Guidance Filter Fundamentals

Neil F. Palumbo, Gregg A. Harrison, Ross A. Blauwkamp, and Jeffrey K. Marquart

When designing missile guidance laws, all of the states necessary to mechanize the implementation are assumed to be directly available for feedback to the guidance law and uncorrupted by noise. In practice, however, this is not the case. The separation theorem states that the solution to this problem separates into the optimal deterministic controller driven by the output of an optimal state estimator. Thus, this article serves as a companion to our other article in this issue, "Modern Homing Missile Guidance Theory and Techniques," wherein optimal guidance laws are discussed and the aforementioned assumptions hold. Here, we briefly discuss the general nonlinear filtering problem and then turn our focus to the linear and extended Kalman filtering approaches; both are popular filtering methodologies for homing guidance applications.

INTRODUCTION

Our companion article in this issue, "Modern Homing Missile Guidance Theory and Techniques," discusses linear-quadratic optimal control theory as it is applied to the derivation of a number of different homing guidance laws. Regardless of the specific structure of the guidance law—e.g., proportional navigation (PN) versus the "optimal guidance law"—all of the states necessary to mechanize the implementation are assumed to be (directly) available for feedback and uncorrupted by noise. We refer to this case as the "perfect state information problem," and the resulting linear-quadratic optimal controller is deterministic. For example, consider the Cartesian version of PN, derived in the abovementioned companion article and repeated below for convenience:

$$u_{PN}(t) = \frac{3}{t_{go}^2}[x_1(t) + x_2(t)t_{go}]. \qquad (1)$$

Examining Eq. 1, and referring to Fig. 1, the states of the PN guidance law are $x_1(t) \triangleq r_{T_y} - r_{M_y}$,

γ_M =	Missile flight path angle
γ_T =	Target flight path angle
λ =	LOS angle
$\bar{\mathbf{r}}_M$ =	Missile inertial position vector
$\bar{\mathbf{r}}_T$ =	Target inertial position vector
$\bar{\mathbf{v}}_M$ =	Missile velocity vector
$\bar{\mathbf{v}}_T$ =	Target velocity vector
$\bar{\mathbf{a}}_M$ =	Missile acceleration, normal to LOS
$\bar{\mathbf{a}}_T$ =	Target acceleration, normal to V_T
L =	Lead angle
r_x =	Relative position x $(r_{Tx} - r_{Mx})$
r_y =	Relative position y $(r_{Ty} - r_{My})$
R =	Range to target

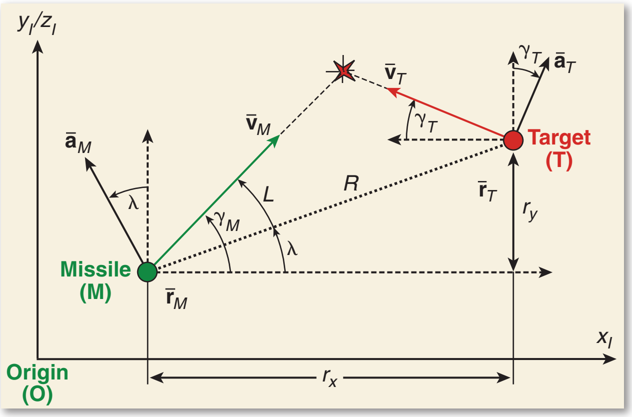


Figure 1. Planar engagement geometry. The planar intercept problem is illustrated along with most of the angular and Cartesian quantities necessary to derive modern guidance laws. The *x* axis represents downrange while the *y/z* axis can represent either crossrange or altitude. A flat-Earth model is assumed with an inertial coordinate system that is fixed to the surface of the Earth. The positions of the missile (M) and target (T) are shown with respect to the origin (O) of the coordinate system. Differentiation of the target–missile relative position vector yields relative velocity; double differentiation yields relative acceleration.

$x_2(t) \triangleq v_{T_y} - v_{M_y}$ (that is, components of relative position and relative velocity perpendicular to the reference x axis shown in Fig. 1). Recall from the companion article that the relative position "measurement" is really a pseudo-measurement composed of noisy line-of-sight (LOS) angle and relative range measurements. In addition to relative position, however, this (Cartesian) form of the deterministic PN controller also requires relative velocity and time-to-go, both of which are not (usually) directly available quantities. Hence, in words, the relative position pseudo-measurement must be filtered to mitigate noise effects, and a relative velocity state must be derived (estimated) from the noisy relative position pseudo-measurement (we dealt with how to obtain time-to-go in the companion article mentioned above). Consequently, a critical question to ask is: *Will deterministic linear optimal control laws (such as those derived by using the techniques discussed in our companion article in this issue) still produce optimal results given estimated quantities derived from noisy measurements?* Fortunately, the separation theorem states that the solution to this problem separates into the optimal deterministic controller driven by the output of an optimal state estimator.[1–6] In this article, we will introduce the optimal filtering concepts necessary to meet these needs.

In general terms, the purpose of filtering is to develop estimates of certain states of the system, given a set of noisy measurements that contain information about the states to be estimated. In many instances, one wants to perform this estimation process in some kind of "optimal" fashion. Depending on the assumptions made about the dynamic behavior of the states to be estimated, the statistics of the (noisy) measurements that are taken, and how optimality is defined, different types of filter structures (sometimes referred to as "observers") and equations can be developed. In this article, Kalman filtering is emphasized, but we first provide some brief general comments about optimal filtering and the more general (and implementationally complex) Bayesian filter.

BAYESIAN FILTERING

Bayesian filtering is a formulation of the estimation problem that makes no assumptions about the nature (e.g., linear versus nonlinear) of the dynamic evolution of the states to be estimated, the structure of the uncertainties involved in the state evolution, or the statistics of the noisy measurements used to derive the state estimates. It does assume, however, that models of the state evolution (including uncertainty) and of the measurement-noise distribution are available.

In the subsequent discussions on filtering, discrete-time models of the process and measurements will be the preferred representation. Discrete-time processes may arise in one of two ways: (*i*) the sequence of events takes place in discrete steps or (*ii*) the continuous-time process of interest is sampled at discrete times. For our purposes, both options come into play. For example, a radar system may provide measurements at discrete (perhaps unequally spaced) intervals. In addition, the filtering algorithm is implemented in a digital computer, thus imposing the need to sample a continuous-time process. Thus, we will begin by assuming very general discrete-time models of the following form:

$$\begin{aligned} \bar{\mathbf{x}}_k &= \bar{\mathbf{f}}_{k-1}(\bar{\mathbf{x}}_{k-1}, \bar{\mathbf{w}}_{k-1}) \\ \bar{\mathbf{y}}_k &= \bar{\mathbf{c}}_k(\bar{\mathbf{x}}_k, \bar{\boldsymbol{\upsilon}}_k) \,. \end{aligned} \tag{2}$$

In Eq. 2, $\bar{\mathbf{x}}_k$ is the state vector at (discrete) time k, the process noise $\bar{\mathbf{w}}_{k-1}$ is a functional representation of the (assumed) uncertainty in the knowledge of the state evolution from time $k-1$ to time k, $\bar{\mathbf{y}}_k$ is the vector of measurements made at time k, and the vector $\bar{\upsilon}_k$ is a statistical representation of the noise that corrupts the measurement taken at time k. We will revisit continuous-to-discrete model conversion later.

Eq. 2 models how the states of the system are assumed to evolve with time. The function $\bar{\mathbf{f}}_{k-1}$ is not assumed to have a specific structure other than being of closed form. In general, it will be nonlinear. Moreover, no assumption is made regarding the statistical structure of the uncertainty involved in the state evolution; we assume only that a reasonably accurate model of it is available. The second statement in Eq. 2 models how the measurements are related to the states. Again, no assumptions are made regarding the structure of $\bar{\mathbf{c}}_k$ or the statistics of the measurement noise.

Suppose that at time $k - 1$ one has a probability density that describes the knowledge of the system state at that time, based on all measurements up to and including that at time $k - 1$. This density is referred to as the prior density of the state expressed as $p(\bar{\mathbf{x}}_{k-1} \mid \mathbf{Y}_{k-1})$ where $\mathbf{Y}_{k-1}$ represents all measurements taken up to and including that at time $k - 1$. Then, suppose a new measurement becomes available at time k. The problem is to update the probability density of the state, given all measurements up to and including that at time k. The update is accomplished in a propagation step and a measurement-update step.

The propagation step predicts forward the probability density from time $k - 1$ to k via the Chapman–Kolmogorov equation (Eq. 3).[7]

$$\underbrace{p(\bar{\mathbf{x}}_k \mid \mathbf{Y}_{k-1})}_{\text{Prediction}} = \int \underbrace{p(\bar{\mathbf{x}}_k \mid \bar{\mathbf{x}}_{k-1})}_{\text{Transitional density}} \; \underbrace{p(\bar{\mathbf{x}}_{k-1} \mid \mathbf{Y}_{k-1})}_{\text{Prior}} \, d\bar{\mathbf{x}}_{k-1} \quad (3)$$

Eq. 3 propagates the state probability density function from the prior time to the current time. The integral is taken of the product of the probabilistic model of the state evolution (sometimes called the transitional density) and the prior state density. This integration is over the multidimensional state vector, which can render it quite challenging. Moreover, in general, no closed-form solution will exist.

The measurement-update step is accomplished by applying Bayes' theorem to the prediction shown above; the step is expressed in Eq. 4:

$$\underbrace{p(\bar{\mathbf{x}}_k \mid \mathbf{Y}_k)}_{\substack{\text{Posterior}\\ \text{Density}}} = \frac{\overbrace{p(\bar{\mathbf{y}}_k \mid \bar{\mathbf{x}}_k)}^{\text{Likelihood}} \; \overbrace{p(\bar{\mathbf{x}}_k \mid \mathbf{Y}_{k-1})}^{\text{Prediction}}}{\underbrace{\int p(\bar{\mathbf{y}}_k \mid \bar{\mathbf{x}}_k)\, p(\bar{\mathbf{x}}_k \mid \mathbf{Y}_{k-1})\, d\bar{\mathbf{x}}_k}_{\text{Normalizing Constant}}} . \quad (4)$$

Given a measurement $\bar{\mathbf{y}}_k$, the likelihood function (see Eq. 4) characterizes the probability of obtaining that value of the measurement, given a state $\bar{\mathbf{x}}_k$. The likelihood function is derived from the sensor-measurement model. Equations 3 and 4, when applied recursively, constitute the Bayesian nonlinear filter. The posterior density encapsulates all current knowledge of the system state and its associated uncertainty. Given the posterior density, optimal estimators of the state can be defined.

Generally, use of a Bayesian recursive filter paradigm requires a methodology for estimating the probability densities involved that often is nontrivial. Recent research has focused on the use of particle filtering techniques as a way to accomplish this. Particle filtering has been applied to a range of tracking problems and, in some instances, has been shown to yield superior performance as compared with other filtering techniques. For a more detailed discussion of particle filtering techniques, the interested reader is referred to Ref. 7.

KALMAN FILTERING

For the purpose of missile guidance filtering, the more familiar Kalman filter is widely used.[3,6,8,9] The Kalman filter is, in fact, a special case of the Bayesian filter. Like the Bayesian filter, the Kalman filter (*i*) requires models of the state evolution and the relationship between states and measurements and (*ii*) is a two-step recursive process (i.e., first predict the state evolution forward in time, then update the estimate with the measurements). However, Kalman revealed that a closed-form recursion for solution of the filtering problem could be obtained if the following two assumptions were made: (*i*) the dynamics and measurement equations are linear and (*ii*) the process and measurement-noise sequences are additive, white, and Gaussian-distributed. Gaussian distributions are described rather simply with only two parameters: their mean value and their covariance matrix. The Kalman filter produces a mean value of the state estimate and the covariance matrix of the state estimation error. The mean value provides the optimal estimate of the states.

As mentioned above, discrete-time models of the process and measurements will be the preferred representation when one considers Kalman filtering applications. In many instances, this preference will necessitate the representation of an available continuous-time model of the dynamic system by a discrete-time equivalent. For that important reason, in Box 1 we review how this process is applied. In Box 2, we provide a specific example of how one can discretize a constant-velocity continuous-time model based on the results of Box 1.

BOX 1. DEVELOPMENT OF A DISCRETE-TIME EQUIVALENT MODEL

We start with a linear continuous-time representation of a stochastic dynamic system, as shown in Eq. 5:

$$\begin{aligned} \dot{\bar{\mathbf{x}}}(t) &= \mathbf{A}(t)\bar{\mathbf{x}}(t) + \mathbf{B}(t)\bar{\mathbf{u}}(t) + \bar{\mathbf{w}}(t) \\ \bar{\mathbf{y}}(t) &= \mathbf{C}\bar{\mathbf{x}}(t) + \bar{\boldsymbol{\upsilon}}(t)\,. \end{aligned} \tag{5}$$

In this model, $\bar{\mathbf{x}} \in \mathbb{R}^n$ is the state vector, $\bar{\mathbf{u}} \in \mathbb{R}^m$ is the (deterministic) control vector (e.g., guidance command applied to the missile control system at time t), $\bar{\mathbf{y}} \in \mathbb{R}^p$ is the measurement vector, $\bar{\mathbf{w}} \in \mathbb{R}^n$ and $\bar{\upsilon} \in \mathbb{R}^p$ are vector white-noise processes (with assumed zero cross-correlation), and t represents time. Matrices **A**, **B**, and **C** are compatibly dimensioned so as to support the vector-matrix operations in Eq. 5. The white-noise processes are assumed to have covariance matrices as given in Eq. 6:

$$\begin{aligned} E[\bar{\mathbf{w}}(t)\bar{\mathbf{w}}^T(\tau)] &= \mathbf{Q}\delta(t-\tau) \\ E[\bar{\boldsymbol{\upsilon}}(t)\bar{\boldsymbol{\upsilon}}^T(\tau)] &= \mathbf{R}\delta(t-\tau)\,. \end{aligned} \tag{6}$$

In Eq. 6, $E(\cdot)$ represents the expectation operator defined as $E(x) = \int_{-\infty}^{\infty} xp(x)dx$, where $p(x)$ is the probability density of x. Above, note that the continuous Dirac delta function has the property that $\int_{-\infty}^{\infty} f(\tau)\delta(t-\tau)d\tau = f(t)$, for any $f(\cdot)$, continuous at t. Next, we consider samples of the continuous-time dynamic process described by Eq. 5 at the discrete times $t_0, t_1, \ldots, t_k$, and we use state-space methods to write the solution at time t_{k+1}[4, 6, 10]:

$$\bar{\mathbf{x}}(t_{k+1}) = \boldsymbol{\Phi}(t_{k+1}, t_k)\bar{\mathbf{x}}(t_k) + \underbrace{\int_{t_k}^{t_{k+1}} \boldsymbol{\Phi}(t_{k+1}, \tau)\mathbf{B}(\tau)\bar{\mathbf{u}}(\tau)d\tau}_{\Gamma_k\bar{\mathbf{u}}_k} + \underbrace{\int_{t_k}^{t_{k+1}} \boldsymbol{\Phi}(t_{k+1}, \tau)\bar{\mathbf{w}}(\tau)d\tau}_{\bar{\mathbf{w}}_k}\,. \tag{7}$$

In Eq. 7, $\boldsymbol{\Phi}(t_{k+1}, t_k) = e^{\mathbf{A}(t_{k+1}-t_k)}$ represents the system state transition matrix from time t_k to t_{k+1}, where the matrix exponential can be expressed as $e^{\mathbf{A}(t_{k+1}-t_k)} = \sum_{k=0}^{\infty} \mathbf{A}^k(t_{k+1}-t_k)^k/k!\,.$[1, 3–5, 10] Note that if the dynamic system is linear and time-invariant, then the state transition matrix may be calculated as $\boldsymbol{\Phi}(t_{k+1}, t_k) = \mathcal{L}^{-1}\{[s\mathbf{I}-\mathbf{A}]^{-1}\}$, where $\mathcal{L}^{-1}\{\cdot\}$ represents the inverse Laplace transform and **I** is a compatibly partitioned identity matrix. Thus, using Eq. 7, we write the (shorthand) discrete-time representation of Eq. 5 as given below:

$$\begin{aligned} \bar{\mathbf{x}}_{k+1} &= \boldsymbol{\Phi}_k\bar{\mathbf{x}}_k + \Gamma_k\bar{\mathbf{u}}_k + \bar{\mathbf{w}}_k \\ \bar{\mathbf{y}}_k &= \mathbf{C}_k\bar{\mathbf{x}}_k + \bar{\boldsymbol{\upsilon}}_k\,. \end{aligned} \tag{8}$$

In Eq. 8, the representation of the measurement equation is written directly as a sampled version of the continuous-time counterpart in Eq. 5. In addition, the discrete-time process and measurement-noise covariance matrices, $\mathbf{Q}_k$ and $\mathbf{R}_k$, respectively, are defined as shown below:

$$\begin{aligned} E[\bar{\mathbf{w}}_k\bar{\mathbf{w}}_i^T] &= \mathbf{Q}_k\delta_{k-i} \\ E[\bar{\boldsymbol{\upsilon}}_k\bar{\boldsymbol{\upsilon}}_i^T] &= \mathbf{R}_k\delta_{k-i} \\ E[\bar{\mathbf{w}}_k\bar{\boldsymbol{\upsilon}}_i^T] &= 0\,\forall i, k\,. \end{aligned} \tag{9}$$

Here we have used the discrete Dirac delta function, defined as $\delta_o = 1$, $\delta_n = 0$ for $n \neq 0$. Thus, as part of the discretization process, we also seek the relationships between the continuous and discrete-time pairs $\{\mathbf{Q}, \mathbf{Q}_k\}$ and $\{\mathbf{R}, \mathbf{R}_k\}$. It can be shown that given the continuous-time process disturbance covariance matrix **Q** and state transition matrix $\boldsymbol{\Phi}$, and referring to Eqs. 7 and 9, the discrete-time process disturbance covariance matrix $\mathbf{Q}_k$ can be approximated as given in Eq. 10[4]:

$$\mathbf{Q}_k \approx \int_{t_k}^{t_{k+1}} \boldsymbol{\Phi}(t_{k+1}, \tau)\mathbf{Q}(\tau)\boldsymbol{\Phi}^T(t_{k+1}, \tau)d\tau\,. \tag{10}$$

To obtain an approximation of the measurement-noise covariance, we take the average of the continuous-time measurement over the time interval $\Delta t = t_k - t_{k-1}$ as shown below[4]:

$$\bar{\mathbf{y}}_k = \frac{1}{\Delta t}\int_{t_{k-1}}^{t_k} [\mathbf{C}\bar{\mathbf{x}}(t) + \bar{\boldsymbol{\upsilon}}(t)]dt \approx \mathbf{C}\bar{\mathbf{x}}_k + \frac{1}{\Delta t}\int_{t_{k-1}}^{t_k} \bar{\boldsymbol{\upsilon}}(t)dt\,. \tag{11}$$

From Eqs. 6, 9, and 11, we obtain the desired relationship between the continuous-time measurement covariance **R** and its discrete-time equivalent $\mathbf{R}_k$:

$$\mathbf{R}_k = E[\bar{\boldsymbol{\upsilon}}_k\bar{\boldsymbol{\upsilon}}_i^T] = \frac{1}{\Delta t^2}\int_{t_{k-1}}^{t_k}\int_{t_{k-1}}^{t_k} E[\bar{\boldsymbol{\upsilon}}(\alpha)\bar{\boldsymbol{\upsilon}}^T(\tau)]d\alpha d\tau = \frac{\mathbf{R}}{\Delta t}\,. \tag{12}$$

BOX 2. DISCRETIZATION EXAMPLE: CONSTANT-VELOCITY MODEL

Here, we illustrate (with an example) how one can derive a discrete-time model from the continuous-time representation. For this illustrative example, we consider the state-space equations associated with a constant-velocity model driven by Gaussian white noise $\omega(t)$ (i.e., the velocity state is modeled as a Weiner process[4]).

$$\begin{aligned} \frac{d}{dt}\begin{bmatrix} r(t) \\ v(t) \end{bmatrix} &= \underbrace{\begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}}_{\mathbf{A}} \underbrace{\begin{bmatrix} r(t) \\ v(t) \end{bmatrix}}_{\mathbf{x}(t)} + \underbrace{\begin{bmatrix} 0 \\ 1 \end{bmatrix}}_{\mathbf{D}} \omega(t) \\ y(t) &= \underbrace{\begin{bmatrix} 1 & 0 \end{bmatrix}}_{\mathbf{C}} \begin{bmatrix} r(t) \\ v(t) \end{bmatrix} + \upsilon(t) \end{aligned} \tag{13}$$

(Compare the specific structure above to the general expression in Eq. 5.) In Eq. 13, the process and measurement-noise statistics are given by the following: $E[\omega(t)] = 0$, $E[\omega(t)\omega(\tau)] = Q\delta(t-\tau)$, $E[\upsilon(t)] = 0$, $E[\upsilon(t)\upsilon(\tau)] = R\delta(t-\tau)$, and $E[\omega(t)\upsilon(\tau)] = 0$. We compute the state transition matrix $\mathbf{\Phi}_k$ as $\mathbf{\Phi}_k = \mathcal{L}^{-1}\{[s\mathbf{I}-\mathbf{A}]^{-1}\}|_{t \leftarrow T}$, where we have defined the sample time as $T = t_{k+1} - t_k$. Then, from Eq. 7, we obtain the following discrete-time representation for this system:

$$\begin{aligned} \begin{bmatrix} r_{k+1} \\ v_{k+1} \end{bmatrix} &= \underbrace{\begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix}}_{\mathbf{\Phi}_k} \underbrace{\begin{bmatrix} r_k \\ v_k \end{bmatrix}}_{\mathbf{x}_k} + \omega_k \\ y_k &= \underbrace{\begin{bmatrix} 1 & 0 \end{bmatrix}}_{\mathbf{C}_k} \begin{bmatrix} r_k \\ v_k \end{bmatrix} + \upsilon_k \,. \end{aligned} \tag{14}$$

(Compare the discrete-time representation in this example to the more general one shown in Eq. 8.) Based on the results of the previous subsection, the discrete-time process and measurement-noise components in Eq. 14 are given by $\boldsymbol{\omega}_k = \int_{t_k}^{t_{k+1}} \mathbf{\Phi}(t_{k+1},\tau)\mathbf{D}\omega(\tau)d\tau = \int_0^T [T-\tau \;\; 1]^T \omega(\tau)d\tau$ and $\boldsymbol{\upsilon}_k = \frac{1}{T}\int_{t_k}^{t_{k+1}} \boldsymbol{\upsilon}(\tau)d\tau$, respectively. Consequently, by using Eqs. 10 and 12, the discrete-time process and measurement-covariance matrices are computed as

$$\begin{aligned} \mathbf{Q}_k &= E[\omega_k\omega_k^T] = E\left[\int_0^T \begin{bmatrix} T-\tau \\ 1 \end{bmatrix} \omega(\tau)d\tau \int_0^T [T-\eta \;\; 1]\omega(\eta)d\eta\right] \\ &= \int_0^T \begin{bmatrix} T-\tau \\ 1 \end{bmatrix}[T-\tau \;\; 1]Qd\tau = \begin{bmatrix} \frac{1}{3}T^3 & \frac{1}{2}T^2 \\ \frac{1}{2}T^2 & T \end{bmatrix} Q \\ \mathbf{R}_k &= E[\boldsymbol{\upsilon}_k\boldsymbol{\upsilon}_k^T] = \frac{1}{T^2}\int_0^T\int_0^T E[\boldsymbol{\upsilon}(u)\boldsymbol{\upsilon}^T(\zeta)]dud\zeta = \frac{1}{T^2}\int_0^T Rd\zeta = \frac{R}{T}\,. \end{aligned} \tag{15}$$

The Discrete-Time Kalman Filter

The theory says that the Kalman filter provides state estimates that have minimum mean square error.[4, 11, 12] An exhaustive treatment and derivation of the discrete-time Kalman filter is beyond the scope of this article. Instead, we shall introduce the design problem and directly present the derivation results. To this end, we note that if $\bar{\mathbf{x}}_k$ is the state vector at time k, and $\hat{\mathbf{x}}_k$ is an estimate of the state vector at time k, then the design problem may be stated as given below:

$$\begin{aligned} &\text{min:} && J = \mathit{trace}\{E([\bar{\mathbf{x}}_k - \hat{\mathbf{x}}_k][\bar{\mathbf{x}}_k - \hat{\mathbf{x}}_k]^T \,|\, \mathbf{Y}_k)\} \\ &\text{subject to:} && \begin{cases} \bar{\mathbf{x}}_k = \mathbf{\Phi}_{k-1}\bar{\mathbf{x}}_{k-1} + \mathbf{\Gamma}_{k-1}\bar{\mathbf{u}}_{k-1} + \bar{\mathbf{w}}_{k-1} \\ \bar{\mathbf{y}}_k = \mathbf{C}_k\bar{\mathbf{x}}_k + \boldsymbol{\upsilon}_k \end{cases} \\ &\text{where:} && \begin{cases} E[\bar{\mathbf{w}}_k\bar{\mathbf{w}}_i^T] = \mathbf{Q}_k\delta_{k-i} \\ E[\bar{\boldsymbol{\upsilon}}_k\bar{\boldsymbol{\upsilon}}_i^T] = \mathbf{R}_k\delta_{k-i} \\ E[\bar{\mathbf{w}}_k\bar{\boldsymbol{\upsilon}}_i^T] = 0 \,\forall i,k \end{cases} . \end{aligned} \tag{16}$$

The expressions in Eq. 16 embody the optimal design problem, which is to minimize the mean square estimation error $\mathit{trace}\{E([\bar{\mathbf{x}}_k - \hat{\mathbf{x}}_k][\bar{\mathbf{x}}_k - \hat{\mathbf{x}}_k]^T \,|\, \mathbf{Y}_k)\}$ subject to the assumed plant dynamics and given a sequence of measurements up to time k represented by $\mathbf{Y}_k = \{\bar{\mathbf{y}}_1, \bar{\mathbf{y}}_2, \ldots, \bar{\mathbf{y}}_k\}$.

As previously discussed, the discrete-time Kalman filter (algorithm) is mechanized by employing two distinct steps: (*i*) a prediction step (taken prior to receiving a new measurement) and (*ii*) a measurement-update step. As such, we will distinguish a state estimate that exists prior to a measurement at time k, $\hat{\mathbf{x}}_k^{(-)}$ (the *a priori* estimate) from one constructed after a measurement at time k, $\hat{\mathbf{x}}_k^{(+)}$ (the *posteriori* estimate). Moreover, we use the term $\mathbf{P}_k$ to denote the covariance of the estimation error, where $\mathbf{P}_k^{(-)} = E[\bar{\mathbf{x}}_k - \hat{\mathbf{x}}_k^{(-)}][\bar{\mathbf{x}}_k - \hat{\mathbf{x}}_k^{(-)}]^T$ and $\mathbf{P}_k^{(+)} = E[\bar{\mathbf{x}}_k - \hat{\mathbf{x}}_k^{(+)}][\bar{\mathbf{x}}_k - \hat{\mathbf{x}}_k^{(+)}]^T$. In what follows, we denote $\hat{\mathbf{x}}_0^{(+)}$ as our initial estimate, where $\hat{\mathbf{x}}_0^{(+)} = E[\bar{\mathbf{x}}(0)]$.

Table 1. Discrete-time Kalman filter algorithm.

	Step	Description	Expression
Initialization	(a)	Initial conditions	$\hat{\mathbf{x}}_0^{(+)} = E[\bar{\mathbf{x}}(0)],\ \mathbf{P}_0^{(+)} = E\left[\bar{\mathbf{x}}_0 - \hat{\mathbf{x}}_0^{(+)}\right]\left[\bar{\mathbf{x}}_0 - \hat{\mathbf{x}}_0^{(+)}\right]^T$
Prediction	(b)	State extrapolation	$\hat{\mathbf{x}}_k^{(-)} = \boldsymbol{\Phi}_{k-1}\hat{\mathbf{x}}_{k-1}^{(+)} + \boldsymbol{\Gamma}_{k-1}\bar{\mathbf{u}}_{k-1}$
	(c)	Error-covariance extrapolation	$\mathbf{P}_k^{(-)} = \boldsymbol{\Phi}_{k-1}\mathbf{P}_{k-1}^{(+)}\boldsymbol{\Phi}_{k-1}^T + \mathbf{Q}_{k-1}$
Correction	(d)	Kalman gain update	$\mathbf{K}_k = \mathbf{P}_k^{(-)}\mathbf{C}_k^T\left[\mathbf{C}_k\mathbf{P}_k^{(-)}\mathbf{C}_k^T + \mathbf{R}_k\right]^{-1}$
	(e)	Measurement update	$\hat{\mathbf{x}}_k^{(+)} = \hat{\mathbf{x}}_k^{(-)} + \mathbf{K}_k\left(\bar{\mathbf{y}}_k - \mathbf{C}_k\hat{\mathbf{x}}_k^{(-)}\right)$
	(f)	Error-covariance update	$\mathbf{P}_k^{(+)} = [\mathbf{I} - \mathbf{K}_k\mathbf{C}_k]\mathbf{P}_k^{(-)}$

Based on this description, the discrete-time Kalman filter algorithm is encapsulated as shown in Table 1.

In Table 1, the filter operational sequence is shown in the order of occurrence. The filter is initialized as given in step a. Steps b and c are the two prediction (or extrapolation) steps; they are executed at each sample instant. Steps d, e, and f are the correction (or measurement-update) steps; they are brought into the execution path when a new measurement $\bar{\mathbf{y}}_k$ becomes available to the filter. Figure 2 illustrates the basic structure of the linear Kalman filter, based on the equations and sequence laid out in Table 1.

Example: Missile Guidance State Estimation via Linear Discrete-Time Kalman Filter

In our companion article in this issue, "Modern Homing Missile Guidance Theory and Techniques," the planar version of augmented proportional navigation (APN) guidance law, repeated below for convenience (Eq. 17), requires estimates of relative position $x_1(t) \equiv r(t)$, relative velocity $x_2(t) \equiv v(t)$, and target acceleration $x_3(t) \equiv a(t)$ perpendicular to the target–missile LOS in order to develop missile acceleration commands (see Fig. 1).

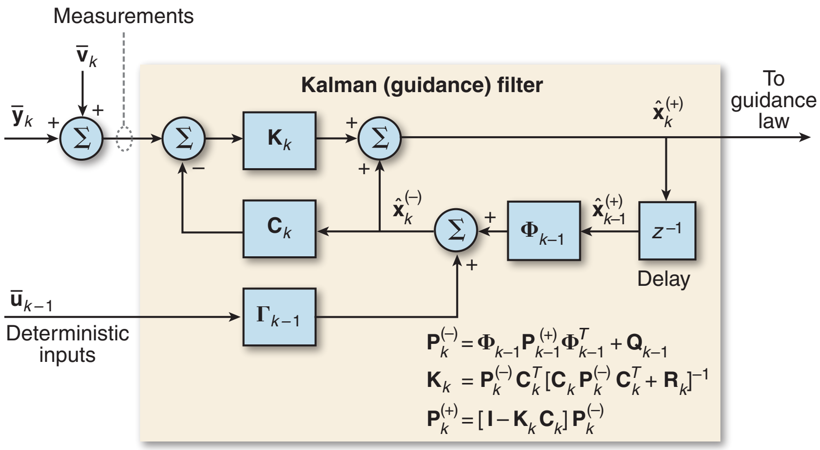

Measurements
$\bar{\mathbf{v}}_k$
$\bar{\mathbf{y}}_k$
+
+
Σ
Kalman (guidance) filter
Σ
−
$\mathbf{K}_k$
+
Σ
+
$\hat{\mathbf{x}}_k^{(+)}$
To guidance law
$\mathbf{C}_k$
$\hat{\mathbf{x}}_k^{(-)}$
Σ
+
$\boldsymbol{\Phi}_{k-1}$
$\hat{\mathbf{x}}_{k-1}^{(+)}$
z^{-1}
Delay
+
$\bar{\mathbf{u}}_{k-1}$
Deterministic inputs
$\boldsymbol{\Gamma}_{k-1}$
$\mathbf{P}_k^{(-)} = \boldsymbol{\Phi}_{k-1}\mathbf{P}_{k-1}^{(+)}\boldsymbol{\Phi}_{k-1}^T + \mathbf{Q}_{k-1}$
$\mathbf{K}_k = \mathbf{P}_k^{(-)}\mathbf{C}_k^T[\mathbf{C}_k\mathbf{P}_k^{(-)}\mathbf{C}_k^T + \mathbf{R}_k]^{-1}$
$\mathbf{P}_k^{(+)} = [\mathbf{I} - \mathbf{K}_k\mathbf{C}_k]\mathbf{P}_k^{(-)}$

Figure 2. The block diagram of the discrete-time, linear Kalman filter.

$$u_{APN}(t) = \frac{3}{t_{go}^2}\left[x_1(t) + x_2(t)t_{go} + \frac{1}{2}x_3(t)t_{go}^2\right] \quad (17)$$

Referring back to the planar engagement geometry shown in Fig. 1, consider the following stochastic continuous-time model representing the assumed engagement kinematics in the y (or z) axis:

$$\begin{bmatrix}\dot{r}(t)\\ \dot{v}(t)\\ \dot{a}_T(t)\end{bmatrix} = \begin{bmatrix}0 & 1 & 0\\ 0 & 0 & 1\\ 0 & 0 & 0\end{bmatrix}\begin{bmatrix}r(t)\\ v(t)\\ a_T(t)\end{bmatrix} + \begin{bmatrix}0\\ -1\\ 0\end{bmatrix}u(t) + \begin{bmatrix}0\\ 0\\ 1\end{bmatrix}\omega(t)$$
$$y(t) = \begin{bmatrix}1 & 0 & 0\end{bmatrix}\begin{bmatrix}r(t)\\ v(t)\\ a_T(t)\end{bmatrix} + \upsilon(t)\,. \quad (18)$$

In this example, the target acceleration state is driven by white noise; it is modeled as a Weiner process.[4] It can be shown that this model is statistically equivalent to a target maneuver of constant amplitude and random maneuver start time.[13] As in the previous discretization example, we assume that the process and measurement-noise statistics are given by the following relations: $E[\omega(t)] = 0$, $E[\omega(t)\omega(\tau)] = Q\delta(t-\tau)$, $E[\upsilon(t)] = 0$, $E[\upsilon(t)\upsilon(\tau)] = R\delta(t-\tau)$, and $E[\omega(t)\upsilon(\tau)] = 0$.

If we discretize the continuous-time system considered in Eq. 18, we obtain the following discrete-time dynamics and associated discrete-time process and measurement-noise covariance matrices:

$$\begin{bmatrix} r_{k+1} \\ v_{k+1} \\ a_{T_{k+1}} \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & T & \frac{1}{2}T^2 \\ 0 & 1 & T \\ 0 & 0 & 1 \end{bmatrix}}_{\boldsymbol{\Phi}_k} \underbrace{\begin{bmatrix} r_k \\ v_k \\ a_{T_k} \end{bmatrix}}_{\mathbf{x}_k} + \Gamma_k u_k + \overline{\boldsymbol{\omega}}_k, \qquad \mathbf{Q}_k = \begin{bmatrix} \frac{1}{20}T^5 & \frac{1}{8}T^4 & \frac{1}{6}T^3 \\ \frac{1}{8}T^4 & \frac{1}{3}T^3 & \frac{1}{2}T^2 \\ \frac{1}{6}T^3 & \frac{1}{2}T^2 & T \end{bmatrix} Q$$

$$y_k = \underbrace{[1 \;\; 0 \;\; 0]}_{\mathbf{C}_k} \begin{bmatrix} r_k \\ v_k \\ a_{T_k} \end{bmatrix} + \upsilon_k, \qquad \mathbf{R}_k = \frac{R}{T}, \qquad \Gamma_k = \begin{bmatrix} \frac{-T^2}{2} \\ -T \\ 0 \end{bmatrix}. \tag{19}$$

To illustrate the structure of the linear Kalman filter for the APN estimation problem, we will develop a block diagram of the filter based on the discrete-time model presented above. To help facilitate this, we assume that T is the rate at which measurements are available and that the filter runs at this rate (in the general case, this assumption is not necessary). For this case, the equations presented in Table 1 can be used to express the estimation equation in the alternate (and intuitively appealing) form given below:

$$\hat{\mathbf{x}}_k^{(+)} = [\mathbf{I} - \mathbf{K}\mathbf{C}_k][\boldsymbol{\Phi}_{k-1}\hat{\mathbf{x}}_{k-1}^{(+)} + \boldsymbol{\Gamma}_{k-1}\bar{\mathbf{u}}_{k-1}] + \mathbf{K}\bar{\mathbf{y}}_k . \tag{20}$$

Recall that, for APN, components of the state vector $\bar{\mathbf{x}}$ are defined to be relative position, $x_1 \triangleq r$, relative velocity, $x_2 \triangleq v$, and target acceleration, $x_3 \triangleq a_T$, leading to $\bar{\mathbf{x}} \triangleq [x_1 \;\; x_2 \;\; x_3]^T$. Therefore, using the system described by Eq. 19 in the alternate Kalman filter form shown in Eq. 20, we obtain the APN estimation equations below:

$$\begin{bmatrix} \hat{r}_k^{(+)} \\ \hat{v}_k^{(+)} \\ \hat{a}_{T_k}^{(+)} \end{bmatrix} = \begin{bmatrix} (1 - K_1)(\hat{r}_{k-1}^{(+)} + T\hat{v}_{k-1}^{(+)} + \frac{1}{2}T^2\hat{a}_{T_{k-1}}^{(+)} - \frac{1}{2}T^2 u_{k-1}) + K_1 y_k \\ -K_2(\hat{r}_{k-1}^{(+)} + T\hat{v}_{k-1}^{(+)} + \frac{1}{2}T^2\hat{a}_{T_{k-1}}^{(+)} - \frac{1}{2}T^2 u_{k-1} - y_k) + \hat{v}_{k-1}^{(+)} + T\hat{a}_{T_{k-1}}^{(+)} - Tu_{k-1} \\ -K_3(\hat{r}_{k-1}^{(+)} + T\hat{v}_{k-1}^{(+)} + \frac{1}{2}T^2\hat{a}_{T_{k-1}}^{(+)} - \frac{1}{2}T^2 u_{k-1} - y_k) + \hat{a}_{T_{k-1}}^{(+)} \end{bmatrix}. \tag{21}$$

Figure 3 depicts the structure dictated by the filtering equations above. Referring to Fig. 3, the makeup of the Kalman gain matrix, ***K***, is shown below (see Eq. d in Table 1):

$$K = \begin{bmatrix} K_1 \\ K_2 \\ K_3 \end{bmatrix} = \begin{bmatrix} \dfrac{p_{11}T}{p_{11}T + \sigma_r^2} \\ \dfrac{p_{12}T}{p_{11}T + \sigma_r^2} \\ \dfrac{p_{13}T}{p_{11}T + \sigma_r^2} \end{bmatrix}. \tag{22}$$

In this expression, T is the filter and measurement sample time (in seconds), $\sigma_r^2 \equiv R$ is the (continuous-time) relative position measurement variance (in practice, an estimate of the actual variance), and p_{ij} represents the {i, j}th entry of the (symmetric) *a priori* error-covariance matrix $\mathbf{P}_k^{(-)}$ as given below:

$$\mathbf{P}_k^{(-)} = \begin{bmatrix} p_{11} & p_{12} & p_{13} \\ p_{12} & p_{22} & p_{23} \\ p_{13} & p_{23} & p_{33} \end{bmatrix}. \tag{23}$$

$\mathbf{P}_k^{(-)}$ is recursively computed using Eq. c in Table 1 and Eq. 19 to compute the Kalman gain. The *a posteriori* error-covariance matrix, $\mathbf{P}_k^{(+)}$, is recursively computed by using Eq. f in Table 1 and leads to the following structure:

$$\mathbf{P}_k^{(+)} = \begin{bmatrix} (1-K_1)p_{11} & (1-K_1)p_{12} & (1-K_1)p_{13} \\ (p_{12}-p_{11}K_2) & (p_{22}-p_{12}K_2) & (p_{23}-p_{13}K_2) \\ (p_{13}-p_{11}K_3) & (p_{23}-p_{12}K_3) & (p_{33}-p_{13}K_3) \end{bmatrix}. \tag{24}$$

To mechanize this filter, an estimate of the measurement-noise variance, σ_r^2, is required. This parameter is important because it directly affects the filter bandwidth. There are a number of ways in which one may set or select this parameter. The conceptually simple thing to do is to set the parameter based on knowledge of the sensor characteristics, which may or may not be easy in practice because the noise variance may change as the engagement unfolds (depending on the type of targeting sensor being employed). Another, more effective, approach is to adaptively adjust or estimate the measurement variance in real time. See Ref. 14 for a more in-depth discussion on this topic.

Example: Discrete-Time APN Kalman Filter Performance

As a simple example, the Kalman filter shown in Fig. 3 was implemented to estimate the lateral motion of a weaving target. The total target simulation time was 5 s, and the filter time step (T) was 0.1 s. Figure 4 illustrates the results for this example problem. The target lateral acceleration (shown as a_T in Fig. 3) was modeled as a sinusoidal function in the $x1$ (lateral) direction with a 10-g amplitude and 5-s period. Motion in the $x2$ direction is a constant velocity of Mach 1 (1116.4 ft/s). The target initial conditions are $\bar{\mathbf{x}}$ = [0, 10,000]^T (ft) and $\bar{v}$ = [0, 1116]^T (ft/s) for position and velocity, respectively. The pseudo-measurement is the lateral position ($x1$), which was modeled as true $x1$ plus Gaussian white noise

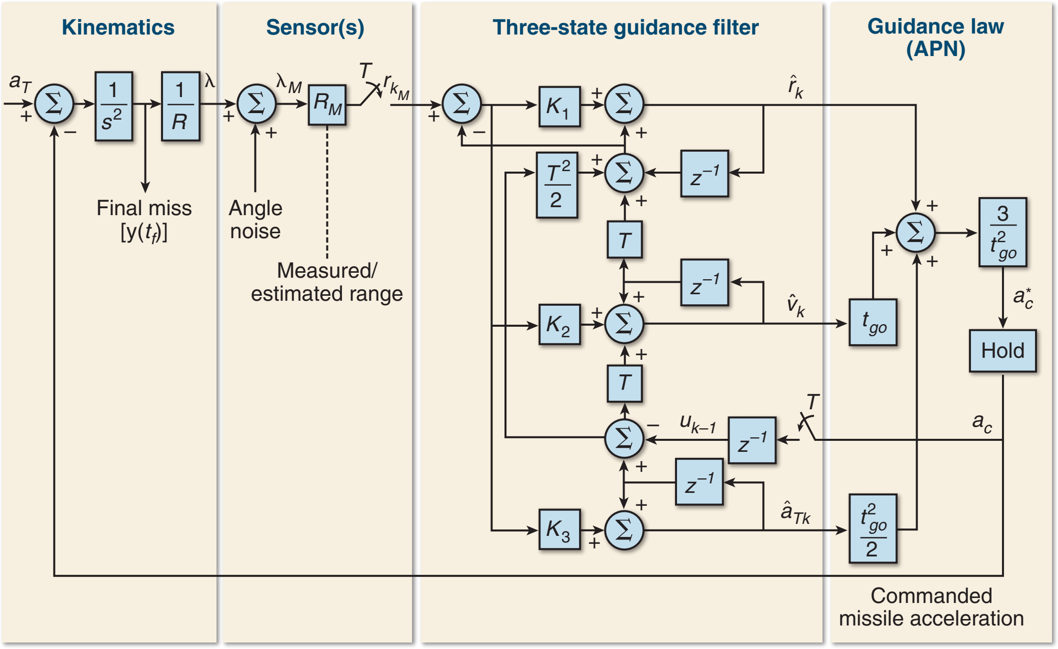


Figure 3. APN guidance filter. A discrete-time three-state Kalman filter is illustrated here, as is its place within the guidance loop. The filter state estimates are relative position, relative velocity, and target acceleration. These estimates are passed to the APN guidance law, which generates the acceleration commands necessary to achieve intercept. Notice that a perfect interceptor response to the acceleration command is assumed in this simplified feedback loop.

with statistics $N(0, \sigma = 1 \text{ ft})$ for the "low-noise" case and $N(0, \sigma = 10 \text{ ft})$ for the "high-noise" case. The estimated states of the filter comprise target lateral position, velocity, and acceleration. The filter was initialized by first collecting four lateral position measurement samples $\{x_{M1}(1), x_{M1}(2), x_{M1}(3), x_{M1}(4)\}$ and assigning the initial state values as shown below:

$$\hat{x}_0 = \sum_{i=1}^{4} \frac{x_{M1}(i)}{4}, \quad \hat{v}_0 = \frac{(x_{M1}(4) - x_{M1}(3))}{\Delta t}, \qquad (25)$$
$$\hat{a}_{T_0} = \frac{\hat{v}_0}{2\Delta t} - \frac{(x_{M1}(2) - x_{M1}(1))}{2\Delta t^2}.$$

As mentioned, two cases are shown in Fig. 4: (left) a low-noise case with a measurement standard deviation σ = 1 ft and (right) a high-noise case with measurement standard deviation of σ = 10 ft. The error-covariance matrix was initialized as $P_0|_{\sigma = 1 \text{ ft}} = \text{diag}\{1, 225, 2000\}$ for the low-noise case and $P_0|_{\sigma = 10 \text{ ft}} = \text{diag}\{10, 2000, 50{,}000\}$ for the high-noise case. For each case, the top plot shows the target position as $x1$ versus $x2$ (time is implicit). True, measured, and estimated positions are shown along with the 3σ bounds. For the low-noise case, it is difficult to distinguish truth from measurement or estimate (given the resolution of the plot). For the high-noise case, the position estimation error is more obvious. The second-row plots show the estimated and measured lateral position error for each case. The third-row plots illustrate lateral velocity, and the bottom plots show lateral acceleration. It is clear that with the high-noise measurement, the estimates deviate from truth much more as compared to the low-noise case.

NONLINEAR FILTERING VIA THE EXTENDED KALMAN FILTER

The conventional linear Kalman filter produces an optimal state estimate when the system and measurement equations are linear (see Eq. 5). In many filtering problems, however, one or both of these equations are nonlinear, as previously illustrated in Eq. 2. In particular, this nonlinearity can be the case for the missile guidance filtering problem. The standard way in which this issue of nonlinearity is treated is via the extended Kalman filter (EKF). In the EKF framework, the system and measurement equations are linearized about the current state estimates of the filter. The linearized system of equations then is used to compute the (instantaneous) Kalman gain sequence (including the *a priori* and *a posteriori* error covariances). However, state propagation is carried out by using the nonlinear equations. This "on-the-fly" linearization approach implies that the EKF gain sequence will depend on the particular series of (noisy) measurements as the engagement unfolds rather

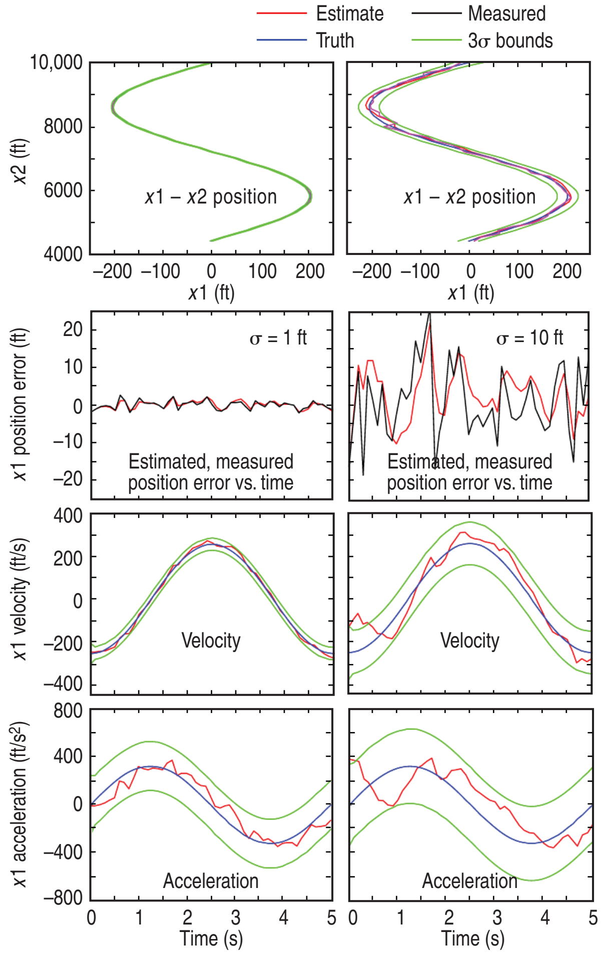


Figure 4. APN Kalman filter results. A planar linear Kalman filter is applied to estimate the position, velocity, and acceleration of a target that is maneuvering (accelerating) perpendicular to the *x*1 coordinate. The filter takes a position measurement in the *x*1 direction. The (sensor) noise on the lateral position measurement was modeled as true *x*1 plus zero-mean Gaussian white noise with standard deviation σ. The target maneuver is modeled as a sinusoid with a 10-*g* magnitude and a period of 5 s. Target motion in the *x*2 direction is constant, with a sea-level velocity of Mach 1 (~1116.4 ft/s). Two cases are shown: (left) a low-noise measurement case (σ = 1 ft) and (right) a high-noise case (σ = 10 ft). The plots illustrate the true and estimated position, velocity, and acceleration of the target, along with the 3σ bounds for the respective estimate. For each case, the second-row plot shows the errors in the measured and estimated position compared to truth vs. time.

than be predetermined by the process and measurement model assumptions (linear Kalman filter). Hence, the EKF may be more prone to filter divergence given a particularly poor sequence of measurements. Nevertheless, in many instances, the EKF can operate very well and, therefore, is worth consideration.

A complete derivation of the EKF is beyond the scope of this article. (See Refs. 3, 4, 11, and 12 for more on this topic.) Instead, we introduce the concept and present the results as a modification to the linear Kalman filter computations illustrated in Table 1. To start, consider the nonlinear dynamics and measurement equations given below, where the (deterministic) control and the process and measurement disturbances are all assumed to be input-affine:

$$\begin{aligned}\bar{\mathbf{x}}_k &= \bar{\mathbf{f}}_{k-1}(\bar{\mathbf{x}}_{k-1}) + \bar{\mathbf{b}}_{k-1}(\bar{\mathbf{x}}_{k-1})\bar{\mathbf{u}}_{k-1} + \bar{\mathbf{w}}_{k-1} \\ \bar{\mathbf{y}}_k &= \bar{\mathbf{c}}_k(\bar{\mathbf{x}}_k) + \bar{\boldsymbol{\upsilon}}_k \;.\end{aligned} \tag{26}$$

As before, we assume that the system disturbances are zero-mean Gaussian white-noise sequences with the following properties: $E[\bar{\mathbf{w}}_k\bar{\mathbf{w}}_i^T] = \mathbf{Q}_k\delta_{k-i}$, $E[\bar{\boldsymbol{\upsilon}}_k\bar{\boldsymbol{\upsilon}}_i^T] = \mathbf{R}_k\delta_{k-i}$, and $E[\bar{\mathbf{w}}_k\bar{\boldsymbol{\upsilon}}_i^T] = 0 \;\forall\; i, k$. In Eq. 26, $\bar{\mathbf{f}}_k$, $\bar{\mathbf{b}}_k$, and $\bar{\mathbf{c}}_k$ are nonlinear vector-valued functions of the state. We note that, given the *n*-dimensional state vector $\bar{\mathbf{x}}_k^* = [x^*_{1_k}, \ldots, x^*_{n_k}]^T$ and any vector-valued function of the state $\bar{\mathbf{m}}_k(\bar{\mathbf{x}}_k^*) = [m_{1_k}(\bar{\mathbf{x}}_k^*), \ldots, m_{n_k}(\bar{\mathbf{x}}_k^*)]^T$, we will denote the Jacobian matrix $\mathbf{M}_k$ as shown:

$$\mathbf{M}_k \triangleq \frac{\partial\bar{\mathbf{m}}_k(\bar{\mathbf{x}}_k^*)}{\partial\bar{\mathbf{x}}_k} = \begin{bmatrix} \dfrac{\partial m_{1_k}(\bar{\mathbf{x}}_k^*)}{\partial x_{1_k}} & \cdots & \dfrac{\partial m_{1_k}(\bar{\mathbf{x}}_k^*)}{\partial x_{n_k}} \\ \vdots & \ddots & \vdots \\ \dfrac{\partial m_{n_k}(\bar{\mathbf{x}}_k^*)}{\partial x_{1_k}} & \cdots & \dfrac{\partial m_{n_k}(\bar{\mathbf{x}}_k^*)}{\partial x_{n_k}} \end{bmatrix} . \tag{27}$$

Consequently, we can modify the Table 1 linear Kalman filter calculations to implement the sequence of EKF equations (Table 2).

Notice that the step sequence is identical to the linear Kalman filter. However, unlike the linear Kalman filter, the EKF is not an optimal estimator. Moreover, because the filter uses its (instantaneous) state estimates to linearize the state equations on the fly, the filter may quickly diverge if the estimation error becomes too great or if the process is modeled incorrectly. Nevertheless, the EKF is the standard in many navigation and GPS applications. The interested reader is referred to Refs. 4 and 8 for some additional discussion on this topic.

CLOSING REMARKS

In our companion article in this issue, “Modern Homing Missile Guidance Theory and Techniques,” a number of optimal guidance laws were derived and discussed. In each case, it was assumed that all of the states necessary to mechanize the implementation (e.g., relative position, relative velocity, target acceleration) were directly available for feedback and uncorrupted by noise (referred to as the perfect state information problem). In practice, this generally is not the case. In this article, we pointed to the separation theorem that states that an optimal solution to this problem separates into the opti-

Table 2. Discrete-time EKF algorithm.

Step	Description	Expression
(a)	Initial conditions	$\hat{\mathbf{x}}_0^{(+)} = E[\bar{\mathbf{x}}(0)],\ \mathbf{P}_0^{(+)} = E\left[\bar{\mathbf{x}}_0 - \hat{\mathbf{x}}_0^{(+)}\right]\left[\bar{\mathbf{x}}_0 - \hat{\mathbf{x}}_0^{(+)}\right]^T$
(b)	State extrapolation	$\hat{\mathbf{x}}_k^{(-)} = \bar{\mathbf{f}}_{k-1}\left(\bar{\mathbf{x}}_{k-1}^{(+)}\right) + \bar{\mathbf{b}}_{k-1}\left(\bar{\mathbf{x}}_{k-1}^{(+)}\right)\bar{\mathbf{u}}_{k-1}$
(c)	Error-covariance extrapolation	$\mathbf{P}_k^{(-)} = \left[\frac{\partial \bar{\mathbf{f}}_{k-1}(\hat{\mathbf{x}}_{k-1}^{(+)})}{\partial \bar{\mathbf{x}}_{k-1}}\right]\mathbf{P}_{k-1}^{(+)}\left[\frac{\partial \bar{\mathbf{f}}_{k-1}(\hat{\mathbf{x}}_{k-1}^{(+)})}{\partial \bar{\mathbf{x}}_{k-1}}\right]^T + \mathbf{Q}_{k-1}$
(d)	Kalman gain update	$\mathbf{K}_k = \mathbf{P}_k^{(-)}\left[\frac{\partial \bar{\mathbf{c}}_k(\hat{\mathbf{x}}_k^{(-)})}{\partial \bar{\mathbf{x}}_k}\right]^T\left(\left[\frac{\partial \bar{\mathbf{c}}_k(\hat{\mathbf{x}}_k^{(-)})}{\partial \bar{\mathbf{x}}_k}\right]\mathbf{P}_k^{(-)}\left[\frac{\partial \bar{\mathbf{c}}_k(\hat{\mathbf{x}}_k^{(-)})}{\partial \bar{\mathbf{x}}_k}\right]^T + \mathbf{R}_k\right)^{-1}$
(e)	Measurement update	$\hat{\mathbf{x}}_k^{(+)} = \hat{\mathbf{x}}_k^{(-)} + \mathbf{K}_k\left[\bar{\mathbf{y}}_k - \bar{\mathbf{c}}_k(\bar{\mathbf{x}}_k^{(-)})\right]$
(f)	Error-covariance update	$\mathbf{P}_k^{(+)} = \left(\mathbf{I} - \mathbf{K}_k\left[\frac{\partial \bar{\mathbf{c}}_k(\hat{\mathbf{x}}_k^{(-)})}{\partial \bar{\mathbf{x}}_k}\right]\right)\mathbf{P}_k^{(-)}$

mal deterministic controller driven by the output of an optimal state estimator. Thus, we focused here on a discussion of optimal filtering techniques relevant for application to missile guidance; this is the process of taking raw (targeting, inertial, and possibly other) sensor data as inputs and estimating the necessary signals (estimates of relative position, relative velocity, target acceleration, etc.) upon which the guidance law operates. Moreover, we focused primarily on (by far) the most popular of these, the discrete-time Kalman filter.

We emphasized the fact that the Kalman filter shares two salient characteristics with the more general Bayesian filter, namely, (*i*) models of the state dynamics and the relationship between states and measurements are needed to develop the filter and (*ii*) a two-step recursive process is followed (prediction and measurement update) to estimate the states of the system. However, one big advantage of the Kalman filter (as compared to general nonlinear filtering concepts) is that a closed-form recursion for solution of the filtering problem is obtained if two conditions are met: (*i*) the dynamics and measurement equations are linear and (*ii*) the process and measurement-noise sequences are additive, white, and Gaussian-distributed. Moreover, because discrete-time models of the process and measurements are the preferred representation when one considers Kalman filtering applications, we also discussed (and illustrated) how one can discretize a continuous-time system for digital implementation. As part of the discretization process, we pointed out the necessity to determine the relationships between the continuous and discrete-time versions of the process covariance matrix $\{\mathbf{Q}, \mathbf{Q}_k\}$ and the measurement-covariance matrix $\{\mathbf{R}, \mathbf{R}_k\}$. Reasonable approximations of these relationships were given that are appropriate for many applications.

Finally, we recognize that most real-world dynamic systems are nonlinear. As such, the application of linear Kalman filtering methods first requires the designer to linearize (i.e., approximate) the nonlinear system such that the Kalman filter is applicable. The EKF is an intuitively appealing heuristic approach to tackling the nonlinear filtering problem, one that often works well in practice when tuned properly. However, unlike its linear counterpart, the EKF is not an optimal estimator. Moreover, care must be taken when using an EKF because the approach is based on linearizing the state dynamics and output functions about the current state estimate and then propagating an approximation of the conditional expectation and covariance forward. Thus, if the initial estimate of the state is wrong, or if the process is modeled incorrectly, the EKF filter may quickly diverge.

REFERENCES

[1] Athans, M., and Falb, P. L., Optimal Control: *An Introduction to the Theory and Its Applications*, McGraw-Hill, New York (1966).

[2] Basar, T., and Bernhard, P., *H-Infinity Optimal Control and Related Minimax Design Problems*, Birkhäuser, Boston (1995).

[3] Bar-Shalom, Y., Li, X. R., and Kirubarajan, T., *Estimation with Applications to Tracking and Navigation*, John Wiley and Sons, New York (2001).

[4] Brown, R. G., and Hwang, P. Y. C., *Introduction to Random Signals and Applied Kalman Filtering*, 2nd Ed., John Wiley and Sons, New York (1992).

[5] Bryson, A. E., and Ho, Y.-C., *Applied Optimal Control*, Hemisphere Publishing Corp., Washington, DC (1975)

[6] Lewis, F. L., and Syrmos, V. L., *Optimal Control*, 2nd Ed., John Wiley and Sons, New York (1995).

[7] Ristic, B., Arulampalam, S., and Gordon, N., *Beyond the Kalman Filter: Particle Filters for Tracking Applications*, Artech House, Norwood, MA (2004).

[8]Siouris, G. M., *An Engineering Approach to Optimal Control and Estimation Theory*, John Wiley and Sons, New York (1996).

[9]Zarchan, P., and Musoff, H., *Fundamentals of Kalman Filtering: A Practical Approach*, American Institute of Aeronautics and Astronautics, Reston, VA (2000).

[10]Franklin, G. F., Powel, J. D., and Workman, M. L., *Digital Control of Dynamic Systems*, Chap. 2, Addison-Wesley, Reading, MA (June 1990).

[11]Chui, C. K., and Chen, G., *Kalman Filtering with Real-Time Applications*, 3rd Ed., Springer, New York (1999).

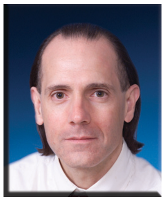
[12]Grewal, M. S., and Andrews, A. P., *Kalman Filtering Theory and Practice*, Prentice Hall, Englewood Cliffs, NJ (1993).

[13]Zames, G., "Feedback and Optimal Sensitivity: Model Reference Transformations, Multiplicative Seminorms, and Approximate Inverses," *IEEE Trans. Autom. Control* **26**, 301–320 (1981).

[14]Osborne, R. W., and Bar-Shalom, Y., "Radar Measurement Noise Variance Estimation with Targets of Opportunity," in *Proc. 2006 IEEE/AIAA Aerospace Conf.*, Big Sky, MT (Mar 2006).

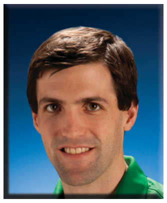
The Authors

Neil F. Palumbo is a member of APL's Principal Professional Staff and is the Group Supervisor of the Guidance, Navigation, and Control Group within the Air and Missile Defense Department (AMDD). He joined APL in 1993 after having received a Ph.D. in electrical engineering from Temple University that same year. His interests include control and estimation theory, fault-tolerant restructurable control systems, and neuro-fuzzy inference systems. Dr. Palumbo also is a lecturer for the JHU Whiting School's Engineering for Professionals program. He is a member of the Institute of Electrical and Electronics Engineers and the American Institute of Aeronautics and Astronautics. **Gregg A. Harrison** is a Senior Professional Staff engineer in the Guidance, Navigation, and Control Group of AMDD at APL. He holds B.S. and M.S. degrees in mathematics from the University of California, Riverside, an M.S.E.E. (aerospace controls emphasis) from the University of Southern California, and an M.S.E.E. (controls and signal processing emphases) from The Johns Hopkins University. He has more than 25 years of experience working in the aerospace industry, primarily on missile system and spacecraft programs. Mr. Harrison has extensive expertise in missile guidance, navigation, and control; satellite attitude control; advanced filtering techniques; and resource optimization algorithms. He is a senior member of the American Institute of Aeronautics and Astronautics. **Ross A. Blauwkamp** received a B.S.E. degree from Calvin College in 1991 and an M.S.E. degree from the University of Illinois in 1996; both degrees are in electrical engineering. He is pursuing a Ph.D. from the University of Illinois. Mr. Blauwkamp joined APL in May 2000 and currently is the supervisor of the Advanced Concepts and Simulation Techniques Section in the Guidance, Navigation, and Control Group of AMDD. His interests include dynamic games, nonlinear control, and numerical methods for control. He is a member of the Institute of Electrical and Electronics Engineers and the American Institute of Aeronautics and Astronautics. **Jeffrey K. Marquart** is a member of APL's Associate Professional Staff in AMDD. He joined the Guidance, Navigation, and Control Group in January 2008 after receiving both his B.S. and M.S. degrees in aerospace engineering from the University of Maryland at College Park. He currently is working on autopilot analysis, simulation validation, and guidance law design for the Standard Missile. Mr. Marquart is a member of the American Institute of Aeronautics and Astronautics. For further information on the work reported here, contact Neil Palumbo. His e-mail address is neil.palumbo@jhuapl.edu.


Neil F. Palumbo

Gregg A. Harrison

Ross A. Blauwkamp


Jeffrey K. Marquart

The *Johns Hopkins APL Technical Digest* can be accessed electronically at **www.jhuapl.edu/techdigest**.