Appendix A
Introduction to Kalman Filtering of Time-Series Data

A.1 Introduction

The topic of Kalman filtering has a vast associated literature (Brookner 1998; Brown and Hwang 1997), so the simple overview in this Appendix can only touch on some on the more important aspects used in this book. Thus the purpose of this Appendix is to present the basic mathematical underpinnings of Kalman filtering, and to present the method of Kalman filter design appropriate for particular applications.

Because location systems have inherent errors in determining positions, methods of improving the accuracy are an important part in the overall design of a system. One of the simplest methods of removing errors in positional data is to simply smooth the raw positional data in real time. This concept is based on the observation that moving objects have finite dynamics, while positional (particularly radiolocation) data can result in large changes in position between successive position updates. Under these circumstances it is intuitive that some form of data smoothing will result in a measured track which more closely represents the actual track than the raw data. However, smoothing the raw data to minimize the noise will also distort the desired positional data track of a moving object being tracked. Thus there needs to be a compromise between reducing the noise component while having minimal effect on the tracking of positions. Clearly the optimum solution will depend on the dynamics of the object being tracked—a tracked person will have much more agile dynamics then (say) a motor vehicle. The most popular method of such data smoothing is the Kalman Filter (the subject of this Appendix) which provides the optimum compromise solution for the given specified dynamics and noise characteristics.

The data associated with position fixing will be corrupted with random errors. To minimise the effect of these errors, a Kalman filter is applied to the measured or computed data. Reducing the bandwidth of the filter will certainly reduce the noise

I. Sharp and K. Yu, *Wireless Positioning: Principles and Practice*, Navigation: Science and Technology, https://doi.org/10.1007/978-981-10-8791-2

(variance), but due to the dynamics of the data (changes in the position as well as velocity and acceleration changes) the filtering will also introduce errors in noise-free data. To minimise the overall errors, the parameters of the Kalman filter must be optimised for the dynamics appropriate to the particular time-series data of the application. The order of the filter is also important in minimising the errors. For a third-order state vector with $[x, x', x'']$, the filter will track with zero nominal error both the position x and speed x' components for a constant acceleration x''. As both position and speed estimates are important for position location applications, a third order (or g-h-k) filter will be described in a later section.

The following Sect. A.2 provides details of the implementation of the Kalman filter. The method closely follows that given in the book "Tracking and Kalman Filtering Made Easy", by Eli Brookner. The Kalman filter algorithms are presented in matrix form without proof. The elegant nature of this form is that the basic algorithm is the same for all filters, regardless of the details of the particular application. However, the matrices must be configured for each case. The details for the typical application are given in Sect. A.3 following.

A.2 Matrix Representation

The matrix representation of the Kalman filter is summarised in this section. The aim of the algorithm is to optimally filter a state vector **X**, based on measured input data **Y**. Estimated variables are indicated by an asterisk (*) superscript. It is assumed that the measurements and the state vector are updated at a regular fixed period T (s), and samples are indicated by a subscript n. Typical update periods for tracking vary from 0.1 to 1 s for people, and 1 to 5 s for vehicles.

There are two types of uncertainty in the measured input data **Y**. The first type of uncertainty (process noise) is associated with the dynamics of the particular application. In the case of a tracking system, this uncertainty is associated with the motion of the mobile device; such motion is modelled as a random process. The process noise, specified by the matrix **Q**, is defined as an input (often assumed invariant) to the Kalman filter. The nature of the matrix is dependent on the particular application and its dynamics. See Sect. A.3 for more details.

The second type of uncertainty is associate with **Y** measurement errors (measurement noise). Such noise is assumed to be random (typically Gaussian), and are usually uncorrelated with the process noise. As well as a smoothed estimate of the state vector, the Kalman filter also outputs an estimate of the measurement noise, based on the input data. In a positioning system this can be roughly interpreted as the "accuracy" of the position fixes.

The Kalman filter uses two types of estimates, namely a priori estimates of the state vector (and its associated covariance matrix), and a posteriori estimates of the same parameters. The a priori estimate occurs just before the new data are available at time period n, and can be considered as a prediction of the state vector for time n. The a posteriori estimate at time n is based on the latest data (at time n), and can be

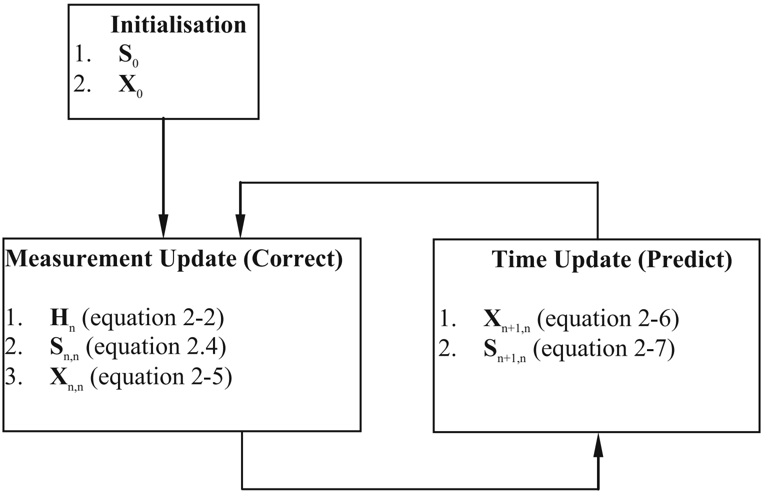

Fig. A.1 Block diagram of the Kalman filter process, showing the cyclical loop used for calculations

considered as a update to the estimate. These two groups of calculations are performance in a cyclical fashion, with in input of one set coming from the output of the second set. This cyclical process is shown in more detail in Fig. A.1.

A.2.1 Initialization

The filter is initialized with an estimate of the state vector $\mathbf{X_0}$ (a priori estimate). This initial value is not too critical, as the filter dynamics will soon track the error to their steady-state values. In typical practical cases, the measurement vector and the state vector may not be of the same order. For example, a positioning system measures the position only, whereas the state vector may also include the speed and acceleration. In some cases, with other sensors, these two elements of the state vector may also be measured by other sensors (for example accelerometers), but this is not essential for implementation of the Kalman filter.

Thus for time period n, the measurements $\mathbf{Y_n}$ are related to the state vector by

$$\mathbf{Y}_n = \mathbf{M}\mathbf{X}_n + \mathbf{N}_n \tag{A.1}$$

In (A.1), $\mathbf{N}$ is the random measurement noise component which corrupts the measurements. The $\mathbf{M}$ vector is applied (if necessary) to convert the measurement data to match the state vector; for example if the measurements are just the position x but the state vector is $[x, x', x'']^T$, then $M = [1, 0, 0]$.

The other a priori input is the initial estimate of the error matrix $\mathbf{S}^*$. The matrix $\mathbf{S}^*$ is associated with the accuracy in predicting the state vector. Mathematically $\mathbf{S}^*$ is related to the covariance of the predicted state vector $\mathbf{X}^*$, and is updated by the Kalman filter during the filtering process. When initialising the **S** matrix, if there is no other information, all the elements of the matrix can be set to zero. The Kalman filter will update the matrix as measured data become available. However, typically initial values of the matrix can be estimated from approximate knowledge of the errors associated with the measurements and the state vector dynamics.

A.2.2 Measurement Update (Correction)

The first set of calculations is associated with the update of the Kalman loop parameters, based on the a priori data. The first step is to define the filter. These parameters are similar in nature to the parameters of a g-h-k filter (see Sect. A.3), except that in the Kalman filter these parameters vary according to the characteristics of the input measured data (and the associated noise). These parameters are defined by a **H** matrix. The **H** parameter vector can now be updated as defined in (A.2)

$$\mathbf{H}_n = \mathbf{S}^*_{n,n-1}\mathbf{M}^T[\mathbf{R} + \mathbf{M}\mathbf{S}^*_{n,n-1}\mathbf{M}^T]^{-1} \tag{A.2}$$

The dual subscripts have the following interpretation: the first index indicates the time index for which the calculated value applies, and the second index defines the time index of the data used for the calculation. Equation (A.2) introduces a new matrix associated with the Kalman filter. The **R** matrix is associated with the measurement noise, and is defined by the following covariance matrix

$$\mathbf{R}_n = \mathrm{cov}[\mathbf{N}_n] = E[\mathbf{N}_n\mathbf{N}_n^T] \tag{A.3}$$

The next update is associated with determining the a-posteriori estimate of the **S** matrix. This updating process is described mathematically by the matrix equation

$$\mathbf{S}^*_{n,n} = [\mathbf{I} - \mathbf{H}_n\mathbf{M}]\ \mathbf{S}^*_{n,n-1} \tag{A.4}$$

The filtered state vector is now updated, based on the updated **H** matrix and the input measured data **Y**

$$\mathbf{X}^*_{n,n} = \mathbf{X}^*_{n,n-1} + \mathbf{H}_n[\mathbf{Y}_n - \mathbf{M}\mathbf{X}^*_{n,n-1}] \tag{A.5}$$

A.2.3 Time Update (Prediction)

The time update processing (or predictions) use the measurement updates to determine the a priori (predicted) parameters. First, the predicted state vector is given by

$$\mathbf{X}_{n+1,n} = \mathbf{\Phi}\,\mathbf{X}_{n,n} \tag{A.6}$$

The $\mathbf{\Phi}$ matrix is the predictor matrix, and is based on the system dynamics model appropriate for the particular case. The $\mathbf{\Phi}$ matrix allows the state vector to be updated (predicted) one time period.

The second a priori update is associated with the **S** matrix.

$$\mathbf{S}^*_{n+1,n} = \Phi \mathbf{S}^*_{n,n} \mathbf{\Phi}^T + \mathbf{Q} \tag{A.7}$$

The **Q** matrix is defined as the covariance matrix of the random system dynamics noise (process noise), and is given by

$$\mathbf{Q} = \mathrm{cov}[\mathbf{U}_n] = E[\mathbf{U}_n \mathbf{U}_n^T] \tag{A.8}$$

The filter algorithm is looped repeatedly by processing Eqs. (A.2)–(A.8) to obtain a prediction of the state vector **X** at time $n+1$, based on data **Y** available up to time n.

A.3 Kalman Filter for Position Determination Applications

The general theory presented in Sect. A.2 is now applied for a typical position determination application. The main requirement is usually associated with filtering the estimated position (x, y) of the mobile device. It is assumed that the x and y coordinates are statistically independent, so that the Kalman filter is applied to x and y separately. Each tracking application will have well defined dynamics. For example, tracking people will have a time constant of the order of a second, as the stride rate is typically about 1.5 per second, and the direction can change in a couple of strides. The main period of acceleration is at the beginning of walking, with low acceleration when walking in a straight line. However, direction change can occur at unpredictable times, so that the accelerations should be modelled as a random process with some maximum associated acceleration. For accurate tracking and predictions, it is essential that the position and speed are determined accurately without lag. Thus a third-order filter is appropriate, as this filter type will have zero steady-state lag for constant acceleration.

When tracking the individual x and y coordinates the situation is a little more complicated. In this case, even if the speed is constant, there will be accelerations in both x and y around bends. These accelerations are termed "pseudo-accelerations", as they are a consequence of the geometry and the selected variables to be tracked, and are related only indirectly to the real mobile accelerations. (An alternative set would be in polar coordinates, where both r and θ vary relatively slow, even when in a bend).

For the x or y coordinates, the pseudo-accelerations for a oval track[1] are readily calculated. In particular, assuming an oval track with semi-circular bends, the pseudo-speed and pseudo-acceleration are of the form

$$\begin{aligned} V(t) &= -V_0 \cos\left(\frac{V_0 t}{r}\right) \\ A(t) &= \frac{V_0^2}{r} \sin\left(\frac{V_0 t}{r}\right) \end{aligned} \tag{A.9}$$

where r is the radius of the bend, and V_0 is the mobile speed, assumed constant. As both these parameters can be approximately defined for a given application, the peak acceleration can be specified for the Kalman filter dynamics. For example, if $r = 1$ m and $V = 1.5$ m/s (typical for a person walking), then the peak pseudo-acceleration is 2.25 m/s^2.

The proposed implementation of the Kalman filter is a simplification of the Singer g-h-k Kalman filter described in Sect. 2.9 of the above-referenced book. This filter has a number of components in defining the process noise statistics (probability density function) that can be tuned to a wide variety of applications, including walking. The PDF is illustrated in Fig. A.2, which shows that there are three delta functions plus a uniform distribution. The delta function at zero acceleration is associated with no movement or moving at a constant speed and direction. The two delta functions at the maximum acceleration are associated with moving around a bend, as described above. The uniform distribution is associated with other random motions which cannot be described explicitly, but are assumed to cover a wide range of accelerations up to the maximum associated with turning a tight bend. The relative magnitude of there components can be estimated by simulation or actual measurements. Figure A.3 shows the PDF of movement around an oval track at constant speed; as can be observed, the resulting PDF of the acceleration approximates the assumed model of the PDF. The PDF has a dominant peak at the origin (acceleration is assumed zero on straight-line segments), and approximately a constant distribution up to the maximum acceleration defined by (A.9). The Singer-Kalman filter parameters for the human walking application are assumed to be as follows: $P_0 \approx 0.4$, $P_{\max} = 0$, $A_{\max} = 2.25$ m/s^2. Note that it is assumed that people spend a large proportion of time when walking with zero acceleration.

[1]An oval track is an appropriate model as it includes both straight-line segments and constant radius bends, which approximate the types of path segments for people walking around a building.

This filter has an exponential autocorrelation function given by

$$E[\ddot{x}(t)\ddot{x}(t+t')] = \sigma_a^2 e^{-|t'/\tau|} \quad (A.10)$$

The time constant τ of the acceleration can be computed from the autocorrelation function of the pseudo-acceleration around a path. For the Kalman filter to be effective the data sampling period (T) should be much smaller than the dynamics time constant. As a guide it is recommended that $T < \tau/10$. For this Singer distribution, the variance can be computed to be

$$\sigma_a^2 = \frac{A_{\max}}{3}[1 + 4P_{\max} - P_0] \quad (A.11)$$

For any given application, the above parameters can be computed, thus optimising the solution for each track. For example, while the above parameters are appropriate for people walking, the same model could be used for tracking vehicles, but with different parameters. However, race cars on a track will be different from cars on suburban streets, but the same model is broadly appropriate.

The measured data input to the filter are (separately) the computed x and y positions from the position location system. The corresponding state vector includes the position, speed and acceleration. (In the following, only the x coordinate is used; the y coordinate analysis is identical).

Thus the state noise and observation matrices are

$$\mathbf{X}_n = \begin{bmatrix} x_n \\ \dot{x}_n \\ \ddot{x}_n \end{bmatrix} \qquad \mathbf{U}_n = \begin{bmatrix} u_{x_n} \\ u_{v_n} \\ u_{a_n} \end{bmatrix} \qquad \mathbf{M} = [1 \quad 0 \quad 0] \quad (A.12)$$

As the system dynamics parameter satisfies $\tau \gg T$, the transition matrix $\mathbf{\Phi}$ reduces to that associated with constant acceleration, namely

$$\mathbf{\Phi} = \begin{bmatrix} 1 & T & T^2/2 \\ 0 & 1 & T \\ 0 & 0 & 1 \end{bmatrix} \quad (A.13)$$

The covariance of the white-noise manoeuvre excitation vector $\mathbf{U}_\mathrm{n}$ can be shown to be

$$\mathbf{Q} = \frac{2\sigma_a^2}{\tau} \begin{bmatrix} \frac{T^5}{20} & \frac{T^4}{8} & \frac{T^3}{6} \\ \frac{T^4}{8} & \frac{T^3}{3} & \frac{T^2}{2} \\ \frac{T^3}{3} & \frac{T^2}{2} & T \end{bmatrix} \quad (A.14)$$

The predictor covariance matrix $\mathbf{S}$ is initialised as follows

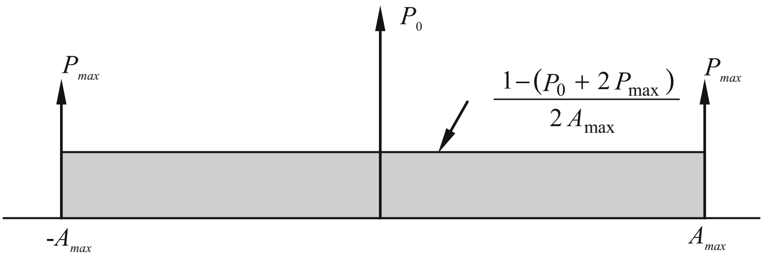

Fig. A.2 Generic PDF for Singer-Kalman filter

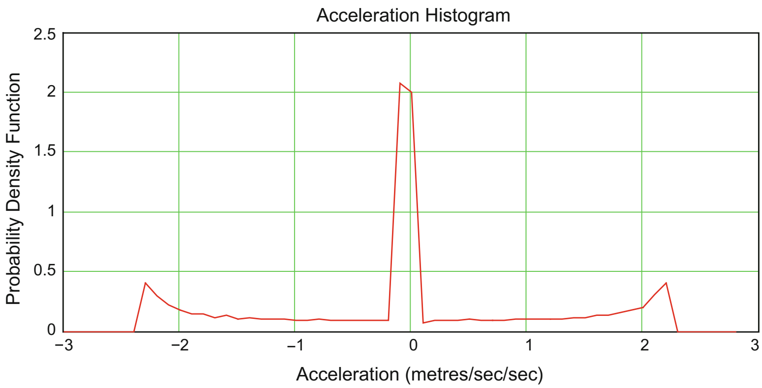

Fig. A.3 Probability Density Function of the acceleration for constant speed around an oval path

$$\mathbf{S}_0 = \begin{bmatrix} \sigma_x^2 & \frac{\sigma_x^2}{T} & 0 \\ \frac{\sigma_x^2}{T} & \frac{2\sigma_x^2}{T^2} & \tau\sigma_a^2 \\ 0 & \tau\sigma_a^2 & \sigma_a^2 \end{bmatrix} \tag{A.15}$$

where σ_x is the measurement accuracy (standard deviation) of the x measurements. By applying these Kalman filter matrix definitions to the generic algorithm given in Sect. A.2, the optimum filtering of the positional data can be obtained. The Kalman filter is implemented using the generic matrix equations described in Sect. A.2, and the specific definitions of the matrices described above.

A.4 G-H-K Filter Equivalent

The Kalman filter implementation defined in Sect. A.2 uses matrix-based algorithm to implement the filter. While the algorithm is relatively simple when expressed in matrix algebra form, the requirement for floating-point matrix calculations can be intensive in computation, particularly if the simple processors in Wireless Sensor Networks are used. Thus an alternative method is sought which requires less computational resources, yet retains the benefits of the Kalman filter. In this section, the Kalman filter is shown to be equivalent (with some minor constraints) to a g–h–k filter, which can be implemented as a classical finite impulse response (FIR) filter, which only requires a few multiplications and additions for its implementation.

A g–h–k filter is based on defining the motion differential equations assuming constant acceleration. It is shown in the cited reference that this filter is defined by three parameters $(g, h \text{ and } k)$, and three differential equations for the position, speed and acceleration. Importantly, the (g, h, k) parameters can be defined in terms of the Kalman **H** matrix, and thus the Kalman filter design can be transformed into a g–h–k filter. An overview of the key results will now be presented.

The Kalman filter described in Sect. A.3 approaches a steady-state g-h-k filter, provided the process noise (**Q**) and the measurement noise (**R**) are constant matrices (which is normally the case in the typical applications). In this case the third-order filter parameters (g, h, k) are defined by the **H** matrix as follows

$$\mathbf{H} = \begin{bmatrix} g \\ h/T \\ 2k/T^2 \end{bmatrix} \tag{A.16}$$

Additionally, the filter parameters obey the following conditions

$$\begin{aligned} g &= \sqrt{2h} - h/2 \\ h &= 4 - 2g - 4\sqrt{1-g} \\ k &= \frac{h^2}{4g} \end{aligned} \tag{A.17}$$

Note that equations (A.17) are in fact only two conditions, as the first two are two different forms of the one condition. Thus these equations are not sufficient to fully define these parameters. One parameter (typically g) is determined from the standard deviation of the acceleration (σ_a) and the standard deviation of x (σ_x). In practice, the following condition always applies: $g \gg h \gg k$.

The prediction equations for the g-h-k filter with constant acceleration are given by

$$\begin{aligned} x^*_{n+1} &= x^*_n + g\,(y_n - x_n) + T\dot{x}^*_n + h\,(y_n - x_n) + \frac{T^2}{2}\ddot{x}^*_n + k\,(y_n - x_n) \\ \dot{x}^*_{n+1} &= \dot{x}^*_n + \frac{h}{T}(y_n - x_n) + T\ddot{x}^*_n + \frac{2k}{T}(y_n - x_n) \\ \ddot{x}^*_{n+1} &= \ddot{x}^*_n + \frac{2k}{T}(y_n - x_n) \end{aligned} \tag{A.18}$$

Equations (A.18) can be converted into the more convenient form of the z-transform of the filter transfer function by taking the z-transform of the equations. Taking the z-transform of equations (A.18) yields

$$
\begin{aligned}
zX_n &= X_n + g\,(Y_n - X_n) + T\dot{X}_n + h\,(Y_n - X_n) + \frac{T^2}{2}\ddot{X}_n + k\,(Y_n - X_n) \\
z\dot{X}_n &= \dot{X}_n + \frac{h}{T}(Y_n - X_n) + T\ddot{X}_n + \frac{2k}{T}(Y_n - X_n) \\
z\ddot{X}_n &= \ddot{X}_n + \frac{2k}{T}(Y_n - X_n)
\end{aligned}
\tag{A.19}
$$

By substituting for $\dot{X}_n$ and $\ddot{X}_n$ in the first equation in (A.19), the z-transform of the transfer function can be determined after much algebraic manipulation

$$
H(z) = \frac{X_n}{Y_n} = \frac{(g+h+k)\,z^2 + (k-2g-h)\,z + g}{z^3 + (g+h+k-3)\,z^2 + (k-2g-h+3)\,z + (g-1)}
\tag{A.20}
$$

Now that the Kalman filter implementation has been reduced to a z-transform transfer function, the filter can be implemented using the standard techniques for FIR filters. As the polynomial in the denominator of the transfer function is a cubic, the FIR filter will require samples with a delay of up to three samples.

The frequency response defined by Eq. (A.20) can be determined by substituting $z = \exp(j\omega)$. Thus at $\omega = 0$, it is easy to show that $H(0) = 1$. Similarly, at the maximum scaled frequency of $f = 0.5$, the magnitude of the transfer function is given by

$$
|\,H(0.5)\,| = a = \left|\frac{2g+h}{2g+h-4}\right| \approx \frac{g}{2}
\tag{A.21}
$$

which is the attenuation (a) of the filter at high frequencies, and thus defines the degree of attenuation of the high-frequency noise. Thus for good attenuation g should be small, but g also largely determines the filter bandwidth. This compromise is one of the essential characteristics of the design of Kalman filters.

The filter delay is another important parameter, and ideally for accurate tracking the delay should be zero for all input signals. A filter group delay is given by

$$
\Gamma(\omega) = \frac{d\Phi(\omega)}{d\omega} \qquad \Phi(\omega) = \arg\left[H\left(e^{j\omega}\right)\right]
\tag{A.22}
$$

where $\Phi(\omega)$ is the phase of the transfer function. With the transfer function given by Eq. (A.21), Eq. (A.22) can be evaluated at $\omega = 0$ (or $z = 1$), which shows that the group delay is indeed zero at zero frequency. Thus input signals with low spectral bandwidth can be tracked with essential zero error.

A typical example is shown in Fig. A.4, based on the values determined by the Singer-Kalman filter (standard deviation in range 1 m. Note that the initial filter gain of 0 dB increases to a peak near the motion dynamics bandwidth parameter $(1/\tau)$, and then falls to about −20 dB at the maximum frequency of 5 Hz (Nyquist frequency for sampling rate of 10 per second assumed in this example). Thus as expected noise with frequencies greater than the motion dynamics bandwidth are significantly filtered, while frequencies below the motion dynamics bandwidth are largely unfiltered.

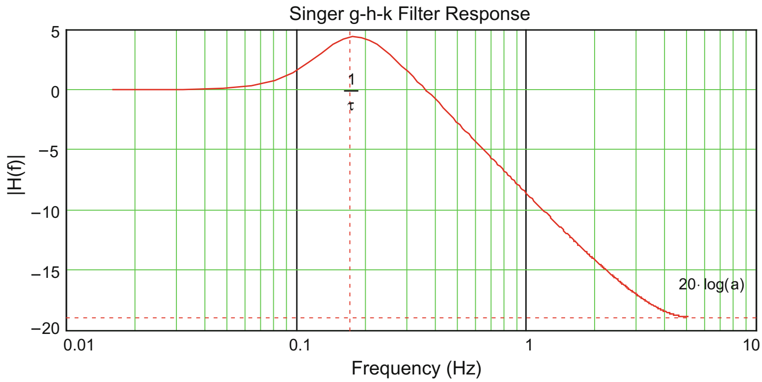

Figure A.4. Example of the transfer function of the g-h-k tracking filter

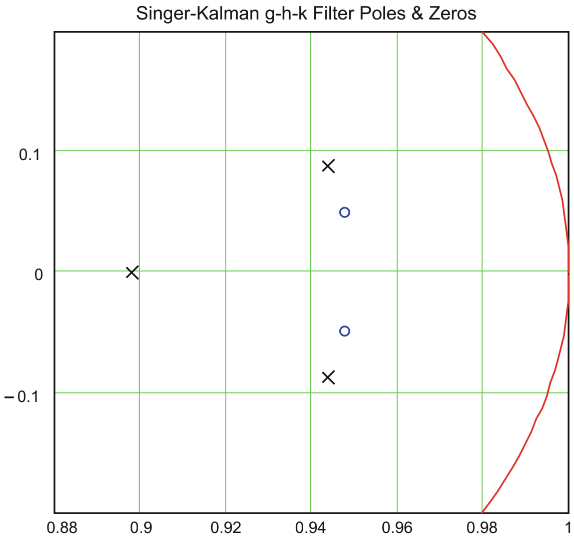

Fig. A.5 Poles and zeros of the g-h-k filter

The corresponding poles and zeros of the transfer functions are shown in Fig. A.5. The pairs of poles and zeros near $z = 1$ define the filter shape (bandwidth) at low frequencies. As the frequency response is determined by the product magnitude of the zero vectors (zero to point on unit circle) divided by the magnitude of the pole vectors (pole to point on unit circle), it is clear that the effect of the poles and zeros approximately cancel at low frequencies, as the pole-zero pairs are close together.

References

Brookner E (1998) Tracking and Kalman filtering made easy. Wiley. ISBN 0-471-18407-1.

Brown R, Hwang P (1997) Introduction to random signals and applied Kalman filtering. Wiley. ISBN 0-0471-12839-2.