading of the vehicle and update our state predictions accordingly. Conversely, if only GPS data is measured and the vehicle turns a corner or crests a hill then the change in heading can only be inferred by later measurements. Combining GPS and inertial navigation units is therefore a logical option and is currently being used regularly for navigation in ships, aircraft and UAV's. For road-based vehicles it is more common to use dead reckoning techniques by manually tapping into the odometer and the wheel angle [1][4]. However this requires manually modifying the vehicle and will not provide attitude estimation. In this paper we lay out an algorithm for combining GPS and accelerometer measurements via a nonlinar Kalman filter which does not require any modification of hardware or interaction with the software of the vehicle and provides superior position and velocity estimates than a standalone GPS unit.

II. Literature Review

Filtering noisy measurements to determine an underlying state is a well-researched problem which has several solutions which depend on the exact requirements of the system. The Kalman filter pioneered by Rudolf E. Kalman and Richard S. Bucy is a recursive, linear filter which provides optimal estimates for any linear system with Gaussian noise [5]. Being a recursive algorithm the Kalman filter is computationally very efficient and so suitable for real-time applications such as navigation [6]. However the standard Kalman filter is only accurate for linear systems and so to make use of the Kalman filter for the nonlinear problem of vehicle navigation a nonlinear extension to the Kalman filter must be used[4].

The two main nonlinear extensions to the Kalman filter are the extended Kalman filter (EKF) and the unscented Kalman filter (UKF) [11]. The EKF invovles linearising the nonlinear state dynamics via a first order Taylor expansion of the state dynamics around the current state estimate. Once the problem is reduced to a pseudo-linear problem the standard Kalman filter algorithm is then used. The UKF involves choosing a deterministic set of points, so-called sigma points, which have the same mean and covariance as the current state estimate. The UKF then propagates each of the sigma points directly through the nonlinear state dynamics with no approximations. Once this is done the mean and covariance of the propagated particles is taken to be the predicted state estimate and state covariance.

Of the two main nonlinear Kalman filters there is little difference in computational requirements. The EKF is easier and quicker to implement and both filters perform similarly provided that the nonlinearities of the system are sufficiently small [7]. However in highly nonlinear systems the UKF has been shown to be superior to the EKF [11].

While the Kalman filter is an impressive algorithm the particle filter is fully capable of handling nonlinear system dynamics as well as non-Gaussian probability density functions [8]. The particle filter is a recursive, sequential Monte Carlo method which numerically approximates the posterior distribution by directly implementing the Bayesian recursion equations [9]. Similarly to the UKF the particle filter represents the current state estimate as a distribution of particles and propagates them directly through the nonlinear system dynamics. The difference is in the selection of particles, the UKF deterministically chooses it's sigma points so that they have the same mean and covariance as the current state estimate, therefore allowing a Gaussian to be fit to their distribution. The particle filter randomly chooses the particles based on their calculated probabilities. This allows the particle filter to approximate arbitrary probability density functions, not just Gaussians. The main numerical difference is the number of particles, for a one dimensional problem the UKF will choose three sigma-points per iteration. The accuracy of the particle filter is determined by the number of 'particles' that it propagates through the system dynamics. The particle filter can, in principle use any number of particles, however the minimum tends to be in the order of a hundred with common numbers being hundreds or thousands of particles, depending on the specific requirements of the system. As the reader can imagine this approach, while more general, has a far higher compuational time than the EKF [9]. Therefore the particle filter is better suited to post-processing roles rather than real-time state estimation. The Kalman filter, meanwhile, is ideally suited for real-time applications such as navigation [6].

For our purposes it could be possible to implement a Kalman filter operating in real-time while the vehicle is operating, then once the experiment is over the same measurement data could be fed to a particle filter with a large number of particles and the performance of the Kalman filter compared against the particle filter as a validation of how accurate the Kalman filter is.

III. The Kalman Filter

The Kalman filter is a common algorthim for sequential inference that provides optimal estimates for the current state and covariance given a linear system with Gaussian noise [10]. The Kalman filter assumes that there is some underlying 'true' state $\mathbf{x}(k)$, of which imperfect measurements $\mathbf{z}(k)$ are made. The filter then uses known parameters of the system to form a current state estimate $\hat{\mathbf{x}}(k|k)$ given the measurements. The Kalman filter also provides an estimate of the covariance of the state estimate $P(k|k)$.

In essence the Kalman filter consists of two steps, prediction and measurement. The current state estimate $\hat{\mathbf{x}}(k|k)$ is propagated forward in time by the state propagation matrix $F(k)$ to give a prediction of the state. The next measurement of the state $\mathbf{z}(k+1)$ is then taken and the prediction and measurement combined, weighted by the Kalman gain $K(k)$ which is a factor used to update the prior distribution. For linear systems it is assumed that the current true state is a linear combination of the previous state and some Gaussian system noise $\mathbf{w}(k)$ with covariance mattrix $Q(k)$.

Prediction:

$$\begin{aligned} \hat{\mathbf{x}}(k|k-1) &= F(k)\hat{\mathbf{x}}(k-1|k-1), \\ P(k|k-1) &= F(k)P(k-1|k-1)F(k)^T + Q(k-1), \end{aligned} \quad (1)$$

Measurement:

$$\begin{aligned} \mathbf{z}(k) &= H(k)\mathbf{x}(k) + \mathbf{v}(k), \\ K(k) &= P(k|k-1)H(k)^T\Big(H(k)P(k|k-1)H(k)^T + R(k)\Big)^{-1}, \\ \hat{\mathbf{x}}(k|k) &= \hat{\mathbf{x}}(k|k-1) + K(k)\Big(\mathbf{z}(k) - H(k)\hat{\mathbf{x}}(k|k-1)\Big), \end{aligned} \quad (2)$$

The measurement matrix $H(k)$ is used to project the current state onto the measurement space. It is assumed that due to the measurement process measurements of the state will have additional Gaussian noise $\mathbf{v}(k)$ with covariance $R(k)$. Combining the measurement noise and projection of the state then provides the measurement $\mathbf{z}(k)$.

As you can see from equation (1), implicit in the formulation of the Kalman filter is the idea that you can propagate the state forwards in time by matrix multiplication and adding noise. The use of matrix multiplication in the Kalman filter greatly eases propagating the state covariances, however it does mean that further steps must be taken to handle systems with nonlinear dynamics.

The Kalman gain of the system depends on the current state covariance $P(k|k)$ as well as the covarinace of the system noise $Q(k)$ and the covariance of the measurement noise $R(k)$. This automatically determines the weighting given to the predictions and measurements. If the measurement covariance is larger than the system covariance then the Kalman gain places a stronger emphasis on the predictions than the measurements and vice versa [5].

For a GPS based navigation system the data returned by GPS units provides some ability to adjust measurement covariances in real time. In addition to position and velocity information most devices can be configured to return the number of satellites used as well as HDOP and VDOP information, Horizontal and Vertical Dilution of Precision, respectively. The HDOP and VDOP data give an indication of the error in the horizontal and vertical directions, caused by the geometric arrangement of satellites. For GPS to work effectively the satellites in the sky should be widely distributed rather than all satellites arranged in one section of the sky. The HDOP is proportional to the error in both the latitude and longitude directions while the VDOP only pertains to the altitude error i.e. (HDOP $\propto \sqrt{\sigma_{Lat}^2 + \sigma_{Lon}^2}$ and VDOP $\propto \sigma_{Alt}^2$). The HDOP and VDOP give an indication of the width of the distribution, 1 for an ideal arrangement, 2-5 for a good arrangement and > 10 for a poor arrangement. Therefore a reasonable measurement covariance matrix for GPS data could be something like the following.

$$R(k) = \frac{1}{N}\begin{bmatrix} c_1 HDOP & 0 & 0 \\ 0 & c_2 HDOP & 0 \\ 0 & 0 & c_3 VDOP \end{bmatrix}, \quad (3)$$

where N is the number of satellites, and the constants c_1, c_2, c_3 are determined by the particular device and in the case of c_1 and c_2 the difference in relative error between latitude and longitude. Therefore by upadating the measurement covariance matrix constantly the Kalman gain should automatically take into account the number of satellites and their arrangement.

The GPS unit differentiates between horizontal and vertical errors because in almost all cases the error in the altitude is considerably larger than the error in either of the horizontal directions. GPS requires a line of sight to the satellites in use and so (while at a low latitude) the satellites in use may be evenly distributed in east-west and north-south directions but the GPS can only ever use satellites above the horizon [1].

IV. Extended Kalman Filter (EKF)

The extended Kalman filter can accept a nonlinear state propagation equation $f(\mathbf{x}(k))$ as well as a nonlinear measurement equation $h(\mathbf{x}(k)$. The extended Kalman filter (EKF) handles the nonlinear state dynamics by taking a first order Taylor series expansion of the state propagation and measurement equations around the current state estimate. In the following equations we denote the true state of the system at time $k\Delta t$ as $\mathbf{x}(k)$ and the state estimate at the same time as $\hat{\mathbf{x}}(k|k)$.

$$\begin{aligned} \mathbf{x}(k+1) &= f(\mathbf{x}(k)) + \mathbf{w}(k), \\ \mathbf{z}(k+1) &= h(\mathbf{x}(k)) + \mathbf{v}(k), \end{aligned} \tag{4}$$

$$\begin{aligned} f(\mathbf{x}(k)) &= f(\hat{\mathbf{x}}(k|k)) + J_f(\hat{\mathbf{x}}(k|k) - \mathbf{x}(k)) + \dots, \\ &\approx f(\hat{\mathbf{x}}(k|k)) + J_f(\hat{\mathbf{x}}(k|k) - \mathbf{x}(k)), \end{aligned} \tag{5}$$

$$J_f = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & & \ddots & \vdots \\ \frac{\partial f_n}{\partial x_1} & \frac{\partial f_n}{\partial x_2} & \cdots & \frac{\partial f_n}{\partial x_n} \end{bmatrix} \tag{6}$$

$$J_h = \begin{bmatrix} \frac{\partial h_1}{\partial x_1} & \frac{\partial h_1}{\partial x_2} & \cdots & \frac{\partial h_1}{\partial x_n} \\ \vdots & & \ddots & \vdots \\ \frac{\partial h_n}{\partial x_1} & \frac{\partial h_n}{\partial x_2} & \cdots & \frac{\partial h_n}{\partial x_n} \end{bmatrix} \tag{7}$$

Equation (5) shows that for iteration of the algorithm the extended Kalman filter takes the nonlinear function $\mathbf{f}(\tilde{\mathrm{x}}(k))$ and truncates the Taylor series (5) at the first order. This allows the EKF to deal with nonlinear dynamics. However for highly nonlinear systems a first order expansion of the Taylor series may not be sufficient. In this case another nonlinear filter must be applied. A good candidate would be the unscented Kalman filter (UKF) which, though more complicated to implement, can be shown to be equivalent to a second order Taylor series expansion for any type of noise and a third order expansion if the system and measurement noises are Gaussian [11].

V. 3D Model of vehicle

In order to model a vehicle's movement accurately over a realistic road (i.e. not flat or straight) it is necessary to develop a three dimensional model of the vehicle's position, as well as the vehicle's pitch, roll and heading. Due to the nature of GPS measurements we have separated the state vector as follows.

$$\mathbf{x} = \begin{bmatrix} x \\ y \\ z \\ v_F \\ a_T \\ a_U \\ a_R \\ \theta \\ \phi \\ \gamma \end{bmatrix} \tag{8}$$

where x and y are the horizontal position vectors (pointing east and north, respectively), z is the vertical position vector and v_F is the forwards speed of the vehicle. The first acceleration term (a_T) is the forwards acceleration of the vehicle due to the driver input, i.e. total forwards acceleration $= a_T - g\sin(\theta)$, where g is the acceleration due to gravity and θ is the incline on the road surface. Following on, a_U is the upwards acceleration (as measured by the upwards facing accelerometer) and a_R is the acceleration vector pointing to the right, as shown in Figure 3.

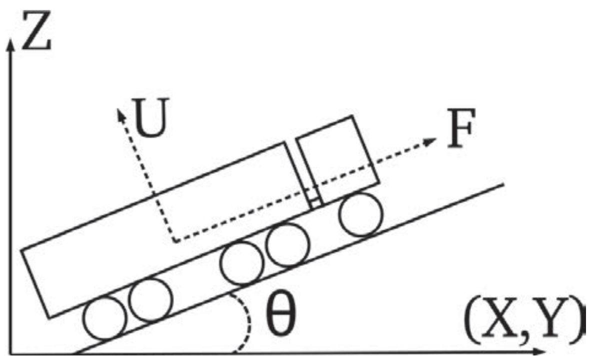


Fig. 1: Side-view of the vehicle with the directions shown. Note that the bottom axis is the projection onto the horizontal X,Y plane. The F and U vectors show the forwards and upwards directions used for the acceleration variables a_T and a_U, respectively.

The last terms in the state vector θ, ϕ, γ are, respectively, the pitch, roll and heading angles as shown in Figures 1, 2 and 3 where the forwards, upwards and rightwards direction for the positioning of the accelerometers are also defined.

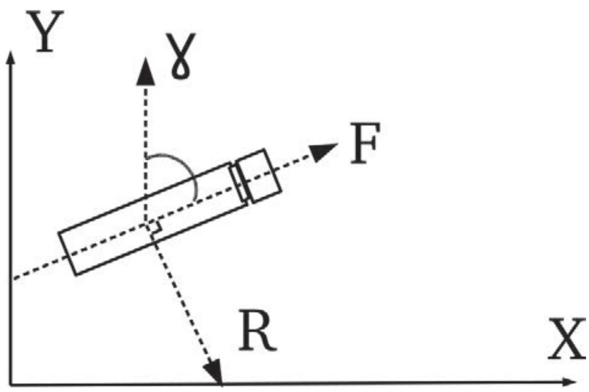


Fig. 2: Top-view of the vehicle with the directions shown. Note that the Y axis points North and the X axis points East. We have the same F direction as seen in figure 1 as well as the R direction for the rightwards facing acceleration term a_R.

It is worth noting that the upwards and rightwards acceleration terms (a_U, a_R) are the accelerations as measured by the accelerometers. Therefore the reaction forces provided by the ground in response to the acceleration of gravity and any centripetal acceleration will not register and must be considered. However by estimating the heading γ, roll ϕ and linear vehicular acceleration a_T through accelerometer measurements we should be able to infer the driver input with respect to acceleration profiles and cornering.

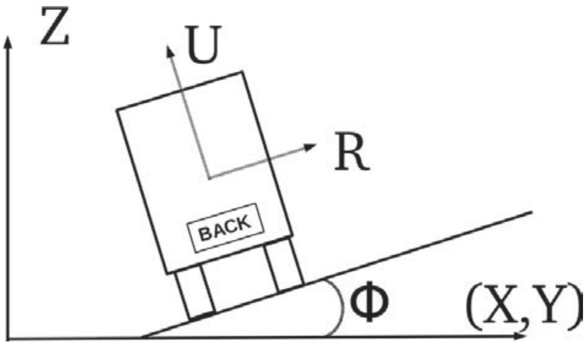


Fig. 3: Rear-view of the vehicle with the directions shown. Note that as in figure 1 the bottom axis is the projection onto the horizontal X,Y plane.

$$\dot{\mathrm{x}} = \begin{bmatrix} v_F \sin(\gamma)\cos(\theta) \\ v_F \cos(\gamma)\cos(\theta) \\ v_F \sin(\gamma)\cos(\theta) \\ v_F \sin(\theta) \\ a_T + g\sin(\theta) \\ j_T \\ j_U \\ j_R \\ -\left(\frac{a_U + g\cos(\theta)\cos(\phi) + v_{\mathrm{roll}}\dot{\phi}}{v_F}\right) \\ -\left(\frac{a_U + g\cos(\theta)\cos(\phi) + v_F\dot{\theta}}{v_{\mathrm{roll}}}\right) \\ -\left(\frac{a_R + g\sin(\theta)\phi}{v_F}\right) \end{bmatrix}, \quad (9)$$

By differentiating equation (8) with respect to time we get equation (9). In equation (9) the dotted terms $\dot{\theta}$ denote the time derivatives of the angles and v_{roll} denotes the speed of the accelerometer in the direction of increasing ϕ. The time derivatives of the three acceleration variables $\dot{a}_T = j_T, \dot{a}_U = j_U, \dot{a}_R = j_R$ are unknowns to do with driver input and the road and are therefore treated as random variables.

For ease of implementation some simplifying assumptions were made, namely that the current filter assume that the corners of the road have no camber and that the vehicle does not lean during corners. Therefore the $\dot{\phi}$ term reduces to noise. Clearly this approximation is not exact, however we believe that for low-speed navigation (e.g. city driving) this assumption should be valid. We intend to add camber and lean terms in once accelerometer data is produced from field testing.

Equation (9) is therefore obviously nonlinear, hence the necessity of a nonlinear filter. Of these however the extended Kalman filter is the simplest to implement and has a similar computational load to the unscented Kalman filter and generally a lighter load than the particle filter [11] [9].

VI. Testing using artificial data

During design we wanted to be able to test the theoretical benefits of combining GPS and accelerometer data using the EKF. We built a model where a simulated vehicle was driving along a hilly, straight road while the 'driver' applied positive or negative accelerations while going up or down the hill.

We then simulated Gaussian noise to be added to both position and accelerometer measurements of the vehicle as it 'drives' along the road. In Figure 4 we see the measured, estimated and true height of a vehicle after one such simulation.

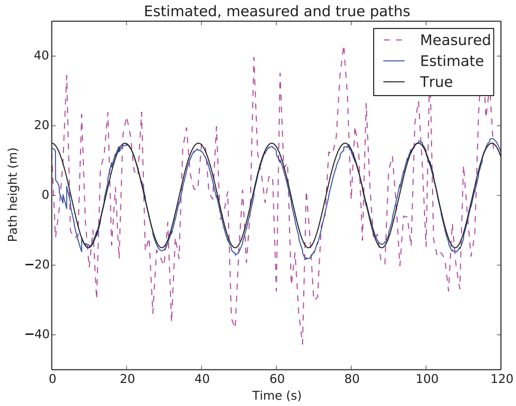


Fig. 4: The height of a simulated 3D vehicle with measured, estimated and true values shown. The covariance for the position measurements was taken from empirical data collected via a commercial GPS positioned at the University of Otago.

We then compared the performance of the system when relying on either position data, accelerometer data or a combination of both. The result of that test is in figure 5.

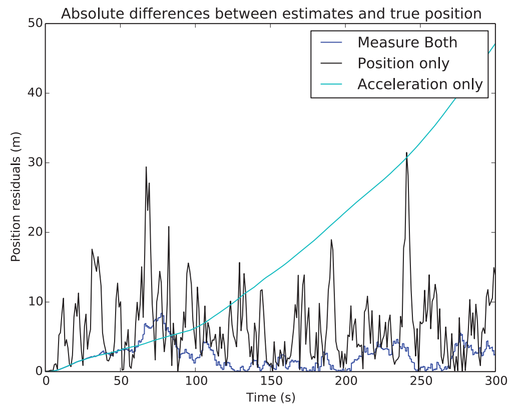


Fig. 5: The absolute error in position is shown for systems using just position (GPS) measurements, just accelerometer measurments or combining both.

As you can see in Figure 5 the result of combining both position and accelerometer measurements, predictably, gives better results than using either system seperately. The figure is instructive, however, in illustrating the benefits of accelerometer data. Note that while the position-only data is jagged and quickly leaps from one position to another by measuring and estimating the second derivatives of position the change in position error is much smoother and more realistic. However using a solely accelerometer-based navigation system means using dead reckoning to infer position and so any errors may accumulate which is why the accelerometer-only data tracks further and further off from the true position.

As another way of comparing the benefit of adding accelerometer data to GPS measurements we compared the effect of increasing the position noise on systems with either only position measurements, only accelerometer measurements or both.

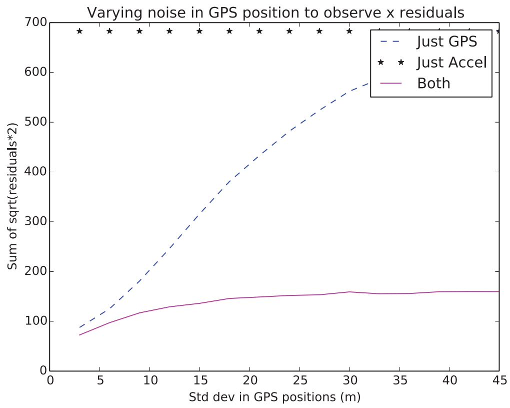


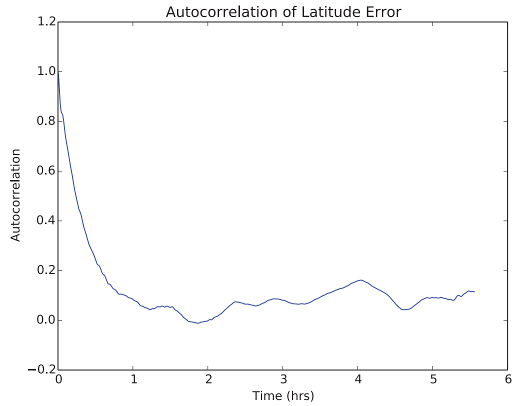
Fig. 6: Standard deviation of position noise versus errors in position for systems using position-only measurements, accelerometer-only measurements or both.

As you can see in Figure 6, increasing the noise in position measurements, predictably increased the errors in systems using position and combined measurements and did not change purely accelerometer-based predictions. What is interesting is how the gradient of the combined system lessens compared to the purely position-based system as the EKF places more emphasis on the accelerometer data rather than the position data. This effect could be very useful for vehicles briefly losing sight of satellites, for example in urban canyons. As the number of satellites decreases the covariance of the GPS measurements increases and the EKF places more emphasis on the accelerometer data until the line of sight to the satellites is reestablished. Therefore for short outages of GPS signals the EKF will allow for accelerometer measurements to more accurately determine the position and heading of a vehicle with obstructed GPS signal.

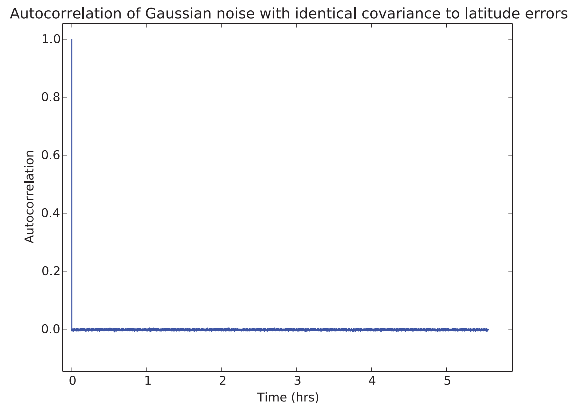
VII. Non-independent measurements

A commonly made assumption in many Kalman filter-based systems is that the measurement noise term is independent from the previous noise terms[1]. However empirical data we obtained seems to imply that for GPS measurements this assumptions is flawed. In Figure 7(a) we see the autocorrelation of empirical GPS data compared to the autocorrelation of independent Gaussian noise with the same mean and variance as the observed data.

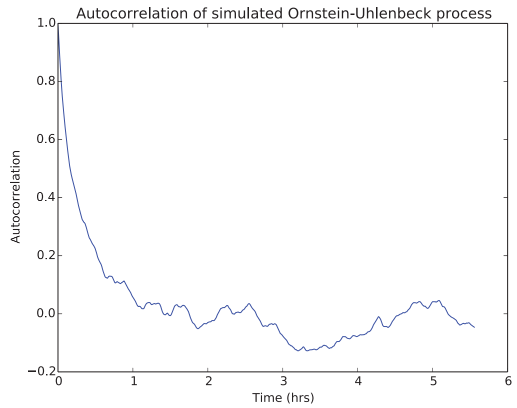
The autocorrelation function of a signal is an indicator of how strongly correlated a signal is with itself. In the case of independent Gaussian noise the time for the autocorrelation to decay to zero is theoretically zero. We see in Figure 7(b) that for a data set of the same length and with the same



(a)



(b)



(c)

Fig. 7: The autocorrelation of empirical GPS data compared with the autocorrelation of independent Gaussian noise with the same mean and variance as the observed data and the autocorrelation of Ornstein-Uhlenbeck with similar structure to the empirical data. The data was taken using a GlobalSat BU-353S4 GPS unit on the L1 frequency with excellent sky view at a rate of 1Hz for a period of more than 250 hours. For readability the time axis has been cut off at 5 hours.

mean and variance as the empirical data this is fairly close to zero. However it is equally clear from Figure 7(a) that the autocorrelation function for the measured latitude error does not quickly decay (keeping in mind that the data is plotted at 1 second intervals with the time scale in hours). This time dependence means that if a GPS position measurement is off by a significant amount then the following measurement will also likely be considerably off from the true position.

We have been working with another noise model known as an Ornstein-Uhlenbeck process to more accurately model the noise seen in empirical GPS measurements. The Ornstein-Uhlenbeck process has some appealing features as a noise model for GPS measurements. As you can see in Figure 7(c) the autocorrelation function of the O-U model shows similar long-term structure to the empirical data. Also the Ornstein-Uhlenbeck process is mean reverting and so will drift towards some long-term mean. The Ornstein-Uhlenbeck model is defined by the stochastic differential equation below,

$$dx_t = \theta(\mu - x_t)dt + \sigma dW_t, \tag{10}$$

where θ, σ and μ are constant parameters of the system, x_t is the observed variable and W_t is the Weiner process noise.

To compare the Ornstein-Uhlenbeck model against the independent Gaussian model we used the Akaike Information Criterion (AIC) which is defined like so,

$$AIC = 2k - 2\ln(L),$$

where k is the number of adjustable parameters and L is the maximised likelihood function of the model [12]. The better the model, the lower that model's AIC value. We used data from a commercial GPS unit (described in Figure 7) with a sampling rate of 1Hz over a period of around 250 hours. The AIC for the O-U noise model for the longitude information was -4.63×10^6 while the AIC for the Gaussian noise model for the same data was 2.857×10^6, this was the smallest difference between O-U and Gaussian AIC values for all of the position variables. Similar readings were taken in different locations and with different equipment and also yielded a considerably lower AIC for the O-U model as opposed to the Gaussian model. Therefore our preliminary results indicate that the Ornstein-Uhlenbeck model is worthy or further study.

VIII. Conclusion

We developed an extended Kalman filter which combines GPS and accelerometer measurements for improved estimates of position, velocity and acceleration on a road vehicle. We are able to infer parameters of driver behaviour without modifying the vehicle and deal with short-term loss of satellites, such as experienced when driving through urban canyons. We also propose the Ornstein-Uhlenbeck noise model as an improvement on the standard independent Gaussian model which better fits the empirical GPS data.

IX. Future Work

We are currently assembling a Raspberry Pi extension board to hold an accelerometer. Once this is built we will be able to begin field testing with a vehicle on the open road. During this phase we aim to characterise the noise for accelerometer measurements as well as how the GPS data is affected by driving vs stationary data.

We also intend to test the perfomance of the extended Kalman filter against other nonlinear filters including the unscented Kalman filter or the particle filter. There is some evidence to support filters such as the unscented Kalman filter and particle filter as being able to better handle highly nonlinear systems [8] [11].

We are currently working on an algorithm that uses different noise models to account for the correlated nature of the gps noise. Possible candidates are the Ornstein-Uhlenbeck, autoregressive or moving average noise models which more accurately represent the noise that we have observed in our data.

Once the algorithm is finished and working with the current GPS unit and accelerometer we intend to adapt it so that it could run using the GPS unit and in-built accelerometer available on most smart phones. This would be doubly useful as the devices are widely available and if the user so desired it would be a simple matter to send the filtered data from the device to any other location via the 3G network.

References

[1] P. D. Groves, *Principles of GNSS, Inertial, and Multisensor Integrated Navigation Systems*. Artech House, 2008.

[2] W. Ding and J. Wang, "Precise velocity estimation with a stand-alone gps receiver," *Journal of Navigation*, vol. 64, no. 2, pp. 311–325, 2011.

[3] A. A. Thompson, "A ballistic filter for gps and accelerometer measurements," *Army Research Laboratory*, April 2010. [Online]. Available: http://oai.dtic.mil/oai/oai?verb=getRecordmetadataPrefix=htmlidentifier =ADA518127

[4] S. Antonov, A. Fehn, and A. Kugi, "Unscented kalman filter for vehicle state estimation," *Vehicle System Dynamics*, vol. 49, no. 9, pp. 1497–1520, September 2011.

[5] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Transactions of the ASME, Series D: Journal of Basic Engineering*, vol. 82, pp. 35–45, 1960.

[6] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation with applications to tracking and navigation*. Wiley-Interscience, 2001.

[7] M. St-Pierre and D. Gingras, "Comparison between the unscented kalman filter and the extended kalman filter for the position estimation module of an integrated navigation information system." *Intelligent Vehicles Symposium, 2004 IEEE*, 2004.

[8] N. J. Gordon, D. J. Salmond, and A. F. M. Smithordon, "Novel approach to nonlinear/non-gaussian bayesian state estimation," *IEEE Proceedings F (Radar and Signal Processing)*, vol. 140, no. 2, pp. 107–113, 1993.

[9] M. Afonso, "Particle filter and extended kalman filter for nonlinear estimation: a comparative study," *IEEE Transactions on Signal Processing*, vol. 10, 2010.

[10] J. D. Hamilton, *Time series analysis*. Princeton university press, vol. 2. 1994.

[11] S. Julier and J. Uhlmann, "Unscented filtering and nonlinear estimation," *Proceedings of the IEEE*, vol. 92, pp. 401–422, March 2004.

[12] H. Akaike, "A new look at the statistical model identification," *IEEE Transactions on Control*, vol. 6, pp.716-723, 1974.