our equations reduce to the Kalman filter measurement update. In general
\[
\hat{h}^-[x(t_k)] \neq h[\hat{x}^-(t_k)],
\]
however, in the linear case
\[
\hat{h}^-[x(t_k)] = \varepsilon^- (Hx_k) = H\hat{x}_k^- = h[\hat{x}^-(t_k)] \tag{5.74}
\]
Substituting this into Equations 5.68, 5.71, and 5.72 yields the measurement update in Table 2.2.

5.3 Extended Kalman Filter
Let us discuss continuous nonlinear systems of the form
\[
\dot{x} = a(x, t) + G(t)w \tag{5.75a}
\]
with measurements at discrete times \( t_k \) given by
\[
z_k = h[x(t_k), k] + v_k \tag{5.75b}
\]
where \( w(t) \sim (0, Q) \) and \( v_k \sim (0, R) \) are white noise processes uncorrelated with each other and with \( x(0) \sim (\tilde{x}_0, P_0) \). For simplicity, we assume the process noise matrix \( G(t) \) is independent of \( x(t) \). The case \( g[x(t), t] \) is covered in the problems.

Exact time and measurement updates for the hyperstate \( f_{x(t)/Z_k} \), where \( Z_k = \{z_k, z_{k-1}, \ldots, \} \) is all the data available through time \( t_k \), are given by Equations 5.31 and 5.29, respectively. Using these equations, we derived the exact time and measurement updates for the estimate (Equations 5.46 and 5.57) and error covariance (Equations 5.50 and 5.58). To make the measurement updates assume a more convenient form, we next restricted the estimate to a linear dependence on the data \( z_k \). This resulted in the measurement updates (Equation 5.68) for the estimate and (Equations 5.61 and 5.71) for the error covariance.

All these update equations are in general intractable, since they depend on the entire hyperstate, or equivalently on all the moments of \( x(t) \). We demonstrated at several points that in the linear Gaussian case, the Kalman filter was recovered.

What we would like to do now is to make some approximations to \( a(x, t) \) and \( h(x, k) \) to obtain a computationally viable algorithm for filtering in nonlinear systems. We shall see that the result is an “extended” version of the Kalman filter. See Gelb (1974).

5.3.1 Approximate Time Update
The exact time update for nonlinear \( a(x, t) \) is given by Equations 5.46 and 5.50. To find a time update that can be conveniently programmed on a computer,
expand \( a(x, t) \) in a Taylor series about \( \hat{x}(t) \), the current state estimate

\[
a(x, t) = a(\hat{x}, t) + \left. \frac{\partial a}{\partial x} \right|_{x=\hat{x}} (x - \hat{x}) + \cdots
\] (5.76)

Taking conditional expectations yields

\[
\hat{a}(x, t) = a(\hat{x}, t) + O(2)
\] (5.77)

where \( O(2) \) denotes terms of order 2. A first-order approximation to \( \hat{a}(x, t) \)
is, therefore, \( a(\hat{x}, t) \). Update 5.46 is thus replaced by

\[
\dot{\hat{x}} = a(\hat{x}, t)
\] (5.78)

To determine a first-order approximation to Equation 5.50, substitute for \( a(x, t) \) using the first two terms of Equation 5.76 and then take the indicated expected values. Representing the Jacobian as

\[
A(x, t) \triangleq \left. \frac{\partial a(x, t)}{\partial x} \right|_{x=\hat{x}}
\] (5.79)

this results in

\[
\dot{P} = A(\hat{x}, t)P + PA^T(\hat{x}, t) + GQG^T
\] (5.80)

Equations 5.78 and 5.80 represent an approximate, computationally feasible time update for the estimate and error covariance. The estimate simply propagates according to the nonlinear dynamics, and the error covariance propagates like that of a linear system with plant matrix \( A(\hat{x}, t) \).

Note that Jacobian \( A(\cdot, t) \) is evaluated for each \( t \) at the current estimate, which is provided by Equation 5.78, so that there is coupling between Equations 5.78 and 5.80. To eliminate this coupling, it is possible to introduce a further approximation and solve not Equation 5.80 but instead

\[
\dot{P}(t) = A[\hat{x}(t_k), t]P(t) + P(t)A^T[\hat{x}(t_k), t] + G(t)QG^T(t); \quad t_k \leq t < t_{k+1}
\] (5.81)

In this equation, the Jacobian is evaluated once using the estimate \( \hat{x}(t_k) \) after updating at \( t_k \) to include \( z_k \). This is used as the plant matrix for the time propagation over the entire interval until the next measurement time \( t_{k+1} \).

We shall soon present an example demonstrating how to accomplish the time update (Equations 5.78 and 5.80) very easily on a digital computer using a Runge–Kutta integrator.

### 5.3.2 Approximate Measurement Update

The optimal linear measurement update for nonlinear measurement matrix \( h[x(t_k), k] \) was found to be given by Equations 5.68, 5.71, and 5.72. To find a measurement update that can be conveniently programmed, expand \( h(x_k) \) in a Taylor series about \( \hat{x}_k^- \), the a priori estimate at time \( t_k \):

\[
h(x_k) = h(\hat{x}_k^-) + \left. \frac{\partial h}{\partial x} \right|_{x=\hat{x}_k^-} (x_k - \hat{x}_k^-) + \cdots
\] (5.82)
Represent the Jacobian as

\[ H(x, k) = \frac{\partial h(x, k)}{\partial x} \]  

(5.83)

To find a first-order approximate update, substitute for \( h(x_k) \) in Equations 5.68, 5.71, and 5.72 using the first two terms of Equation 5.82. The result is

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

(5.84)

\[ K_k = P^-(t_k) H^T(\hat{x}_k^-) [H(\hat{x}_k^-) P^-(t_k) H^T(\hat{x}_k^-) + R]^{-1} \]  

(5.85)

\[ P(t_k) = [I - K_k H(\hat{x}_k^-)] P^-(t_k) \]  

(5.86)

These equations represent an approximate, computationally feasible linear measurement update for the estimate and the error covariance. The residual is computed using the nonlinear measurement function \( h(\cdot, k) \) evaluated at the \textit{a priori} estimate \( \hat{x}_k^- \). The error covariance is found using the Jacobian matrix \( H(\hat{x}_k^-) \).

### 5.3.3 The Extended Kalman Filter

The approximate time and measurement updates we have derived are collected for reference in Table 5.1. They comprise what is known as the EKF for a continuous system with discrete measurements. If \( Q \) and \( R \) are time varying, it is only necessary to use their values \( Q(t) \) and \( R_k \) in the filter. A deterministic input \( u(t) \) is included in the equations for completeness.

An important advantage of the continuous–discrete EKF is that the optimal estimate is available continuously at all times, including times between the measurement times \( t_k \). Note that if \( a(x, u, t) \) and \( h(x, k) \) are linear, then the EKF reverts to the continuous–discrete Kalman filter in Section 3.7.

Given the continuous nature of dynamical systems and the requirement of microprocessors for discrete data, the continuous–discrete EKF is usually the most useful formulation for modern control purposes. The EKF approximation can also be applied to continuous systems with \textit{continuous} measurements. For the sake of completeness, the continuous–continuous EKF is given in Table 5.2. A discrete–discrete Kalman filter could also be written down very easily.

Since the time and measurement updates depend on Jacobians evaluated \textit{at the current estimate}, the error covariance and Kalman gain \( K_k \) cannot be computed off-line for the EKF. They must be computed in real time as the data become available.

In some cases, the nominal trajectory \( x_N(t) \) of the state \( x(t) \) is known \textit{a priori}. An example is a robot arm for welding a car door which always moves along the same path. In this case, the EKF equations can be used with \( \hat{x}(t) \) replaced by \( x_N(t) \). Since \( x_N(t) \) is known beforehand, the error covariance and \( K_k \) can now be computed off-line before the measurements are taken. This procedure of linearizing about a known nominal trajectory results in what is called the \textit{linearized Kalman filter}. If a controller is employed to keep the state
TABLE 5.1
Continuous–Discrete Extended Kalman Filter

<table>
<thead>
<tr>
<th>System model and measurement model</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \dot{x} = a(x, u, t) + G(t)w )</td>
</tr>
<tr>
<td>( z_k = h[x(t_k), k] + v_k )</td>
</tr>
<tr>
<td>( x(0) \sim (\bar{x}_0, P_0), w(t) \sim (0, Q), v_k \sim (0, R) )</td>
</tr>
</tbody>
</table>

Assumptions
\{w(t)\} and \{v_k\} are white noise processes uncorrelated with \( x(0) \) and with each other.

Initialization
\( P(0) = P_0, \hat{x}(0) = \bar{x}_0 \)

Time update
\( \dot{\hat{x}} = a(\hat{x}, u, t) \)
\( \dot{P} = A(\hat{x}, t)P + PA^T(\hat{x}, t) + GQG^T \)

Measurement update
Kalman gain
\( K_k = P^-(t_k)H^T(\hat{x}_k^-)[H(\hat{x}_k^-)P^-(t_k)H^T(\hat{x}_k^-) + R]^{-1} \)
Error covariance
\( P(t_k) = [I - K_kH(\hat{x}_k^-)P^-(t_k)] \)
Estimate
\( \hat{x}_k = \hat{x}_k^- + K_k[z_k - h(\hat{x}_k^-, k)] \)
Jacobians
\( A(x, t) = \frac{\partial a(x, u, t)}{\partial x} \)
\( H(x) = \frac{\partial h(x, k)}{\partial x} \)

\( x(t) \) on the nominal trajectory \( x_N(t) \), the linearized Kalman filter performs quite well.

Higher-order approximations to the optimal nonlinear updates can also be derived by retaining higher-order terms in the Taylor series expansions. For details, see Jazwinski (1970) and Gelb (1974).

It should be realized that the measurements times \( t_k \) need not be equally spaced. The time update is performed over any interval during which no data are available. When data become available, a measurement update is performed. This means that the cases of intermittent or missing measurements, and pure prediction in the absence of data can easily be dealt with using the EKF.

A block diagram of a software implementation of the continuous–discrete EKF is shown in Figure 3.14. The implementation of the continuous–continuous Kalman filter is even easier. The time update function is used to
Estimation for Nonlinear Systems

TABLE 5.2
Continuous–Continuous Extended Kalman Filter

<table>
<thead>
<tr>
<th>System model and measurement model</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \dot{x} = a(x, u, t) + G(t)w )</td>
</tr>
<tr>
<td>( z = h(x, t) + v )</td>
</tr>
<tr>
<td>( x(0) \sim (\bar{x}_0, P_0), w(t) \sim (0, Q), v(t) \sim (0, R) )</td>
</tr>
</tbody>
</table>

Assumptions
\( \{w(t)\} \) and \( \{v(t)\} \) are white noise processes uncorrelated with \( x(0) \) and with each other.

Initialization
\( P(0) = P_0, \dot{x}(0) = \bar{x}_0 \)

Estimate update
\( \dot{x} = a(\dot{x}, u, t) + K[z - h(\dot{x})] \)

Error covariance update
\( \dot{P} = A(\dot{x}, t)P + PA^T(\dot{x}, t) + GQG^T - PH(\dot{x}, t)R^{-1}H(\dot{x}, t)P \)

Kalman gain
\( K = PH(\dot{x}, t)R^{-1} \)

Jacobians
\( A(x, t) = \frac{\partial a(x, u, t)}{\partial x} \)
\( H(x, t) = \frac{\partial h(x, t)}{\partial x} \)

integrate the estimate and error covariance dynamics simultaneously. The vector \( X \) in the subroutine argument will contain the components of both \( \dot{x}(t) \) and \( P(t) \). The next examples illustrate how to write code to use the EKF. They also make the point that the filter implementation can be considerably simplified by doing some preliminary analysis!

**Example 5.3** FM Demodulation

*System and measurement*

Let message signal \( s(t) \) be normal with zero mean and variance \( \sigma^2 \). Suppose, it has a first-order Butterworth spectrum with a bandwidth of \( \alpha \) (cf., Anderson and Moore 1979),

\[
\Phi_s(\omega) = \frac{2\alpha\sigma^2}{\omega^2 + \alpha^2}
\]  
(5.87)
The FM modulated signal that is transmitted is
\[ y(t) = \sqrt{2}\sigma \sin(\omega_c t + \theta(t)) \] (5.88)
where \( \omega_c \) is the carrier frequency and the phase is related to the message by
\[ \theta(t) = \int_0^t s(\tau)d\tau \] (5.89)

It is desired to use the EKF to estimate \( s(t) \) from samples of the received signal, which is \( z(t) = y(t) + v(t) \) with \( v(t) \sim (0, r) \), a white noise added by the communication channel.

Defining state \( x = [s \quad \theta]^T \), the processes 5.87 and 5.89 can be modeled as (see Figure 2.18).
\[
\dot{x} = \begin{bmatrix} -\alpha & 0 \\ 1 & 0 \end{bmatrix} x + \begin{bmatrix} 1 \\ 0 \end{bmatrix} w \triangleq Ax + Gw \quad (5.90)
\]
with white noise \( w(t) \sim (0, 2\alpha \sigma^2) \). The discrete nonlinear time-varying measurements are
\[
z_k = \sqrt{2}\sigma \sin(\omega_c T_k + C x_k) + v_k \triangleq h(x_k, k) + v_k \quad (5.91)
\]
with \( C = \begin{bmatrix} 0 & 1 \end{bmatrix} \) and \( T \) the sampling period.

b. Time update
The time update is straightforward since the dynamics (Equation 5.90) are linear. Therefore, the error covariance propagates according to
\[
\dot{P} = AP + PA^T + GQG^T \quad (5.92)
\]

Letting
\[
P \triangleq \begin{bmatrix} p_1 \\ p_2 \\ p_3 \\ p_4 \end{bmatrix} \quad (5.93)
\]
we can write Equation 5.92 as the set of linear scalar equations
\[
\begin{align*}
\dot{p}_1 &= -2\alpha p_1 + 2\alpha \sigma^2 \\
\dot{p}_2 &= -\alpha p_2 + p_1 \\
\dot{p}_3 &= 2p_2 \\
\dot{p}_4 &= 2p_2
\end{align*} \quad (5.94a-c)
\]

It is quite easy to solve these analytically; however, we shall deliberately refrain from doing so to make the point that an analytical solution is not required to implement the Kalman filter.

The estimate propagates by
\[
\dot{\hat{x}} = A\hat{x} \quad (5.95)\]
Estimation for Nonlinear Systems

or

\[
\dot{s} = -\alpha \dot{s} \tag{5.96a}
\]

\[
\dot{\theta} = \dot{s} \tag{5.96b}
\]

c. Measurement update
The measurement is nonlinear, so the EKF update in Table 5.1 is required here. The measurement Jacobian is

\[
H(x) = \frac{\partial h(x, k)}{\partial x} = \sqrt{2}\sigma C \cos(\omega_c T_k + C x) \tag{5.97}
\]

Letting \( H(\hat{x}_k^-) \triangleq [h_1 \ h_2] \), we have

\[
h_1 = 0 \tag{5.98a}
\]

\[
h_2 = \sqrt{2}\sigma \cos(\omega_c T_k + \hat{\theta}_k^-) \tag{5.98b}
\]

The updated Kalman gain is found as follows. Let

\[
\delta \triangleq H(\hat{x}_k^-)P^- (t_k)H^T(\hat{x}_k^-) + \frac{r}{T} \tag{5.99}
\]

Defining

\[
P^- (t_k) \triangleq \begin{bmatrix} p_{1^-}^- & p_{2^-}^- \\ p_{2^-} & p_{4^-}^- \end{bmatrix} \tag{5.100}
\]

After simplifying, we have

\[
\delta = 2\sigma^2 p_{4^-}^- \cos^2(\omega_c T_k + \hat{\theta}_k^-) + r = h_2^2 p_{4^-}^- + \frac{r}{T} \tag{5.101}
\]

Hence,

\[
K_k \triangleq \begin{bmatrix} k_1 \\ k_2 \end{bmatrix} = \begin{bmatrix} p_{2^-}^- h_2 / \delta \\ p_{4^-}^- h_2 / \delta \end{bmatrix} \tag{5.102}
\]

The update covariance is given by

\[
P(t_k) = \begin{bmatrix} 1 & -k_1 h_2 \\ 0 & 1 - k_2 h_2 \end{bmatrix} P^- (t_k) \tag{5.103}
\]

or

\[
p_1 = p_{1^-}^- - k_1 h_2 p_{2^-}^- = p_{2^-}^- - \frac{h_2^2 (p_{2^-}^-)^2}{\delta} \tag{5.104a}
\]

\[
p_2 = (1 - k_2 h_2) p_{2^-}^- = \frac{r p_{2^-}^-}{\delta T} \tag{5.104b}
\]

\[
p_4 = (1 - k_2 h_2) p_{4^-}^- = \frac{r p_{4^-}^-}{\delta T} \tag{5.104c}
\]
Finally, the estimate update is (omitting the time subscript $k$)

$$\dot{s} = \dot{s}^- + k_1 [z - \sqrt{2} \sigma \sin(\omega_c T_k + \hat{\theta}^-)]$$

$$\dot{\theta} = \dot{\theta}^- + k_2 [z - \sqrt{2} \sigma \sin(\omega_c T_k + \hat{\theta}^-)]$$ (5.105a)

(5.105b)

d. Filter implementation

The time update (Equations 5.94 and 5.96) is implemented between measurement with an integration routine such as MATLAB’s ODE45. The measurement update (Equations 5.98, 5.101, 5.102, 5.104, and 5.105) is performed whenever data become available. Note that it is not necessary for measurements to be taken at regular intervals. For irregular or intermittent measurements at times $t_k$, it is only necessary to change the time dependence $T_k$ in Equations 5.98b and 5.105 to $t_k$. Note also that the EKF yields the optimal estimate at all times, even between the measurement times $t_k$.

The preliminary analysis we performed resulted in scalar updates, which are easier to program and faster to run than matrix updates. MATLAB implementations of the time update and measurement update are given in Figure 5.4. The vector $X$ in that figure contains both the estimates and the error covariance entries. We have avoided using two sets of error covariances, one for $P(t_k)$ and one for $P^-(t_k)$, by paying attention to the order of operations. Thus, in the program Equation 5.104a must be performed before Equation 5.104b since it needs the a priori value $p^-_2$, which is updated in Equation 5.104b.

We would expect the error covariance to behave much like the error covariance in Example 3.9.

**Example 5.4** Satellite Orbit Estimation

a. Satellite equations of motion

The equations of motion of a satellite in a planar orbit about a point mass are

$$\ddot{r} = r \dot{\theta}^2 - \frac{\mu}{r^2} + w_r$$

$$\ddot{\theta} = -\frac{2 \dot{r} \dot{\theta}}{r} + \frac{1}{r} w_\theta$$ (5.106)

(5.107)

where $r$ is the radial distance of the satellite from the mass and $\theta$ its angle from a reference point on the orbit (usually perigee, the closest point of the satellite’s orbit to the attracting mass $M$). White process noises $w_r(t) \sim (0, q_r)$ and $w_\theta(t) \sim (0, q_\theta)$ represent disturbance accelerations. The gravitational constant of the point mass $M$ is $\mu \triangleq GM$, with $G$ the universal gravitation constant. See Maybeck (1979), Bryson and Ho (1975), and Jazwinski (1970).
% Example 5.3
% FM Demodulation

function f5p3d1()

    % px(1:3)=Error covariances, px(4:5)=State estimates
    % Sampling period T=0.1 sec, Measurement every 1 sec.
    px=[10 10 10 10 10];
    i=0; wc=1; r=1; del=0.1; del_mu=1;
    global sig; sig=1;
    rnd=randn(1000,1);
    for t=0:del:10

        % Time Update
        t1=t+del;
        [td px] = ode45(@diffeqns,[t t1],px);
        px=px(end,:);

        % Measurement Update
        if (mod(t1,del_mu)==0)
            i=i+1;
            h=[0 sqrt(2)*sig*cos(wc*t1+px(5))];
            d=2*sig^2*px(3)*cos(wc*t1+px(5))+r/del;
            % Kalman Gain
            k=[px(2)*h(2)/d ; px(3)*h(2)/d];
            % Error covariances
            px(1)=px(1)-k(1)*h(2)*px(2);
            px(2)=(1-k(2)*h(2))*px(2);
            px(3)=(1-k(2)*h(2))*px(3);
            % Available measurement zk
            z=sqrt(2)*sig*sin(wc*t1+px(5))+rnd(i);
            % State estimate after measurement update
            px(4)=px(4)+k(1)*(z-sqrt(2)*sig*sin(wc*t1+px(5)));
            px(5)=px(5)+k(2)*(z-sqrt(2)*sig*sin(wc*t1+px(5)));
        end

        % Plotting
        subplot(3,1,1); plot(t1,px(1),’k.’); hold on; ylabel(’p_1’); grid on;
        subplot(3,1,2); plot(t1,px(2),’k.’); hold on; ylabel(’p_2’); grid on;
        subplot(3,1,3); plot(t1,px(3),’k.’); hold on; ylabel(’p_4’); grid on;
        xlabel(’Time (Secs.)’);
    end
    return

function dp=diffeqns(t,px)
    global sig;
    a=1;
    % Differential equations for Error covariance
    dp(1)=2*a*px(1)+2*a*sig^2;
    dp(2)=a*px(2)+px(1);
    dp(3)=2*px(2);
    % Differential equations for states
    dp(4)=-a*px(4);
    dp(5)=px(4);
    return

FIGURE 5.4
Time update and measurement update subroutines for FM demodulation.
b. Orbit estimation

An elliptic orbit in a plane is completely specified if we know

- $a$, the semimajor axis
- $e$, the eccentricity
- $t_p$, the time of perigee passage
- $\theta_p$, the angle between perigee and a reference axis.

Within a specified orbit, the satellite’s position is completely specified if we know $\theta(t)$. The variables $a, e, \theta_p$, and $\theta(t)$ are the Keplerian orbital elements.

The orbital elements of a satellite can be determined from its state $x \triangleq [r \, \dot{r} \, \dot{\theta} \, \dot{\theta}]^T$. For example,

$$a^2 = \frac{2}{r} - \frac{\dot{r}^2}{\mu} \quad (5.108)$$

$$e^2 = \left(1 - \frac{r}{2}\right)^2 + \frac{(r \dot{r})^2}{a \mu} \quad (5.109)$$
Another useful parameter is the period

\[ T = 2\pi \sqrt{\frac{a^3}{\mu}} \]  

(5.110)

The problem of orbit estimation is thus equivalent to the problem of determining the instantaneous state \( x(t) \) of the satellite, which we can do using the continuous–discrete EKF.

c. Time update

The estimate for \( x = [r \ i \ \theta \ \dot{\theta}]^T \) propagates between measurements according to the deterministic (noise-free) version of Equations 5.106 and 5.107.

\[
\begin{align*}
\frac{d\dot{r}}{dt} &= \dot{i} \\
\frac{d\dot{\theta}}{dt} &= \dot{\theta} \\
\frac{d\dot{\theta}}{dt} &= \frac{-2\dot{\theta}\dot{r}}{r} \\
\frac{d\theta}{dt} &= \dot{\theta}
\end{align*}
\]  

(5.111a, b, c, d)

To determine the error covariance time update we require the Jacobian

\[
A(x, t) = \begin{bmatrix}
0 & 1 & 0 & 0 \\
\dot{\theta}^2 + 2\mu/r^3 & 0 & 0 & 2r\dot{\theta} \\
0 & 0 & 0 & 1 \\
2\dot{\theta}\dot{r}/r^2 & -2\dot{\theta}/r & 0 & -2\dot{\theta}/r
\end{bmatrix}
\]  

(5.112)

In addition, the process noise matrix is

\[
g(x, t) = \begin{bmatrix}
0 \\
1 \\
0 \\
0 \\
1/r
\end{bmatrix}
\]  

(5.113)

which is a function of the state, so we must use (see the problems at the end of this chapter)

\[
\dot{P} = A(\hat{x}, t)P + PA^T(\hat{x}, t) + g(\hat{x}, t)Qg^T(\hat{x}, t) \\
+ \frac{1}{2} \sum_{i,j=1}^{n} \frac{\partial g}{\partial x_i} \bigg|_{x=\hat{x}} Q \frac{\partial g^T}{\partial x_j} \bigg|_{x=\hat{x}} p_{ij}(t)
\]  

(5.114)

where \( x \in \mathbb{R}^n \) and \( p_{ij} \) is the \((i, j)\)th element of \( P \). We could approximate by omitting the last term, but we shall include it for completeness since it
presents no real complication. The only nonzero term in the double sum is

\[
\frac{1}{2} \frac{\partial g}{\partial r} \bigg|_{x=\hat{x}} Q \frac{\partial g^T}{\partial r} \bigg|_{x=\hat{x}} p_{11}(t) = \frac{1}{2} \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & p_{11}q_0/\hat{r}^4 \end{bmatrix}
\] (5.115)

By substituting Equations 5.112 and 5.115 in Equations 5.114 and simplifying, 10 scalar differential equations are obtained for the \( n(n+1)/2 \) distinct components of \( P \). Defining auxiliary variables

\[
a_1 \triangleq \ddot{\theta}^2 + \frac{2\mu}{\hat{r}^3}
\] (5.116a)

\[
a_2 \triangleq \frac{2\hat{r}\ddot{\theta}}{\hat{r}^2}
\] (5.116b)

\[
a_3 \triangleq -\frac{2\dot{\theta}}{\hat{r}}
\] (5.116c)

\[
a_4 \triangleq -2\dot{\theta} \ddot{\theta}
\] (5.116d)

\[
a_5 \triangleq -2\ddot{r}/\hat{r}
\] (5.116e)

we have

\[
\dot{p}_{11} = 2p_{12}
\] (5.117a)

\[
\dot{p}_{12} = a_1p_{11} + p_{22} + a_4p_{14}
\] (5.117b)

\[
\dot{p}_{13} = p_{14} + p_{23}
\] (5.117c)

\[
\dot{p}_{14} = a_2p_{11} + a_3p_{12} + a_5p_{14} + p_{24}
\] (5.117d)

\[
\dot{p}_{22} = 2a_1p_{12} + 2a_4p_{24} + q_r
\] (5.117e)

\[
\dot{p}_{23} = a_1p_{13} + a_4p_{34} + p_{24}
\] (5.117f)

\[
\dot{p}_{24} = a_2p_{12} + a_3p_{22} + a_5p_{24} + a_1p_{14} + a_4p_{44}
\] (5.117g)

\[
\dot{p}_{33} = 2p_{34}
\] (5.117h)

\[
\dot{p}_{34} = a_2p_{13} + a_3p_{23} + a_5p_{34} + p_{44}
\] (5.117i)

\[
\dot{p}_{44} = 2a_2p_{14} + 2a_3p_{24} + 2a_5p_{44} + \frac{q_\theta}{\hat{r}^2} + \frac{p_{11}q_0}{2\hat{r}^4}
\] (5.117j)

The time update equations 5.111, 5.116, and 5.117 can easily be programmed. It is hoped that we are making the point that, even for fairly complicated nonlinear systems, some preliminary analysis makes the EKF very straightforward to implement!

d. Measurement update

We have deliberately made no mention yet of measurements. The time update is dependent only on the satellite dynamics, and whatever measurements are used, the subroutine time update routine does not change.
Several measurement schemes can be used, some of which combine measurements by both ground tracking stations and instruments on board the satellite. We shall discuss a simple scheme in which range $r$ and range rate $\dot{r}$ are measured by a station on the Earth’s surface. Thus, the data are

\[ z_k = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} x_k + \begin{bmatrix} v_1(k) \\ v_2(k) \end{bmatrix} \]  

with $v_1(k) \sim (0, \sigma_r^2)$ and $v_2(k) \sim (0, \sigma_{\dot{r}}^2)$. Suppose data are taken at times $t_k$, so that $x_k \triangleq x(t_k)$. Suppose also that $r$ and $\dot{r}$ are determined by independent means (e.g., radar ranges and doppler range rates) so that $v_1(k)$ and $v_2(k)$ are uncorrelated.

The measurements are linear, so the EKF measurement update in Table 5.1 reduces to the regular discrete Kalman filter update. This means that the stabilized measurement update software in Bierman (1977), for example, can be used. Since

\[ R = \begin{bmatrix} \sigma_r^2 & 0 \\ 0 & \sigma_{\dot{r}}^2 \end{bmatrix} \]  

is diagonal, scalar updates can be performed with no prewhitening of the data. Thus, at each time $t_k$ we would perform a two-step update to include separately the two independent measurements

\[ z_1(k) = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix} x_k + v_1(k) \]  

and

\[ z_2(k) = \begin{bmatrix} 0 & 1 & 0 & 0 \end{bmatrix} x_k + v_2(k) \]

**e. Discussion**

The sample times $t_k$ need not be uniform. We propagate through time by integrating the differential equations until data are available, at which point we call a measurement update routine to incorporate the data. If, at a given $t_k$, only one of the measurements (Equations 5.120 and 5.121) is available, we can include it individually with a call only to the appropriate scalar measurement update routine.

These examples clearly demonstrate the power and convenience of the EKF. It is a scheme that can be tailored to each application, and its modular structure can be exploited by using modular programming techniques that make it easy to modify the measurement scheme with a minimum of software redevelopment. It does not depend on a uniform data arrival rate.

### 5.4 Application to Adaptive Sampling

In this section, we present an in-depth discussion of the application of EKF to localization of mobile robotic platforms while concurrently sampling an unknown physical variable of interest.