GNSS Solutions:

A fully digital model for Kalman filters

"GNSS Solutions" is a regular column featuring questions and answers about technical aspects of GNSS. Readers are invited to send their questions to the columnist, **Dr. Mark Petovello**, Department of Geomatics Engineering, University of Calgary, who will find experts to answer them. His e-mail address can be found with his biography below.

Is it possible to define a fully digital state model for Kalman filtering?

The Kalman filter is a mathematical method, purpose of which is to process noisy measurements in order to obtain an estimate of some relevant parameters of a system. It represents a valuable tool in the GNSS area, with some of its main applications related to the computation of the user position/velocity/time (PVT) solution and to the integration of GNSS receivers with an inertial navigation system (INS) or other sensors.

The Kalman filter is based on a state space representation that describes the analyzed system as a set of differential equations that establishes the connections between the inputs, the outputs, and the state variables of the analyzed system.

Although the state space differential equations are expressed in the continuous time domain, the filter itself is implemented in the discrete time domain, as required by the periodic availability of data/measurements. The typical approach to this problem is to linearize the continuous time system using a Taylor series and then obtain a discrete time approximation therefrom. However, it can be helpful to approach the problem from a discrete time point of view directly.

Several such approaches have previously been developed in the signal processing field and can be extended to the Kalman filter. In the following, we compare the classical method based on the Taylor approximation with a method based on the Laplace-domain (*s*-domain) to *z*-domain transformations.

Our purpose is to give some simple rules and methods with which to write the state equations and to prove that the results of the classical methods are only a special case of the more general class of *s-z* transformations, beause the already known results will be obtained with the presented method.

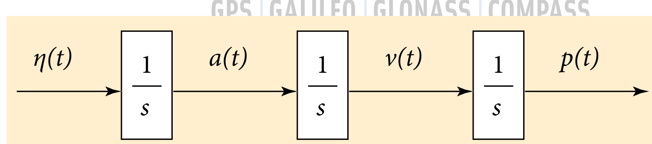

FIGURE 1 State model in Laplace domain

The Position-Velocity-Acceleration (PVA) Model

For illustration purposes, we consider the three-state (position, velocity and acceleration) model shown in Figure 1 in the Laplace domain (*s*-domain).

In Figure 1, $\eta(t)$ represents the white noise input function that models the acceleration $a(t)$ as a random walk. Without any loss of generality, a deterministic input will not be considered in the following analysis. The continuous time-invariant state-space model can be defined as

$$\underbrace{\begin{bmatrix}\dot{p}(t)\\ \dot{v}(t)\\ \dot{a}(t)\end{bmatrix}}_{\dot{\mathbf{x}}(t)} = \underbrace{\begin{bmatrix}0&1&0\\0&0&1\\0&0&0\end{bmatrix}}_{\mathbf{F}} \underbrace{\begin{bmatrix}p(t)\\ v(t)\\ a(t)\end{bmatrix}}_{\mathbf{x}(t)} + \underbrace{\begin{bmatrix}0\\0\\1\end{bmatrix}}_{\mathbf{g}}\eta(t) \qquad (1)$$

Mark Petovello is an Associate Professor in the Department of Geomatics Engineering at the University of Calgary. He has been actively involved in many aspects of positioning and navigation since 1997 including GNSS algorithm development, inertial navigation, sensor integration, and software development.

Email: mark.petovello@ucalgary.ca

$$\dot{\mathbf{x}}(t) = \mathbf{F}\mathbf{x}(t) + \mathbf{g}\eta(t) \quad (2)$$

We can obtain the continuous time states by solving the differential inhomogeneous system of equations given by Equation (2). The analytical solution of the continuous model is

$$\mathbf{x}(t) = e^{\mathbf{F}t}\mathbf{x}(0) + \int_0^t e^{\mathbf{F}(t-\tau)}\mathbf{g}\eta(\tau)d\tau \quad (3)$$

where the term $e^{\mathbf{F}t}$ is the matrix exponential of **F** and **x**(0) is the initial condition.

In our GNSS applications we are mainly interested in state evolution in the discrete-time domain. Therefore, we need to find a way to represent the continuous-time state vector **x**(*t*) as a discrete-time state vector **x**(*n*) and to write the state equations accordingly.

Method based on Taylor expansion

The typical approach to obtain the state evolution in the discrete time-domain consists of sampling (3) by $t = nT_s$, where $T_s = 1/f_s$ is a proper sampling interval that satisfies the Nyquist theorem and f_s is the sampling frequency. Then, the exponential in Equation (3) is expanded using a Taylor series expansion, truncated to the second order, as follows:

$$e^{\mathbf{F}T_s} = I_3 + \mathbf{F}T_s + \frac{\mathbf{F}T_s^2}{2!} + \cdots \quad (4)$$

After some manipulations, the final result is given by the following equation, which is the result often found in the literature:

$$\begin{bmatrix} p[n] \\ v[n] \\ a[n] \end{bmatrix} = \begin{bmatrix} 1 & T_s & \frac{T_s^2}{2} \\ 0 & 1 & T_s \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} p[n-1] \\ v[n-1] \\ a[n-1] \end{bmatrix} + \int_0^{T_s} \begin{bmatrix} \frac{v^2}{2} \\ v \\ 1 \end{bmatrix} \eta(\tau)d\tau \quad (5)$$

where the white noise *η(t)*, introduced in (2), explicitly appears.

This expression, in this form, is not ready yet to be implemented in a digital processor, because of the presence of the integral, which is a typical continuous-time operator. Moreover, *η(t)*, our input noise in (2), is a continuous-time white random process with a theoretically infinite variance, and a flat power spectrum density. We know that this kind of noise is an ideal model, generally adopted to analytically solve some equations, but which cannot be directly digitized.

The way around this is to digitize a filtered version of the white noise. In our application we have to understand how to deal with this and how to directly represent **ω**[*n*], the integrated and digitized version of *η(t)*, given by the last term in Equation (5). We will see that this problem is easily solved with the methods described in the next section.

Method Based on *s-z* Transformations

We present now an alternative approach, which is based on the idea of representing the signals and the systems in Figure 1 in the discrete-time domain, where the continuous-time t becomes the digital time n and the complex plane s becomes z.

These transformations are ruled by some well-known methods of the theory of digital signal processing. We first need to recall two important results of this discipline to find the way to transform the analog systems of Figure 1 into an all-digital system: the concept of a white sequence and the simulation theorem.

The White Sequence. In order to prevent aliasing of the white noise process, it is common to prefilter the signal prior to sampling. This eliminates the frequencies that cannot be represented in the sampled signal (i.e., those outside the Nyquist bandwidth) and avoids impairing the frequencies that can be represented.

The discrete-time version of the driving function $\eta(t)$, namely $\eta[n]$, is therefore the sampled version of the output of an anti-aliasing analog system $H_\eta(f)$ with two-sided bandwidth f_s. The random sequence $\eta[n]$ preserves the zero-mean property of the analog process, while the variance becomes finite, being the variance of the white noise filtered through the anti-aliasing filter $H_\eta(f)$. If $N_0/2$ is the noise spectral density of $\eta(t)$ the variance of $\eta[n]$ will be

$$\sigma_\eta^2 = \frac{N_0 F_s}{2}.$$

The power spectrum of the noise sequence is constant in the bandwidth $(-f_s/2, +f_s/2)$ and the sequence is of the type *iid* (independent and identically distributed). We can conclude that an iid sequence is the digital counterpart of the analog white noise and conveniently models filtered and digitized white noise. If in a real ADC (analog-to-digital converter) the anti-aliasing filter is not ideal, the iid model can be still used and the shape of the anti-aliasing filter can be modeled at the ADC output in the digital version of the system.

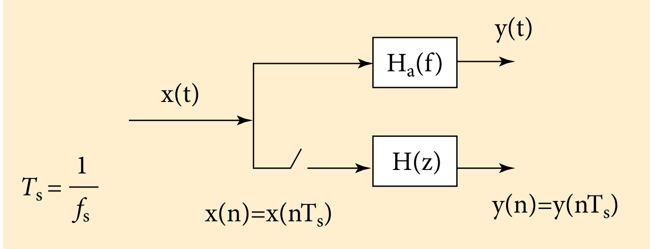

FIGURE 2 Simulation of linear time-invariant systems

The Simulation Theorem. To obtain a numerical version $H(z)$ of a generic analog transfer function $H_a(f)$, the Papoulis simulation theorem has to be considered: a discrete representation of an analytical version $H_a(f)$ can be simulated if a generic input $x[n] = x(nT_s)$ provides an output discrete signal that is a sampled version of the analog output $y(t)$ of the system $H_a(f)$, (Figure 2).

If $H(z)$ exists, then it represents the numeric simulator of $H_a(f)$. Starting from the relation in the analog branch of Figure 2,

$$y(t) = \int_{-\infty}^{\infty} H_a(f) X(f) e^{j2\pi f t} df \qquad (6)$$

we can obtain the discrete time version by

$$y(nT_s) = y[n] = \int_{-\infty}^{\infty} H_a(f) X(f) e^{j2\pi f n T_s} df \qquad (7)$$

However, according to the simulation theorem detailed in A. Papoulis, *Signal Analysis* (McGraw-Hill, New York, 1997), it can also be shown that

$$y[n] = \int_{-\infty}^{+\infty} X(f) H(e^{j2\pi f T_s}) e^{j2\pi f n T_s} df \qquad (8)$$

Compared to (7), it leads to the following conditions for the simulation theorem if $x(t)$ is a band-limited signal:

1. $X(f) = 0 \quad for \ |f| > B_x$
2. $H(e^{j2\pi f T_s}) = H_a(f) \quad for \ |f| < B_x$

where B_x is the input signal limited bandwidth.

From the *s* Plane to the *z* Plane

Having defined the discrete-time forcing function and the conditions to design the digital transfer functions, the next step is to convert a transfer function $H_a(s)$ from the s domain to the z domain, so as to satisfy the requirements imposed by the simulation theorem.

A unique method to perform this transformation does not exist. In fact, we can obtain the transfer function $H(z)$ from $H_a(s)$ by different mappings of the s plane on the unit circle of the z plane.

Whatever the mapping, the infinite imaginary axis of the Laplace domain has to be turned down on the unit circle, and, hence, an approximation has to be performed. A very common choice is to use the Bilinear Transformation. In this case the s plane is represented in the z domain by the following replacement:

$$s = \frac{2}{T_s}\frac{1-z^{-1}}{1+z^{-1}} \quad (9)$$

The bilinear transformation is also called the trapezoidal rule, as it approximates a generic integral $H_a(s) = 1/s$ with the trapezoidal formula. Other s-z transformations are possible, corresponding to different integration methods. The most important ones are summarized in **Figure 3**, where the transformations applied to $H(s) = 1/s$ are indicated, together with the corresponding integration formulas expressed in recursive form.

Digital representation of the PVA system

Starting from the previous example, we can derive a new form of (5), starting from a discrete-time driving function $\eta[n]$ and considering three different mappings of the integrators of Figure 1. Specifically, the first through third integrators are respectively replaced by the rectangular method, the bilinear transform, and the Cavalieri-Simpson method.

We should point out that the order of these transformations is only required to obtain the results already known in the literature, but it is not mandatory. In fact, any other order or transformation will lead to equally valid results, which are based on different approximations and implementation complexity.

By the chosen transformations, we can obtain the $H(z)$ functions for each integrator of **Figure 4**, so that each output can be described as a function of its input. Considering the input-output relations in the z domain and then computing their z inverse transforms, the discrete time input-output relations, shown in Figure 3, are easily obtained:

$$a[n] = a[n-1] + T_s\eta[n-1] \quad (10)$$

$$v[n] = v[n-1] + \frac{T_s}{2}(a[n] + a[n-1]) \quad (11)$$

$$p[n] = \frac{T_s}{3}(v[n] + 4v[n-1] + v[n-2]) + p[n-2]. \quad (12)$$

Finally, the relation between the states and the input noise in the time domain is the following:

$$\begin{aligned} p[n] &= p[n-1] + T_s v[n-1] + \frac{T_s^2}{2}a[n-1] + \frac{T_s^3}{6}\eta[n-1] \\ v[n] &= v[n-1] + T_s a[n-1] + \frac{T_s^2}{2}\eta[n-1] \\ a[n] &= a[n-1] + T_s\eta[n-1] \end{aligned} \quad (13)$$

Hence, the new state transition matrix, $\mathbf{\Phi}_d$ can be defined

$$\mathbf{\Phi}_d(n, n-1) = \begin{bmatrix} 1 & T_s & \frac{T_s^2}{2} \\ 0 & 1 & T_s \\ 0 & 0 & 1 \end{bmatrix} \quad (14)$$

that is equal to the first matrix on the right hand side of Equation (5).

Finally, we obtain

$$\begin{bmatrix} p[n] \\ v[n] \\ a[n] \end{bmatrix} = \begin{bmatrix} 1 & T_s & \frac{T_s^2}{2} \\ 0 & 1 & T_s \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} p[n-1] \\ v[n-1] \\ a[n-1] \end{bmatrix} + \begin{bmatrix} \frac{T_s^3}{6} \\ \frac{T_s^2}{2} \\ T_s \end{bmatrix} \eta[n] \quad (15)$$

As can be observed, the noise term in (15) differentiates from (5). We can easily verify that the noise covariance matrices obtained by applying the two different methods are the same if $\eta(t)$ is considered constant in the intervals $[nT_s;(n+1)T_s]$.

Again we note that the sequence of s-z transformations adopted in Figure 4 is not unique, and other choices are possible. A great advantage of the method is that it can be applied to any kind of rational transfer function $H_a(s)$, not

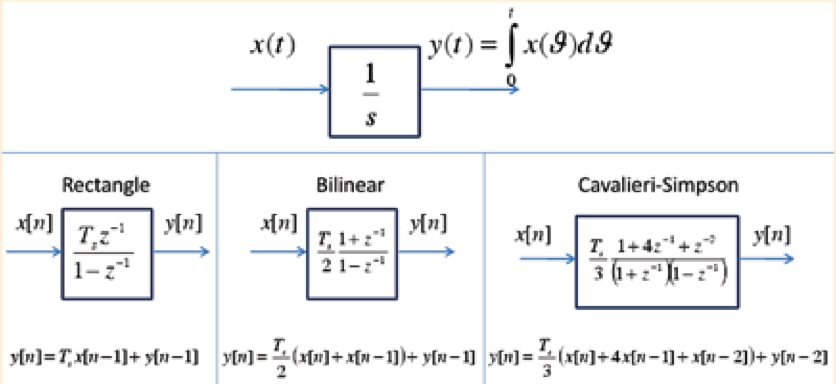

FIGURE 3 *s* plane to *z* plane transformations

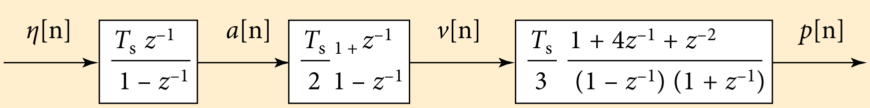

FIGURE 4 State model in the *z* domain

only to the integral of Figure 3.

The three proposed *s-z* transformations are easily obtained from Figure 3 by inverting $1/s$ and the corresponding functions in the z domain. Once applied to any rational $H_a(s)$, they give a rational $H(z)$, which admits a simple recursive input/output relationship.

Conclusion

The equivalence of the results provided by the two different methods shows how the fully digital method is equivalent to the classical procedure, limiting the calculations to discrete equations and vectors operations. The presented discrete time method can be useful for a novice of Kalman filter theory, for anyone who has to deal with complicated model definitions, and/or for those more familiar with discrete systems.

The fully digital approach is easily applied to any kind of $H(s)$; for example, a first order Gauss-Markov process can be modeled in the digital domain applying one among the transformations shown in Figure 3, with different levels of approximation. Even more complex systems such as INS/GPS integrated systems can be described using the fully digital method, obtaining also different results from the ones already described in the literature.

LETIZIA LO PRESTI, MARCO RAO, AND SIMONE SAVASTA

Letizia Lo Presti is a full professor with the Information Engineering Faculty of Politecnico di Torino. Her research covers the field of digital signal processing, simulation of telecommunication systems, and the technology of navigation and positioning systems. She is in the scientific coordinator of the Master on Navigation and Related Applications held by Politecnico di Torino.

Marco Rao is a Ph.D. student at Università di Palermo and is currently visiting the Istituto Superiore Mario Boella in Turin. His research interests covers the field of navigation, the integration of navigation systems, and cooperative positioning.

Simone Savasta is a researcher under grant at Politecnico di Torino. His research interests covers the fields of navigation, interference mitigation, and integration of navigation systems. IG