On Reduced-Order Kalman Filters For GPS Position Filtering

J. Shima

6/2/2001

BACKGROUND

The following paper describes the theory and implementation issues regarding a new position-filtering scheme to smooth GPS lat/lon position coordinates over time.

KALMAN FILTERING

R.E. Kalman devised Kalman filtering in the 1960s. After it was discovered useful for many engineering problems, the Kalman filter is used today in many applications (spacecraft navigation, ballistic missile navigation, radar target tracking, signal filtering and prediction). The Kalman filter is applicable in problems that have definable dynamic models with non-stationary input signals. The Kalman filter will outperform a Wiener filter with non-stationary signals. With stationary signals, the Wiener and Kalman filters produce similar results. It has also been proven that the RLS adaptive filter and a Kalman filter are related. So, the Kalman filter is the optimal linear filter (in the least squares sense) with stationary or non-stationary signal inputs, given a proper process model.

The Kalman filter is a mathematical way to optimally combine incoming measurement data with the predicted filter state. The state of the Kalman filter is controlled by a linear stochastic difference equation, which is typically a time-varying (dynamic) model of the problem to be solved. The filter state is predicted ahead using temporal samples. When a measurement is available, the filter state is then corrected using the residual (or innovation) term, which is the difference in the predicted filter state and the actual observable. This gives rise to the “predictor-corrector” nature of the Kalman filter structure. Furthermore, the predictor equations are sometimes referred to as the “time update” equations, while the corrector equations are known as the “measurement update” equations.

The discrete-time Kalman filter is fully described by the following set of matrix equations.

MODEL EQUATIONS:

System dynamic model, with filter state vector **x**:

$$\begin{aligned} \mathbf{x_{k+1}} &= \mathbf{\Phi}\,\mathbf{x_k} + \mathbf{w_k} \\ \mathbf{w_k} &= \mathrm{N}(\mathbf{0}, \mathbf{Q_k}) \end{aligned} \tag{1}$$

Measurement model, with measurement observable z:

$$\begin{aligned} \mathbf{z_{k+1}} &= \mathbf{H_k}\,\mathbf{x_k} + \mathbf{v_k} \\ \mathbf{v_k} &= \mathrm{N}(\mathbf{0}, \mathbf{R_k}) \end{aligned} \tag{2}$$

KALMAN EQUATIONS:

Time update state prediction:

$$\tilde{\mathbf{x}}_{\mathbf{k+1}} = \mathbf{\Phi}\,\hat{\mathbf{x}}_{\mathbf{k}} \tag{3}$$

Time update state covariance prediction:

$$\tilde{\mathbf{P}}_{\mathbf{k+1}} = \mathbf{\Phi}\,\hat{\mathbf{P}}_{\mathbf{k}}\,\mathbf{\Phi}^{T} + \mathbf{Q_k} \tag{4}$$

Kalman gain:

$$\mathbf{K_k} = \tilde{\mathbf{P}}_\mathbf{k}\mathbf{H}_\mathbf{k}^\mathbf{T}\left[\mathbf{H_k}\tilde{\mathbf{P}}_\mathbf{k}\mathbf{H}_\mathbf{k}^\mathbf{T} + \mathbf{R_k}\right]^{-1} \tag{5}$$

Measurement state update:

$$\hat{\mathbf{x}}_\mathbf{k} = \tilde{\mathbf{x}}_\mathbf{k} + \mathbf{K_k}\left[\mathbf{z_k} - \mathbf{H_k}\tilde{\mathbf{x}}_\mathbf{k}\right] \tag{6}$$

Measurement state covariance update:

$$\hat{\mathbf{P}}_\mathbf{k} = \left[\mathbf{I} - \mathbf{K_k}\mathbf{H_k}\right]\tilde{\mathbf{P}}_\mathbf{k} \tag{7}$$

The definition of the parameters in each of the Kalman filter is described below in relation to the positioning problem at hand. The below parameters are defined for a 2nd-order filter where the unmodeled accelerations are considered to be much shorter in time compared to the filter's sampling rate T (that is, T>> τ_m)

Parameter	Symbol	Size	Description
State vector	$\mathbf{x_k}$	(2x1)	$\mathbf{x_k} = \begin{bmatrix} x & \dot{x} \end{bmatrix}^T$
Predicted state	$\tilde{\mathbf{x}}_\mathbf{k}$	(2x1)	Predicted estimate of unknowns at epoch k
Filtered state	$\hat{\mathbf{x}}_\mathbf{k}$	(2x1)	Filtered (updated) estimates at epoch k
Predicted state covariance	$\tilde{\mathbf{P}}_\mathbf{k}$	(2x2)	Uncertainty (error) in predicted state at epoch k $\mathbf{P_k} = E\left[(\mathbf{x} - \hat{\mathbf{x}})(\mathbf{x} - \hat{\mathbf{x}})^T\right]$
Filtered state covariance	$\hat{\mathbf{P}}_\mathbf{k}$	(2x2)	Uncertainty in filtered state at epoch k
State transition matrix	$\mathbf{\Phi}$	(2x2)	Describes how the states change over time: $\mathbf{\Phi} = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix}$
System driving noise matrix	$\mathbf{Q_k}$	(2x2)	Uncertainty of predicting state forward. $\mathbf{Q_k} = q \cdot \begin{bmatrix} \frac{T^3}{3} & \frac{T^2}{2} \\ \frac{T^2}{2} & T \end{bmatrix}$ q = constant given by Singer model[1]
Measurement vector	$\mathbf{z_k}$	(1x1)	Positions derived from GPS are used as the measurement source in the filtering scheme
Measurement noise covariance	$\mathbf{R_k}$	(1x1)	Covariance of the GPS coordinate position fix
Design matrix	$\mathbf{H_k}$	(1x2)	Coefficient matrix for relating measurements to filter states. $\mathbf{H_k} = \begin{bmatrix} 1 & 0 \end{bmatrix}$
Kalman gain	$\mathbf{K_k}$	(2x1)	Weighting matrix for combining predicted state with incoming measurements
Identity matrix	$\mathbf{I}$	(2x2)	Identity matrix

Table 1. Description of parameters used in Kalman filter.

KALMAN FILTER THEORY FOR POSITION FILTERING

In using a 2^{nd}-order Kalman filter for positioning, the filter states are able to estimate target position and velocity. This is known as a constant-velocity model. However, a constant-velocity model does not consider any target accelerations. So, the filter performs poorly if it does not take these accelerations into account. Since acceleration is consider "unmodeled" by the Kalman filter, loss of tracking when the target undergoes a maneuver is common with this type of filter. In its present state, it is not very useful for position filtering where the target experiences any acceleration.

To solve this problem, one can use a 3^{rd}-order Kalman filter, which estimates position, velocity, and acceleration. This is known as a constant-acceleration model. This filter will now model target maneuvers at the expense of more computations. The matrix orders increase to 3x3 for the gain and covariance. Back in the 1960s and 70s, this was a huge increase in computational power. Since the Kalman filter made its way into radar applications, there is a wealth of information regarding position and position rate target tracking using the Kalman filter. Several target maneuver models were developed, most notably by Singer[2], which made this Kalman filtering problem easier to tackle.

Since targets with unknown acceleration models were common (aircraft, submarines, etc), the Kalman filter had to change its parameters on the fly when a maneuver was detected in order to keep target lock. Singer developed an acceleration model that could be used in the discrete-time Kalman filter using the constant-acceleration model.

For various cases, the Kalman filter transition matrix and process noise covariance matrix take on discrete forms.

Define the target's maneuver time constant is defined as τ_m, the target acceleration variance as $\sigma^2{}_m$, and the filter sampling time as T.

If $T \ll \tau_m$, then

$$\mathbf{Q} = \frac{2\sigma^2{}_m}{\tau_m} \cdot \begin{bmatrix} \frac{T^5}{20} & \frac{T^4}{8} & \frac{T^3}{6} \\ \frac{T^4}{8} & \frac{T^3}{3} & \frac{T^2}{2} \\ \frac{T^3}{6} & \frac{T^2}{2} & T \end{bmatrix} \qquad (8)$$

$$\mathbf{\Phi} = \begin{bmatrix} 1 & T & \frac{T^2}{2} \\ 0 & 1 & T \\ 0 & 0 & 1 \end{bmatrix} \qquad (9)$$

If $T \gg \tau_m$, then

$$\mathbf{Q} = \sigma^2{}_m \cdot \begin{bmatrix} \frac{2T^3}{3} \cdot \tau_m & T^2 \cdot \tau_m & \tau_m^2 \\ T^2 \cdot \tau_m & 2T \cdot \tau_m & \tau_m \\ \tau_m^2 & \tau_m & 1 \end{bmatrix} \qquad (10)$$

$$\mathbf{\Phi} = \begin{bmatrix} 1 & T & T \cdot \tau_m \\ 0 & 1 & \tau_m \\ 0 & 0 & 1 \end{bmatrix} \quad (11)$$

For the above case, when τ_m is small relative to T, acceleration estimates are small, and the filter can conveniently be reduced to a 2^{nd}-order constant velocity filter. This common filter reduction looks like:

$$\mathbf{Q} = 2\sigma_m^2 \cdot \tau_m \cdot \begin{bmatrix} \frac{T^3}{3} & \frac{T^2}{2} \\ \frac{T^2}{2} & T \end{bmatrix} \quad (12)$$

$$\mathbf{\Phi} = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} \quad (13)$$

Which happens to be shown in Table 1.

By using the 2^{nd}-order filter, the computations are straightforward and no matrix operations are necessary. Also, potential matrix ill conditioning is also avoided. The common Joseph's form of the measurement covariance update is unnecessary since the 2^{nd}-order filter is simplified into individual equations by hand deriving all of the Kalman filter equations.

In order to account for unmodeled accelerations in the 2^{nd}-order filter, the general concept of adding "state noise" to the process noise matrix is a common practice. One such way to accomplish this is to change the Q matrix scalar term. Since this scalar embodies the target's acceleration variance and maneuver time constant (using the Singer model), we can effectively change these parameters to reflect the estimated accelerations. This is sometimes referred to as "gear shifting" the noise process matrix.

With this approach, when the Q scalar is small (i.e. no acceleration) the covariance propagation is negligible and the amount of filtering on the position fixes is heavy. As the Q scalar increases in value (denoting some target acceleration), the filter tracks the changes more readily, applying less filtering to the position fixes.

By changing the Q scalar, we can still expect a 2^{nd}-order filter to perform reasonably. But the filter is still considered "suboptimal", since the acceleration state has been eliminated. By changing the Q scalar in the gear-shifting manner, you are effectively adding a constant spectral density to the noise process matrix. It has been shown in this case that one can optimize steady-state performance or transient filter performance, but not both[3].

There has been much research on optimizing steady state and transient performance in a reduced-order Kalman filter. Specifically, the paper by Hutchinson, et al. derives a minimum-variance reduced-order Kalman filter (MVRO) based on the above findings. It is proven that for a MVRO, the best "state noise" to add to the filter is a time-varying function of the unestimated states and a cross correlation of unestimated states and the filter's estimation error.

From Hutchinson[3], the optimal state noise is computed as:

$$\begin{aligned}\mathbf{Q}'(k) &= \mathbf{\Phi_{11}}(k-1)\cdot\mathbf{P_{e2}}(k-1\,|\,k-1)\cdot\mathbf{\Phi_{12}^{T}}(k-1)\\ &+\mathbf{\Phi_{12}}(k-1)\cdot\mathbf{P_{2e}}(k-1\,|\,k-1)\cdot\mathbf{\Phi_{11}^{T}}(k-1)\\ &+\mathbf{\Phi_{12}}(k-1)\cdot\mathbf{P_{22}}(k-1\,|\,k-1)\cdot\mathbf{\Phi_{12}^{T}}(k-1)\end{aligned} \tag{14}$$

and,

$$\mathbf{P_{22}} = \mathbf{\Phi}_{22}(k-1)\cdot\mathbf{P_{22}}(k-1)\cdot\mathbf{\Phi_{22}^{T}}(k-1)+\mathbf{Q_{22}}(k) \tag{15}$$

where,

$$\begin{aligned}\mathbf{P_{e2}} &= E\left[(\mathbf{x_1}-\hat{\mathbf{x}}_\mathbf{1})\mathbf{x_2}\right]\\ \mathbf{P_{22}} &= E[\mathbf{x_2}\mathbf{x_2^T}]\end{aligned} \tag{16}$$

The filter state structure for this case can be written as:

$$\begin{bmatrix}\mathbf{x_1}(k)\\ \mathbf{x_2}(k)\end{bmatrix} = \begin{bmatrix}\mathbf{\Phi_{11}}(k) & \mathbf{\Phi_{12}}(k)\\ 0 & \mathbf{\Phi}_{22}(k)\end{bmatrix}\cdot\begin{bmatrix}\mathbf{x_1}(k-1)\\ \mathbf{x_2}(k-1)\end{bmatrix}+\begin{bmatrix}\mathbf{w_1}(k)\\ \mathbf{w_2}(k)\end{bmatrix} \tag{17}$$

You can see in Eq.17 that the state update model of Eq.1 is split into two sections - the estimated states $\mathbf{x_1}$ and the unestimated states $\mathbf{x_2}$. The state transition matrix is also partitioned in a similar fashion.

In the position filtering application, $\mathbf{x_1}$ is the constant-velocity part of the reduced-order filter and $\mathbf{x_2}$ is the unmodeled acceleration state.

The main difference in the Hutchinson paper is that you start out with a conventional 3rd -order Kalman filter, as defined by Eqs.8-11, and simplify this to a MVRO filter.

Different vehicle applications need to model slow and fast changes. Thus the acceleration time constant can be on the order of seconds. In the worst case of slow movement, the proper 3rd order filter using the Singer model is given by Eqs.8-9, since the sampling time T is much less than the target's acceleration time constant (in general).

Using this structure, the 2nd-order MVRO partitioned filter sections look like the following:

$$\mathbf{x_1} = \begin{bmatrix}x\\ \dot{x}\end{bmatrix},\ \mathbf{x}_2 = [\ddot{x}] \tag{18}$$

$$\mathbf{\Phi}_{11} = \begin{bmatrix}1 & T\\ 0 & 1\end{bmatrix},\ \mathbf{\Phi}_{12} = \begin{bmatrix}\frac{T^2}{2}\\ T\end{bmatrix},\ \mathbf{\Phi}_{22} = [1] \tag{19}$$

$$\mathbf{Q}_{11} = \frac{2\sigma_m^2}{\tau_m}\cdot\begin{bmatrix}\frac{T^5}{20} & \frac{T^4}{8}\\ \frac{T^4}{8} & \frac{T^3}{3}\end{bmatrix},\ \mathbf{Q}_{22} = \frac{2\sigma_m^2}{\tau_m}\cdot[T] \tag{20}$$

One can see that $\mathbf{\Phi_{11}}$ reduces to the standard 2nd-order transition matrix as shown in Eq. 9 and Table 1., while the $\mathbf{Q_{11}}$ matrix is the upper submatrix of the 3rd-order Q matrix shown in Eq. 8.

Since the above equations only effect the "time update" equations, the modified 2nd-order Kalman equations take on the following form:

Time update state prediction:

$$\tilde{\mathbf{x}}_{\mathbf{1}_{\mathbf{k+1}}} = \mathbf{\Phi}_{11}\ \hat{\mathbf{x}}_{\mathbf{1}_{\mathbf{k}}}$$

Time update state covariance prediction:

$$\tilde{\mathbf{P}}_{\mathbf{k+1}} = \mathbf{\Phi}_{11}\ \hat{\mathbf{P}}_{\mathbf{k}}\ \mathbf{\Phi}_{11}^{\ T} + \mathbf{Q}_{11} + \mathbf{Q}'_{k}$$

For the state noise update terms in $\mathbf{Q}'_{\mathbf{k}}$, we already know that the covariance matrix **P** holds the state-estimation errors.

$$\mathbf{P}_{\mathbf{k}} = E\left[(\mathbf{x}-\hat{\mathbf{x}})(\mathbf{x}-\hat{\mathbf{x}})^{T}\right]$$

So, to get $\mathbf{P}_{\mathbf{2e}}$ and $\mathbf{P}_{\mathbf{e2}}$, we can compute a crude estimate by using:

$$\mathbf{P}_{\mathbf{e2}} = E\left[(\mathbf{x}_{\mathbf{1}} - \hat{\mathbf{x}}_{\mathbf{1}})\mathbf{x}_{\mathbf{2}}\right] = E[sqrt(trace(\mathbf{P}_{\mathbf{k}}))\cdot \mathbf{x}_{\mathbf{2}}]$$

$$\therefore \mathbf{P}_{\mathbf{e2}} = \begin{bmatrix} p_1 & p_2 \end{bmatrix}^{T}$$

$$\therefore \mathbf{P}_{\mathbf{22}} = E[\mathbf{x}_{\mathbf{2}}\mathbf{x}_{\mathbf{2}}^{\mathbf{T}}] = \sigma_A^2$$

Where $\mathbf{x}_{\mathbf{2}}$ is our estimate of acceleration. Also note that $\mathbf{P}_{\mathbf{22}}$ is just the 2nd moment of the acceleration estimate. Since we are directly estimating acceleration by differencing subsequent Stinger velocities, we can compute $\mathbf{P}_{\mathbf{22}}$ directly. If we further assume acceleration is a zero-mean stochastic process (which is valid for vehicles that accelerate and de-accelerate equally), then the acceleration variance is also equal to $\mathbf{P}_{\mathbf{22}}$.

We now have a reduced-order (2nd order) Kalman filter used for position filtering, where the unmodeled accelerations are accounted for using the extra "state noise" terms above. It is shown that using a technique described above, the filter is able to optimize both steady-state performance and transient filter performance. This is mostly due to the complex time-varying Q′ terms, which adapts the process noise matrix based on the estimated acceleration state.

KALMAN FILTER EQUATIONS

We use GPS ECEF coordinates for the position filtering, since this allows for separate filters to run on each coordinate X,Y,Z. The Kalman filtering problem can be split into separate filters when the incoming measurements are independent[1]. In the case of using ECEF coordinates, the position problem can be split into 3 separate 2-state Kalman filters. This relaxes the computational requirements of the overall filter structure. If one were to use E,N,U coordinates, the use of separate filters becomes questionable, since the ENU coordinates are no longer independent (they are functions of the ECEF coordinate transformation). A similar argument is presented in Blackman regarding radar range/range rate filtering in transformed Cartesian coordinates.

For a 2nd-order Kalman filter with the previously defined added state noise terms, the update equations can be algebraically manipulated into primitive forms. This also helps relieve the computational burden on the processor.

<u>Let:</u>
q = noise process matrix scalar as shown in Table 1
x = position filter state
$\dot{x}$ = velocity filter state
T = sample time

z = incoming noisy position measurement (from GPS receiver)
R = position measurement noise (from GPS receiver)

P = covariance matrix, $P = \begin{bmatrix} \sigma_x^2 & \sigma_{x\dot{x}} \\ \sigma_{x\dot{x}} & \sigma_{\dot{x}}^2 \end{bmatrix}$

2nd ORDER TIME-UPDATE EQUATIONS

State prediction:

$$x = x + \dot{x}T$$
$$\dot{x} = \dot{x}$$

Covariance prediction:

$$\sigma_x^2 = \sigma_x^2 + 2T\sigma_{x\dot{x}} + T^2\sigma_{\dot{x}}^2 + q\frac{T^5}{20} + q_{11}$$
$$\sigma_{x\dot{x}} = \sigma_{x\dot{x}} + T\sigma_{\dot{x}}^2 + q\frac{T^4}{8} + q_{12}$$
$$\sigma_{\dot{x}}^2 = \sigma_{\dot{x}}^2 + q\frac{T^3}{3} + q_{22}$$

where the added time-varying state noise terms are,

$$q_{11} = p_1T^2 + p_2T^3 + \sigma_A^2\frac{T^4}{4}$$
$$q_{12} = p_1T + 3p_2\frac{T^2}{2} + \sigma_A^2 \cdot \frac{T^3}{2}$$
$$q_{22} = 2p_2T + \sigma_A^2 \cdot T^2$$

2nd ORDER MEASUREMENT-UPDATE EQUATIONS

Kalman gain:

$$K_x = \frac{\sigma_x^2}{\sigma_x^2 + R}$$
$$K_{\dot{x}} = \frac{\sigma_{x\dot{x}}}{\sigma_x^2 + R}$$

Covariance update:

$$\sigma_x^2 = K_x \cdot R$$
$$\sigma_{x\dot{x}} = K_{\dot{x}} \cdot R$$
$$\sigma_{\dot{x}}^2 = \sigma_{\dot{x}}^2 - K_{\dot{x}} \cdot \sigma_{x\dot{x}}$$

Measurement state update:

$$x = x + K_x \cdot (z - x)$$

$$\dot{x} = \dot{x} + K_{\dot{x}} \cdot (z - x)$$

ADDED STATE NOISE TERMS

Let:

vel_k = GPS ECEF velocity at iteration k

α = smoothing filter coeff

$$acc_k = \frac{1}{T}(vel_k - vel_{k-1})$$

$$p_1 = \alpha \cdot p_1 + (1-\alpha) \cdot \sigma_x \cdot |acc_k|$$

$$p_2 = \alpha \cdot p_2 + (1-\alpha) \cdot \sigma_{\dot{x}} \cdot |acc_k|$$

$$\sigma_A^2 = \alpha \cdot \sigma_A^2 + (1-\alpha) \cdot acc_k^2$$

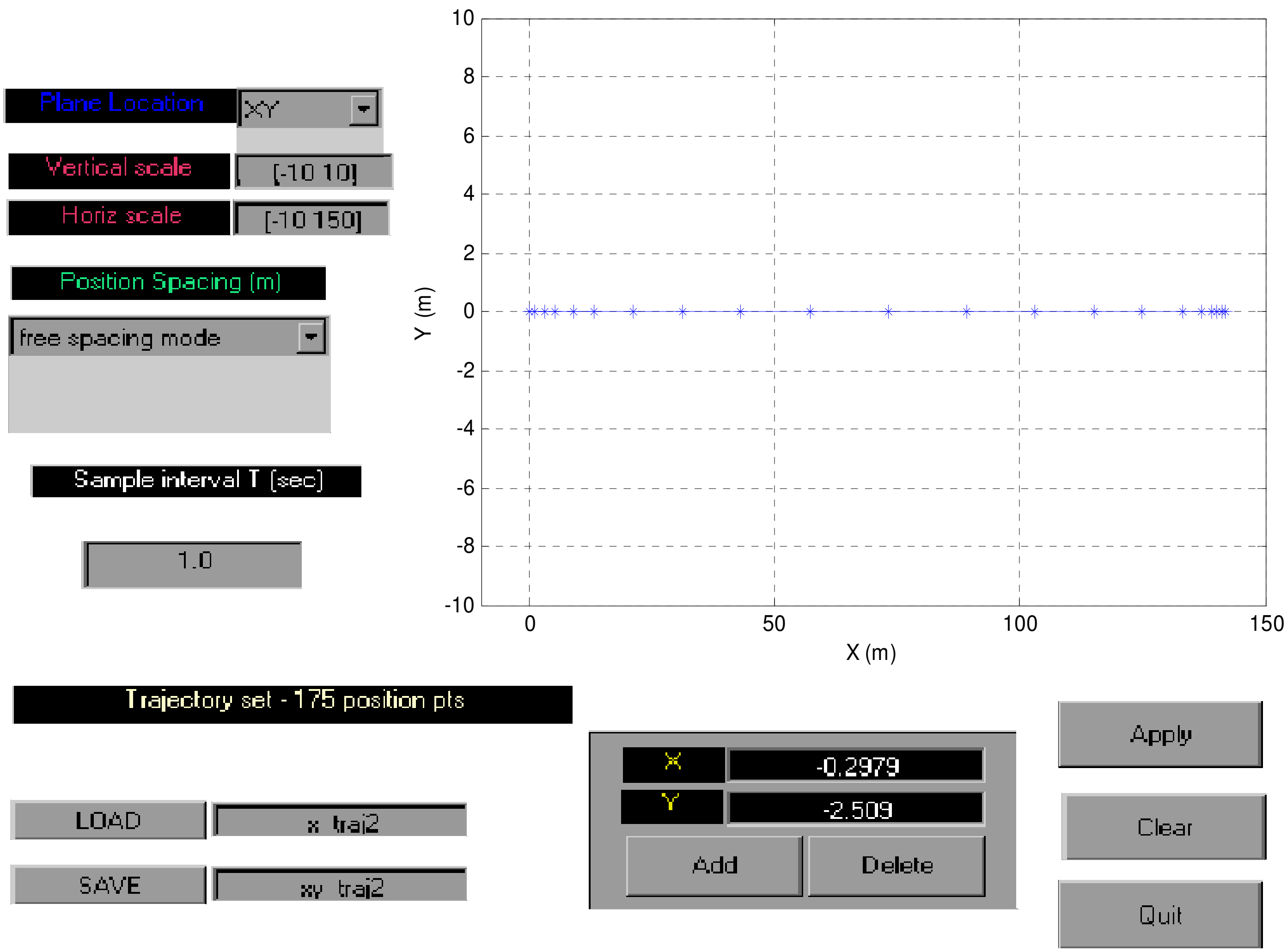
DYNAMIC FILTER SIMULATIONS

The dynamic behavior of the Q scalar (gear-shifted) method and the optimal state noise method was simulated in order to see the differences in dynamic tracking performance.

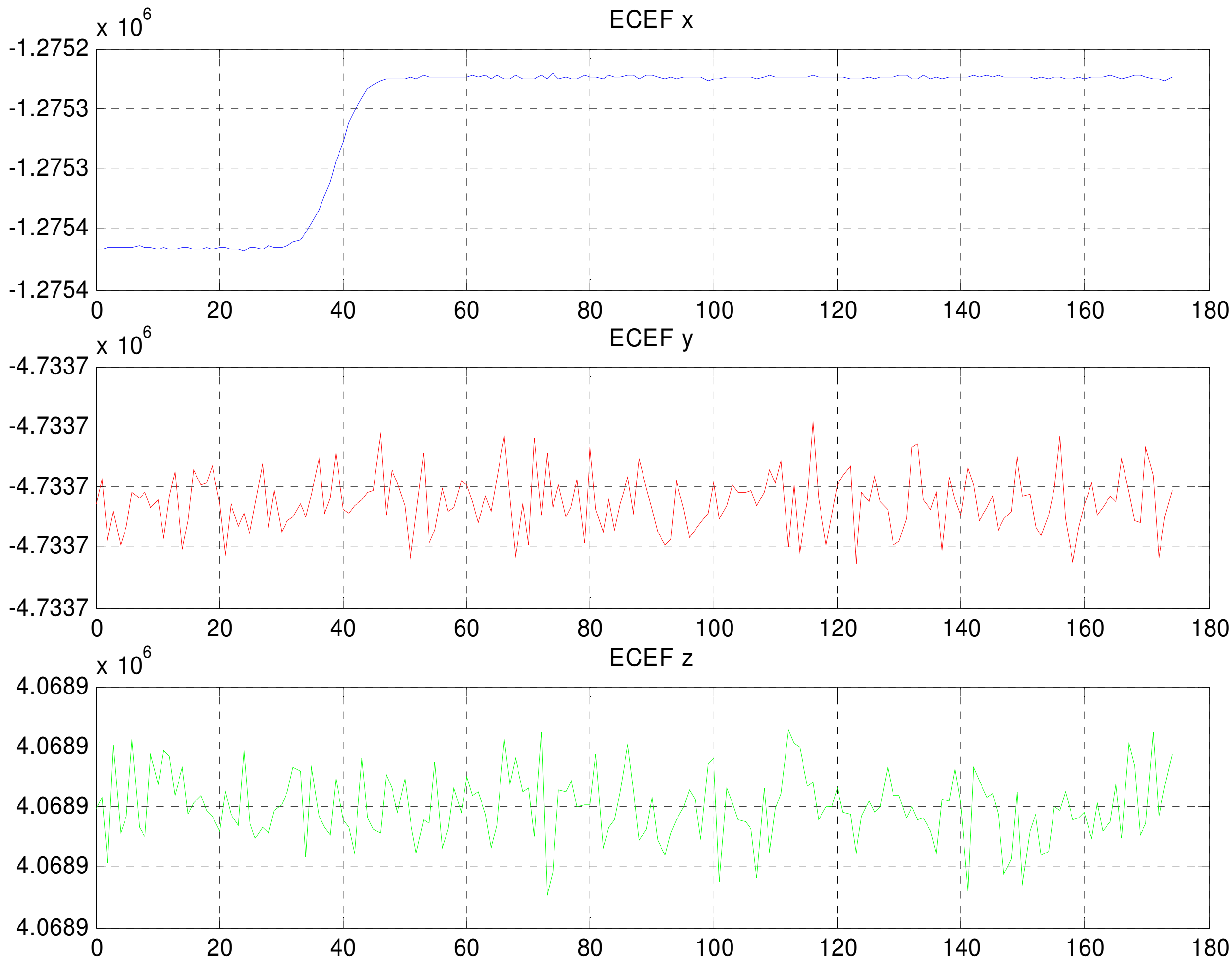
Below is a simulated trajectory along the x direction. Here, Gaussian white noise was added to the position coordinates to simulate a noisy GPS fix. Furthermore, the position covariance was set to a steady state value of 2. The position sampling interval T = 1 sec.

The trajectory was created using a Matlab tool that allows one to place spaced positions from a central location (0,0) in a 2D plane.

Here is a screenshot of the trajectory. The positions are offset from an arbitrary starting point. The starting point (0,0) was chosen to be the center of the simulated antenna.



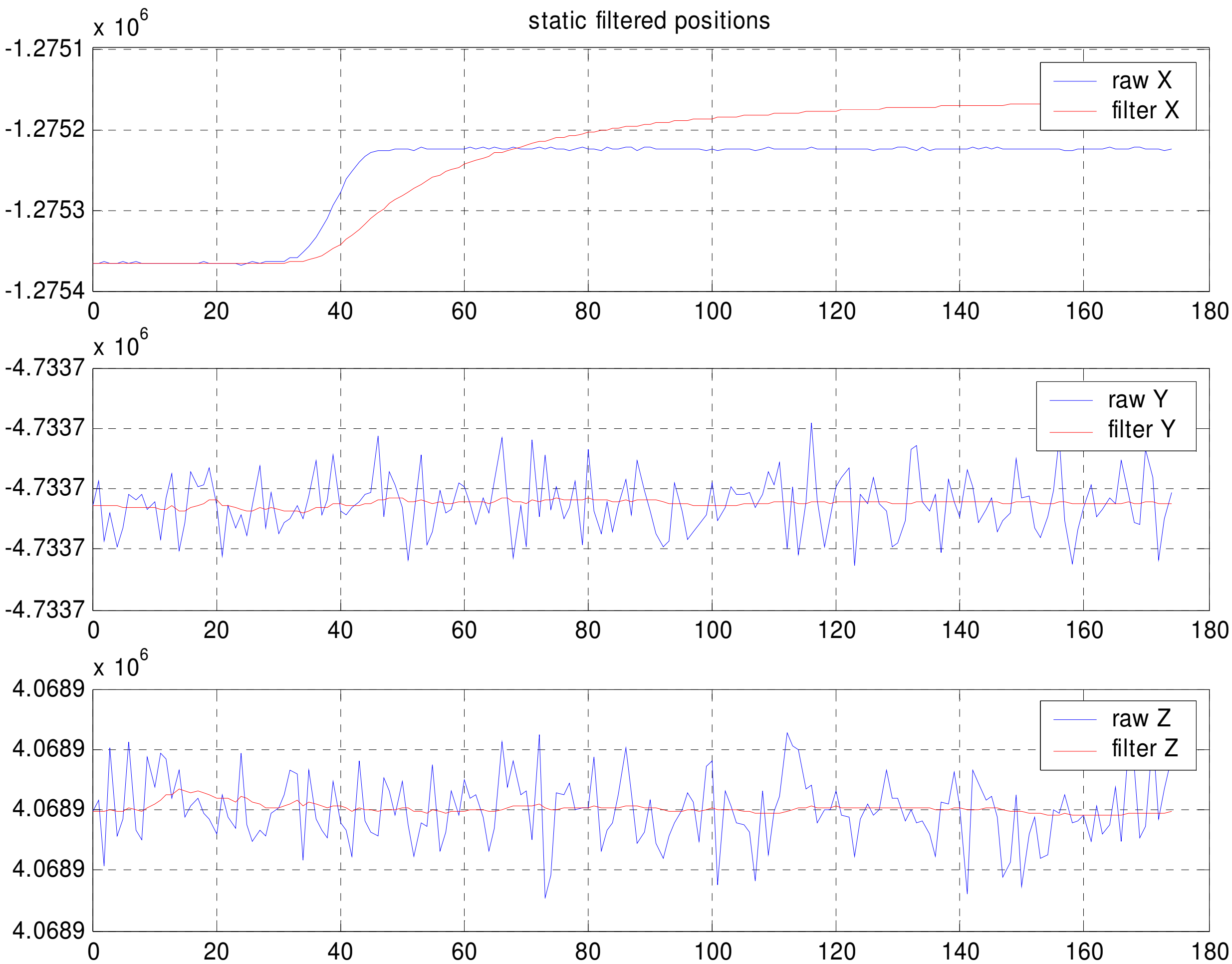
Below is the simulated trajectory in ECEF coordinates. The velocity is estimated from the change in position over each sampling interval T. Also, white noise (variance = 1m) was added to the positions to simulate a 1-sigma error of 1m.



Simulated ECEF position – dynamic trajectory along X axis

Here are the plots for the static operation of the Kalman filter bank (three 2-state filters – constant velocity model). The process noise spectral density (process noise) scalar is set to a constant q = 1e-8 here.

You can see that the amount of filtering is quite large, causing a huge filter delay. The filter is unable to track any dynamics and accelerations, due to the Q scalar being set very low (no acceleration modeled).

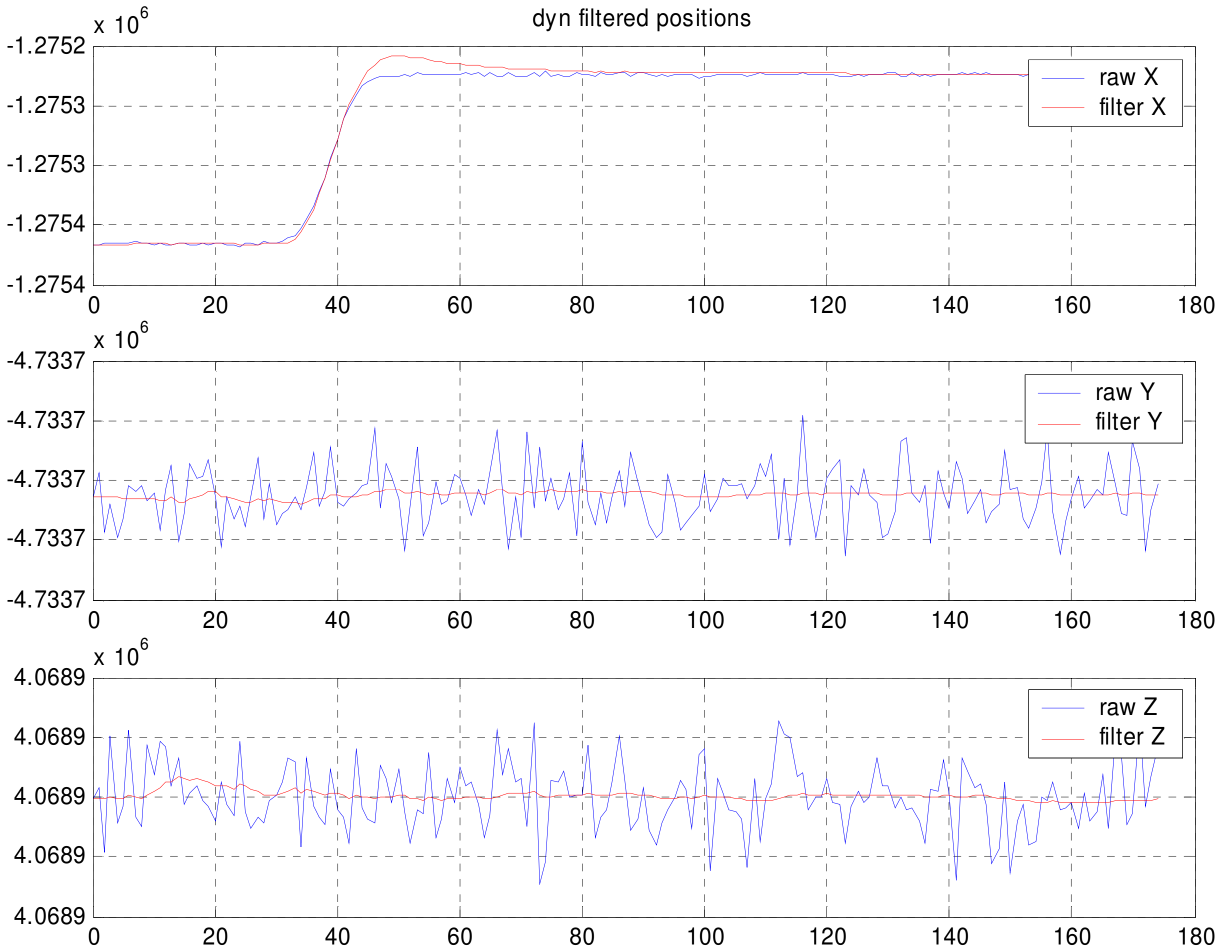


Static Kalman filter vs. truth

The next plot shows the dynamic mode of the Kalman filter. In this filter, the process noise scalar "q" is dynamically changed as a function of estimated acceleration (also known as "gear shifting"). As the acceleration estimate increases, the q constant also increases. The constant is chosen from a heuristic table of values.

This technique is effective since changing the spectral noise scalar will dynamically add more "state noise" into the filter, allowing it to track the unmodeled accelerations. During acceleration, you can see the filter is able to track the position changes quite well.

However, the filter is still suboptimal. It has been shown that just changing the process noise scalar compromises the filter's transient response vs. steady-state response. This is evident in the below plot, where the filter overshoots the true position and gradually converges to the true position over a long period of time. By simply changing the q scalar, you are unable to achieve both fast dynamic response and good steady-state performance. Tuning the filter constant will better the performance to a degree.



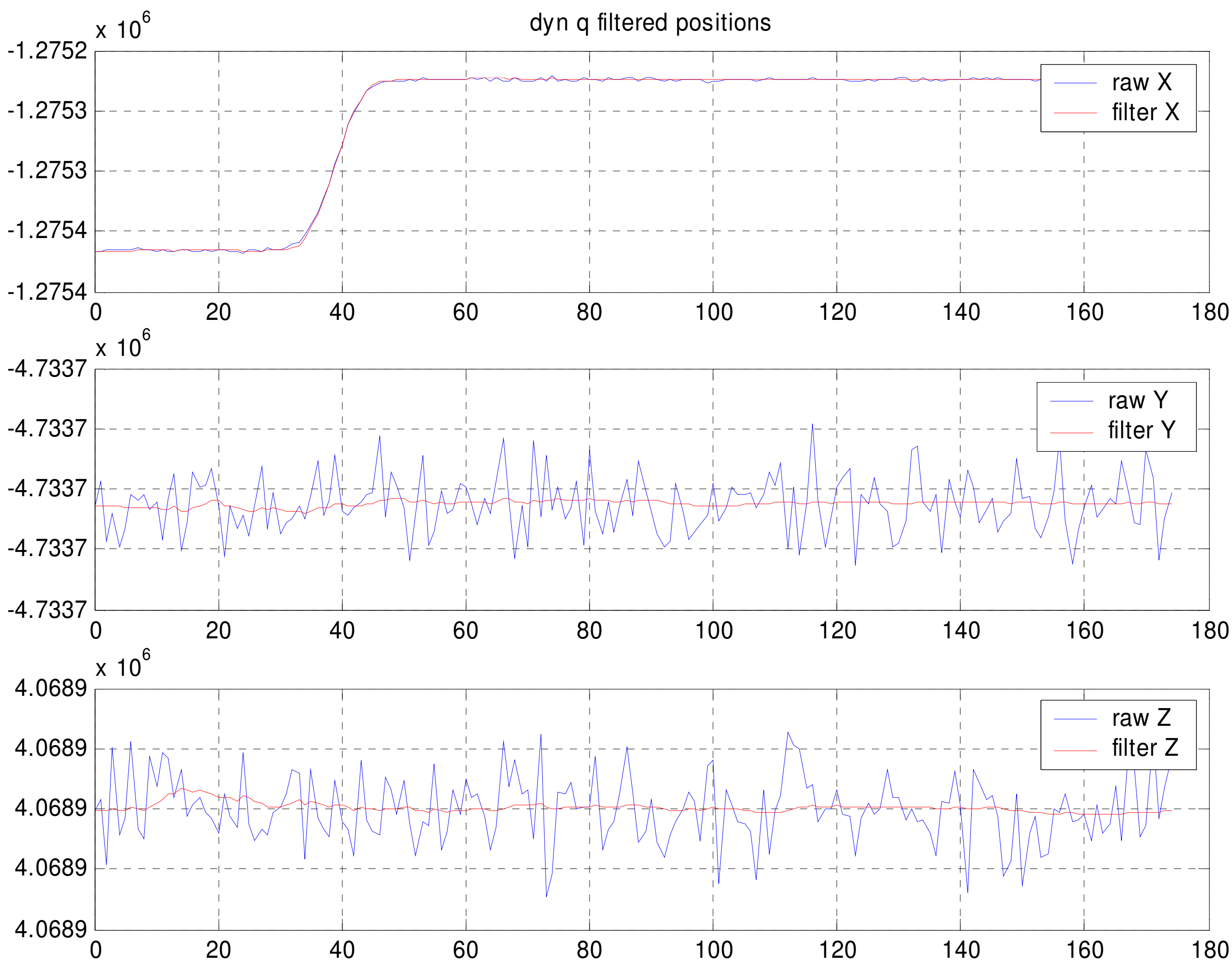
Gear-Shifted Kalman Filter vs. Truth

Sum squared of residuals:
x = 19056.6844, y = 174.1542, z = 181.2787

The next plot shows the dynamic mode of the Kalman filter with dynamic state noise outlined by Hutchinson. In this technique, the added "state noise" is directly injected into the Kalman filter covariance time update. The "best" state noise is a complex function of the estimated acceleration and the cross correlation of the acceleration and the state error.

A time-varying state noise term allows the filter to optimize transient and steady state response. Below is a plot demonstrating the dynamic filter. In this plot, you can see that there is no overshoot when the filter experiences unmodeled acceleration or dynamics. The state noise is adaptively computed from an estimate of the acceleration. In this simulation, the acceleration is found by double differencing the ECEF positions.

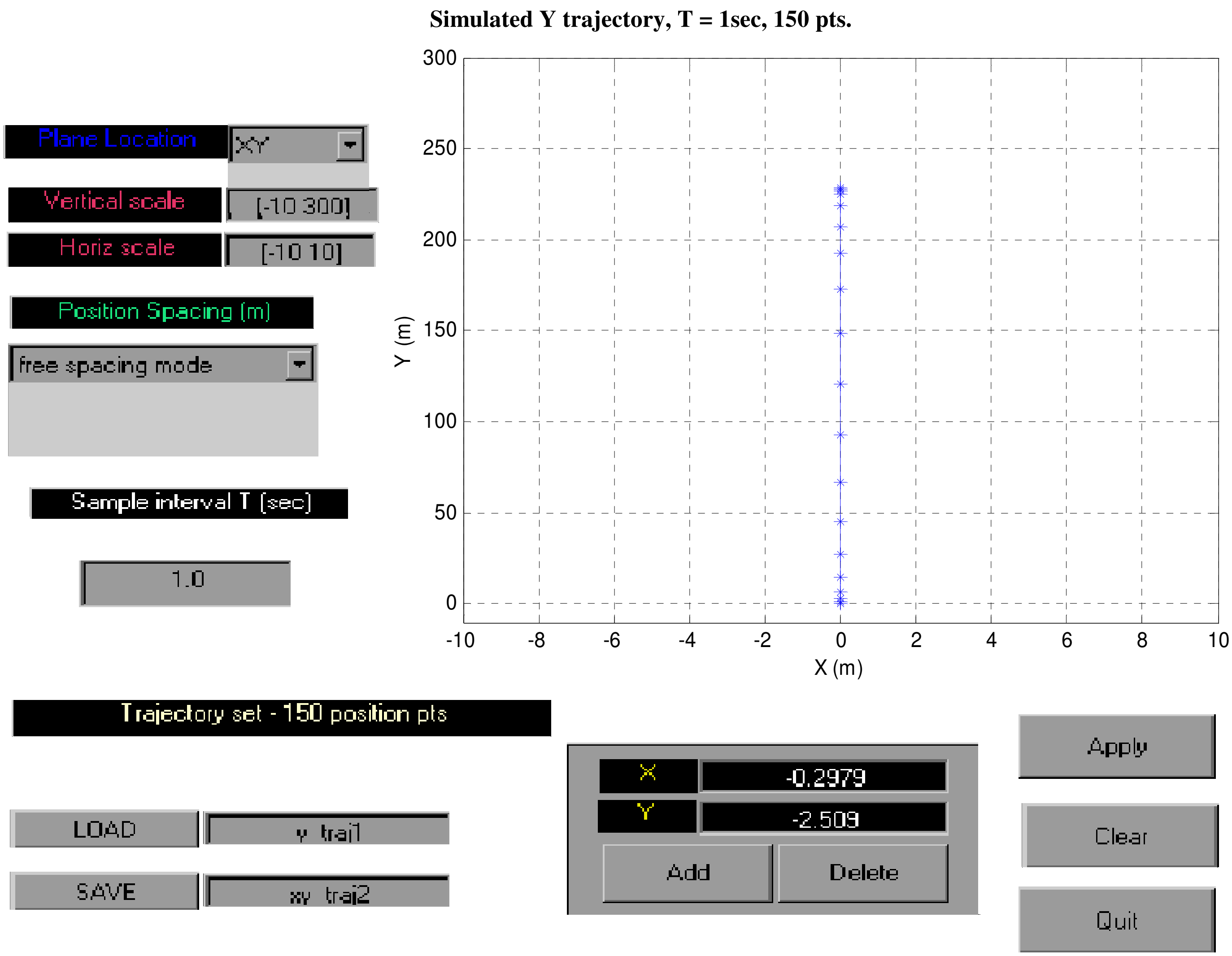
Also note that the filtering on the Y and Z axes are the same for each filter. In the case of no dynamics, the filters use the same q scalar in the noise process matrix (resulting in heavy filtering of the Y and Z measurements).

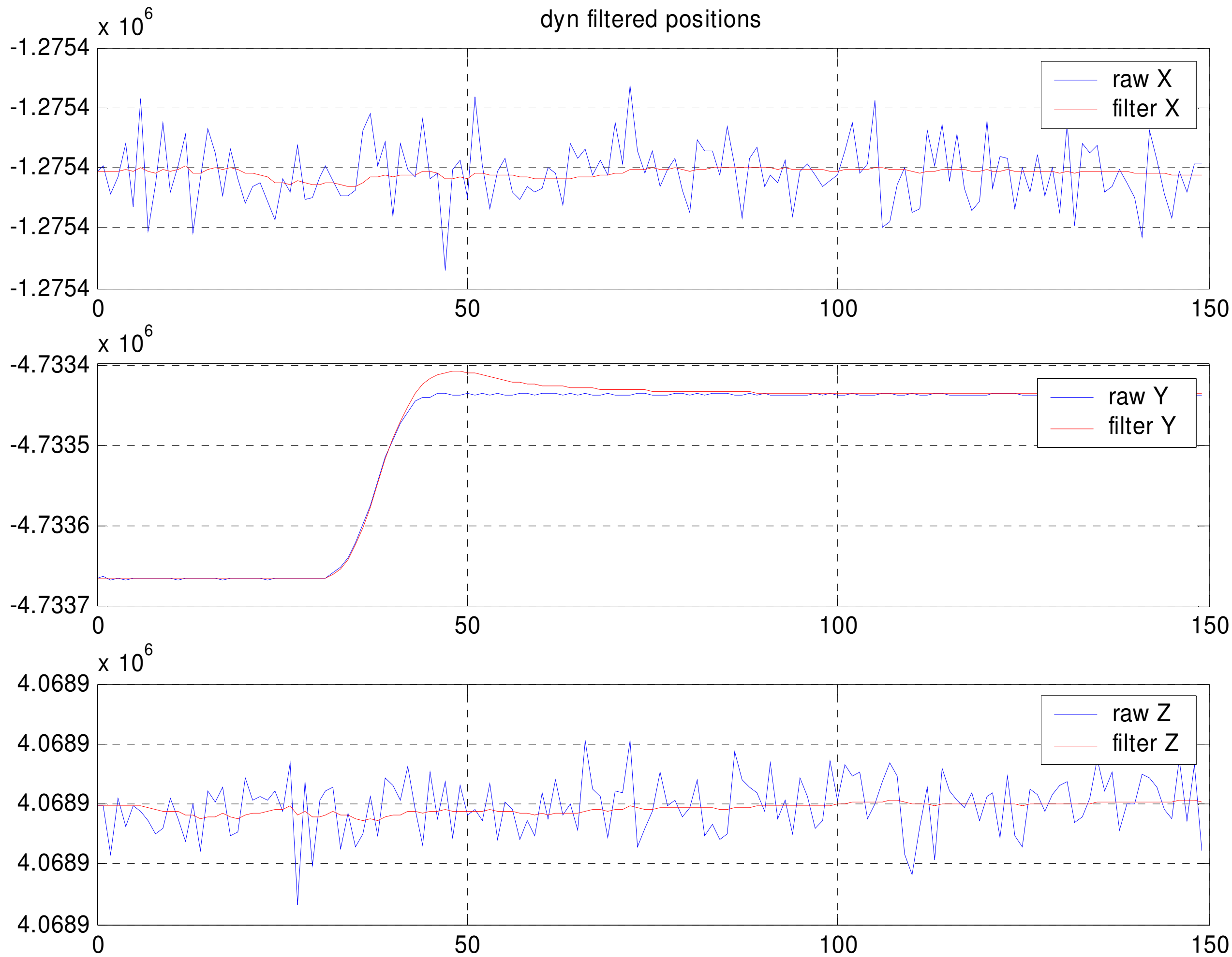


Optimal State Noise Dynamic Kalman Filter vs. Truth

Sum squared of residuals:
x = 478.2861, y = 174.1723, z = 181.271

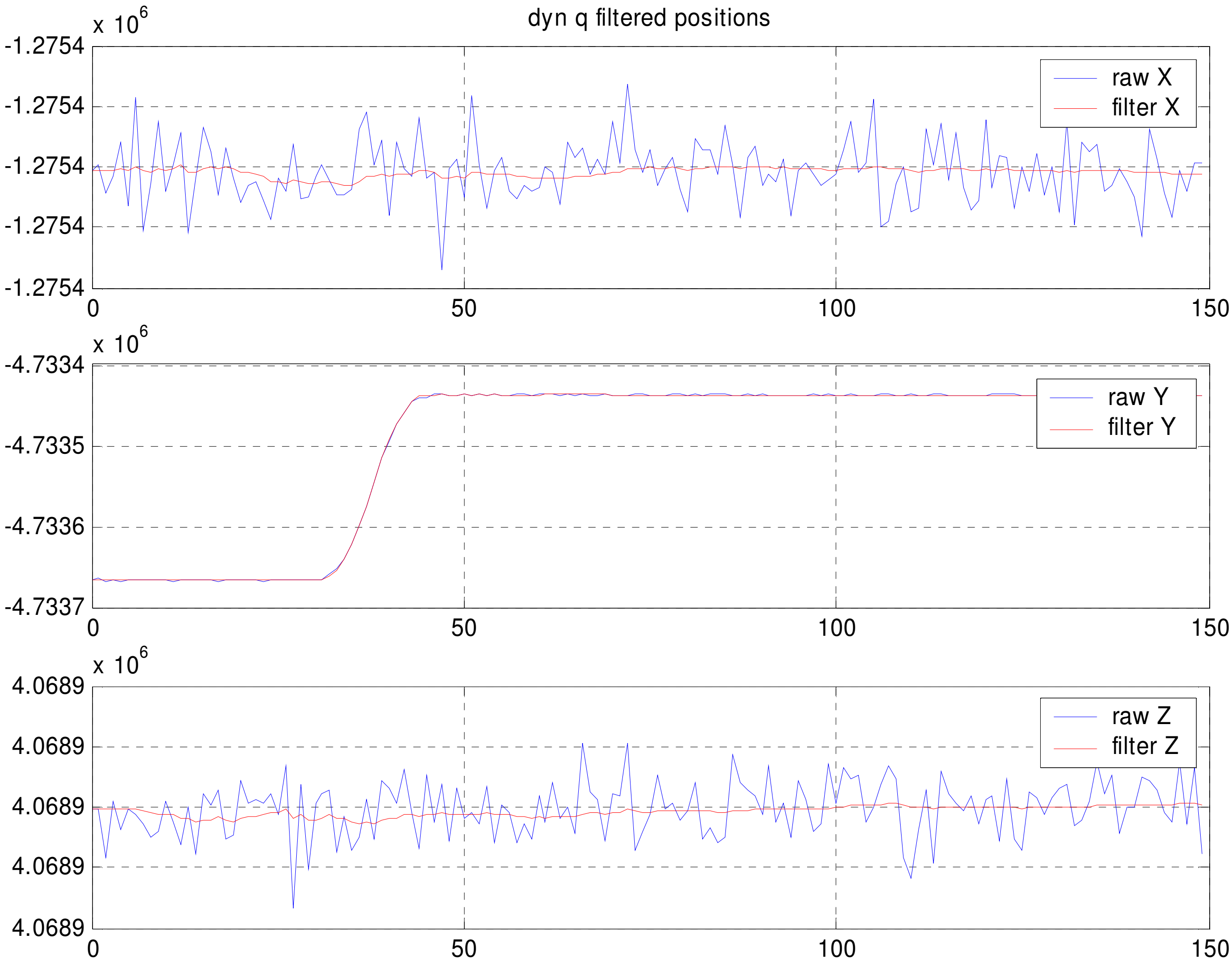
Next shown is a second Y-axis trajectory and the resulting dynamic performance for the 2 competing dynamic Kalman filters.





Gear-Shifted Kalman filter vs. Truth

Sum squared of residuals:
x = 136.9574, **y = 18852.2715**, z = 170.312



Optimal State Noise Kalman filter vs. Truth

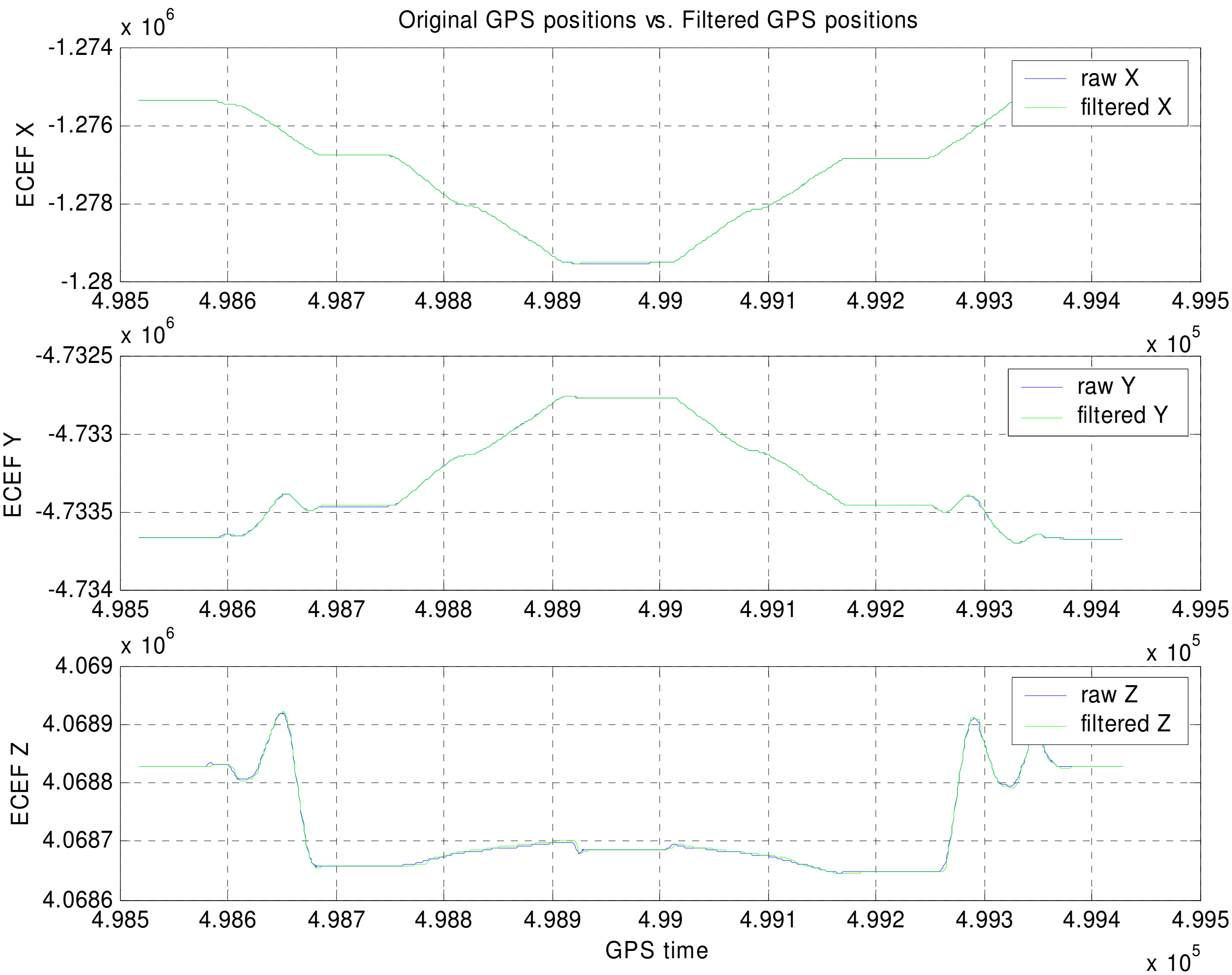
Sum squared of residuals:
x = 136.9648, **y = 678.3746**, z = 170.3039

Again, the optimal state noise filter does not have as much overshoot as the gear shifted method when tracking dynamic targets. This results in a lower MSE of the residuals, which gives a crude global figure of merit for each filter's performance.

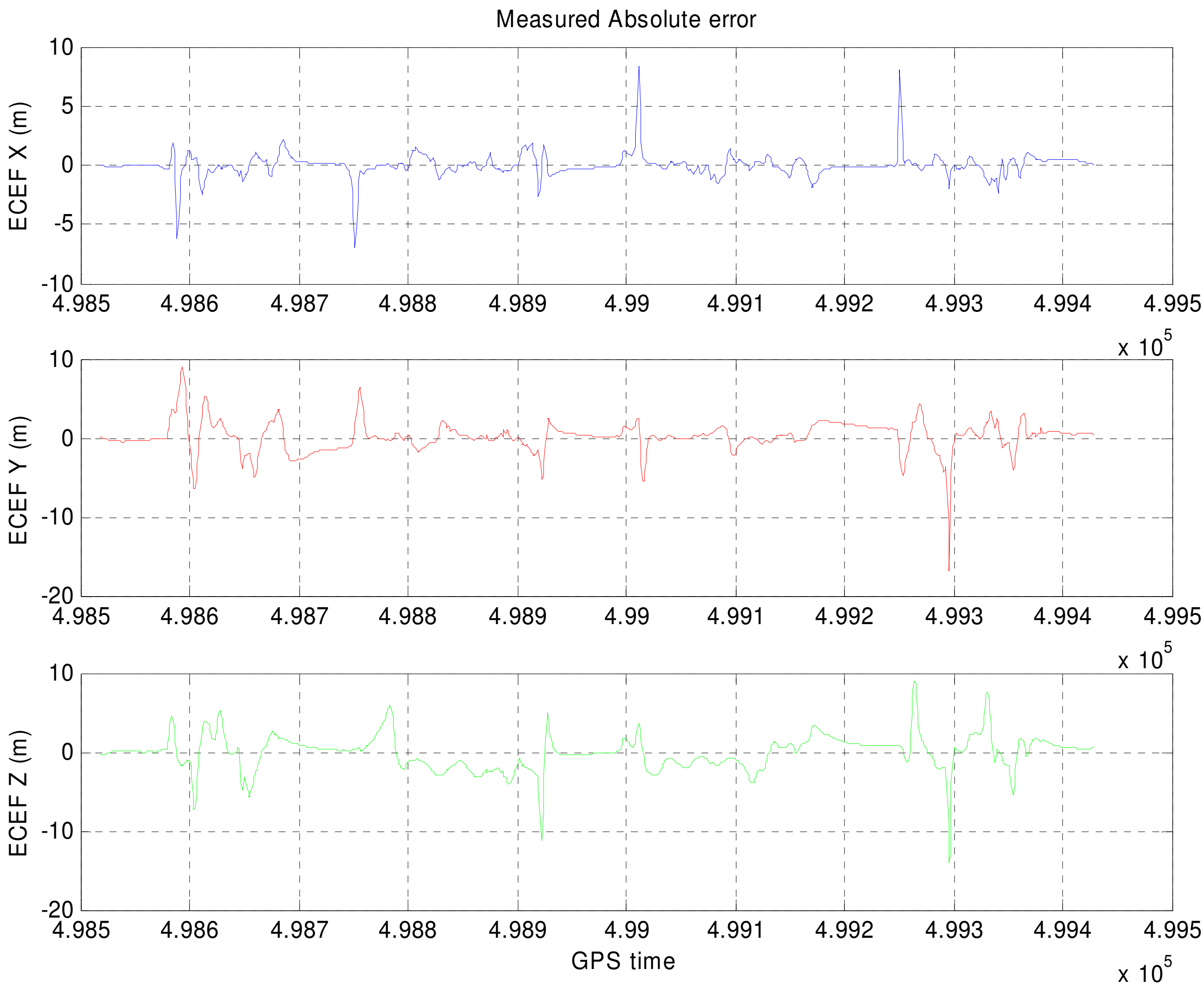
DYNAMIC FILTER TESTS

Next is a driving test to see how the Kalman position filter performs under typical driving dynamics. This is strictly used to determine if the filter behaves well when the vehicle undergoes acceleration. One can also determine how well the filter tracks the GPS position (under dynamics) by examining the filter residuals.

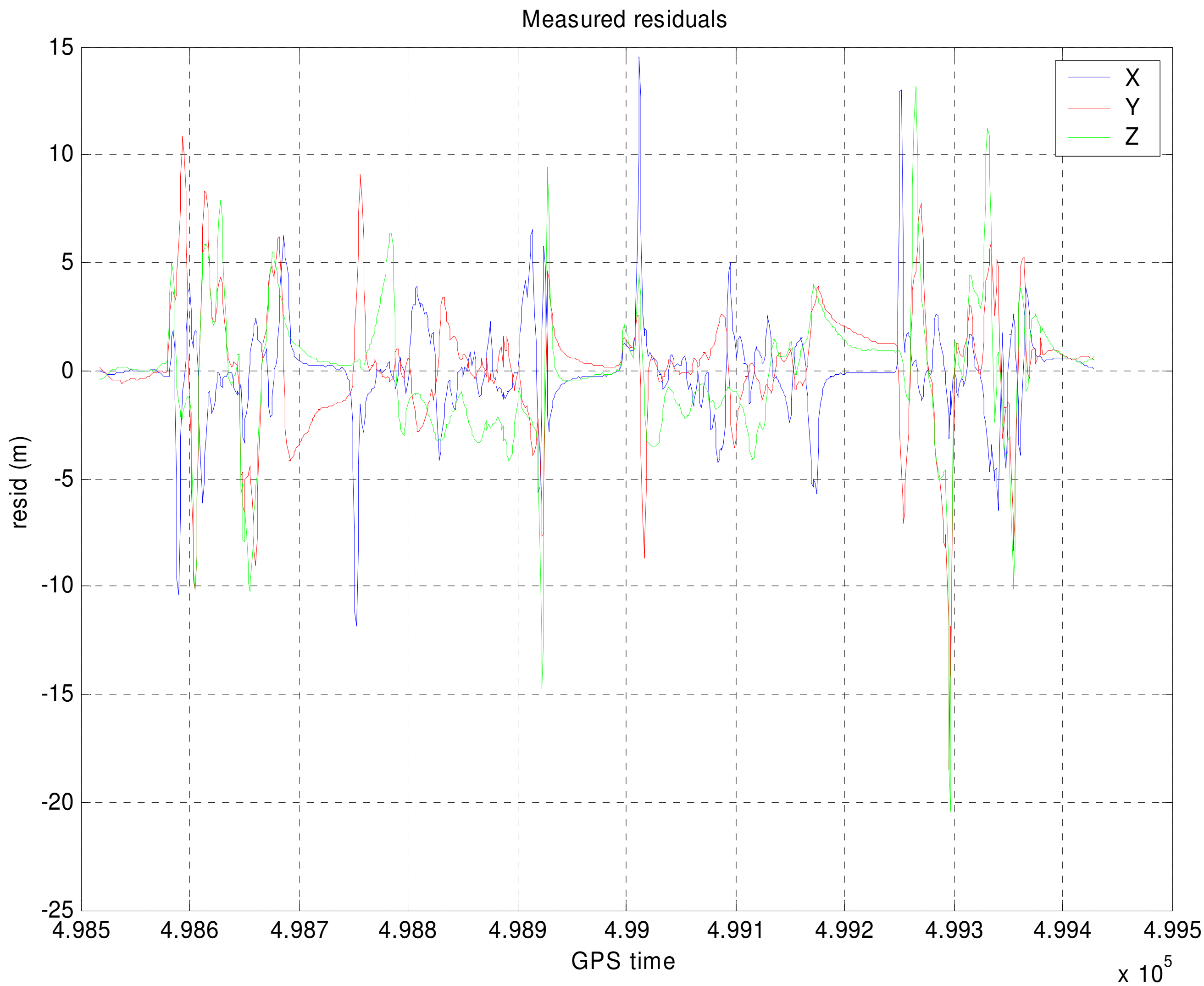
For the data collection, T=1sec. The data was then parsed in MATLAB and plotted. The raw GPS positions were corrected with a WAAS differential source during the test.



Raw GPS Positions vs. Kalman Filtered Positions in Dynamic Environment with Optimal State Noise Technique



Absolute Error between GPS positions vs. Dynamically Filtered Positions



Bank of Three 2nd-order MVRO Kalman Filter Residuals
Operating on GPS ECEF Position Fixes (X,Y,Z)

As a figure of merit, the MSE of the residuals were again computed. This gives a global figure on how the filter is tracking over time.

Sum squared residuals: Measured optimal state noise technique:
x = 4232.5479, y = 7544.0457, z = 8838.664

Sum squared of residuals: MATLAB simulated "gear shifted" q scalar technique:
x = 593930.5214, y = 124532.4209, z = 190053.8286

Sum squared of residuals: MATLAB simulated optimal state noise technique:
x = 8254.4672, y = 7966.465, z = 11296.756

You can see that the measured optimal state noise technique gives superior results in tracking maneuvers in a dynamic environment. The actual measured figures even come in lower than the simulated numbers in

MATLAB. This is because the simulation does not have updated filter parameters at every filter iteration, thus making the simulated filter perform worse when compared to the actual real-time firmware filter.

Further testing is necessary to fully characterize the dynamic performance of the reduced-order filter in Ag-related applications. Filter tuning for peak performance will be a key aspect of this further research.

REFERENCES

1. Blackman, S., Multiple-Target Tracking with Radar Applications, Artech House, 1986.
2. Singer, R.A., “Estimating Optimal Tracking Filter Performance for Manned Maneuvering Targets,” *IEEE Transactions of Aerospace and Electronic Systems*, AES-5, July 1970, pp. 473-483.
3. Hutchinson, C.E, J.A. D’Appolito, K.J. Roy, “Application of Minimum-Variance Reduced-State Estimators,” *IEEE Transactions on Aerospace and Electronic Systems*, AES-11, Sept. 1975, pp. 785-794.