Textbook Models: Expanding Applications

Textbooks on systems and control theory and methods are replete with continuous-time dynamical models of the form

$$\dot{x}(t) = f(x(t), u(t)), \quad (1)$$

with $x(0)$ given, where $x(t)$ is the state and $u(t)$ is the control input at time t. There might also be an output equation such as $y(t) = g(x(t), u(t))$. A simple example is the celebrated linear state-space model

$$\dot{x}(t) = Ax(t) + Bu(t),$$
$$y(t) = Cx(t) + Du(t).$$

In the discrete-time setting, things look similar

$$x[k+1] = f(x[k], u[k]), \quad (2)$$

with $x[0]$ given and where the discrete index k now represents (discrete) time. The square brackets are used to distinguish (2) from its continuous-time counterpart (1).

On the foundations of these discrete- and continuous-time models emerge a rich theory from which a variety of methods for control design, analysis, and computations have been developed. It seems safe to assume that every graduate student in the field of systems and control is infused with this *received tradition* of our discipline.

Any discussion of such canonical models, especially in the classroom setting, would often be accompanied by real-world examples. Typically, these examples come from application domains in engineering. Examples include electrical systems (such as circuits and power grids), mechanical systems (for example, motors, servomechanisms, aeronautics, or fluid systems), electromechanical systems (like robotic manipulators or microelectromechanical systems), chemical systems (which give rise to process control applications), opto-electronic systems (such as lasers and lidar), RF systems (like radar and GPS), acoustic systems, and biological systems. Applications continue to expand. A noteworthy example is the use of systems and control methods in biology, giving rise to the study of *systems biology*.

These examples are all natural, physical systems. Discrete-time models arise from the time sampling of measurements made on such systems, usually via an analog-to-digital converter, for the purpose of digital computation. Granted, even with the plethora of example applications that textbooks and instructors can offer, students are sometimes unconvinced that applicability in the real world is worth the significant effort required to master the requisite concepts and methods.

IEEE Control Systems Society President Edwin Chong and Past-President Frank Doyle at the 2016 Conference on Decision and Control.

Perhaps not as commonly discussed are example applications that represent abstractions of the man-made, nonnatural world (a notable exception being [1]). In such a setting, the time index might not even represent "physical" time. Examples of these abstraction models come from topics such as multiagent systems, manufacturing, the Internet, computer systems, logistics, social networks, supply chains, project management, and finance. Following are somewhat nonstandard examples where our familiar models are equally applicable.

EXAMPLES

Discrete-Time System That Is Not from Sampling a Continuous-Time System

Discrete-time models that come from the time sampling of continuous-time systems, while certainly common in applications, are not the only ones that arise in practice. As a simple example, consider a bank account that starts with D dollars at month 0. Every month, the account accrues compounded interest at rate r. At the end of each month k, $u[k]$ dollars are deposited into the account (for instance, a paycheck). Let $x[k]$ be the account balance at start of month k. Then,

$$x[k+1] = (1+r)x[k] + u[k],$$

with $x[0] = D$, which is, of course, an instance of the usual linear discrete-time model. Therefore, a closed-form expression

Digital Object Identifier 10.1109/MCS.2017.2675039
Date of publication: 18 May 2017

can be written for the account balance each month. Note that the discrete-time index k here represents the index of the month. Although k corresponds to time, the discrete-time model is not the result of time sampling a continuous-time system.

Discrete-Time System with Event-Driven Dynamics

In some cases, the discrete-time index does not represent time at all. Here is a quick example. Consider a post office that opens in the morning with no customers present. Throughout the day, customers arrive and (possibly) queue up to a single postal server. Service is on a first-come, first-served basis, and each customer leaves after being served. k is the index of customer arrival, where $k = 1,2,\dots n$, $x[k]$ is the time spent in the queue by customer k, $a[k]$ is the time between the arrivals of customers k and $k+1$, and $s[k]$ is the time to serve customer k. Therefore,

$$x[k+1] = \max(x[k] + s[k] - a[k], 0), \tag{3}$$

with $x[0] = 0$. Equation (3) is called the *Lindley equation* and is an instance of a canonical model, albeit not exactly linear. Note that the discrete-time index k here represents the index of customers and is not physical time or a sampled version of continuous time. Moreover, the system is modeled *naturally* as a discrete-time system. The dynamic progression is fundamentally *event* driven, not *time* driven. *Queueing* models such as this one arise in telephone networks, Internet routers, call centers, inventory systems, and epidemiology. More general versions of such systems arise in the study of discrete-event systems [2] and hybrid systems [3].

Continuous-Time Model Arising from Discrete-Time System

In some cases, things go the other way around. Continuous-time models can arise from discrete-time systems. For example, in a single-server queue, such as the post office model, $x[k]$ could also represent the unfinished work at the time of arrival of customer k. In a system with a larger number of customers and a high rate of arrivals and departures, it is more convenient to treat the unfinished work as a continuum, much like treating water as a fluid rather than as a huge collection of discrete molecules. To this end, define a function $x(t)$ and write $x(t_k) = x[k]$, where t_k is the continuous-time instant at which customer k arrives. This gives rise to a *fluid* approximation of the unfinished work, which involves treating t as continuous time and the unfinished work $x(t)$ as the level of "fluid" in a tank, which is assumed to be differentiable in time. To account for the case of high arrival rates, consider the case where $a[k] = t_{k+1} - t_k$ is very small. More formally, take $a[k] \to 0$ and assume that there is a continuous-time function $r(t)$ such that $s[k]/a[k] \to r(t_k)$. (Note that technically the convergence above is for a *sequence of systems*.) After massaging the Lindley equation in (3) and taking limits,

$$\dot{x}(t) = \begin{cases} r(t) - 1, & \text{if } x(t) > 0, \\ \max(r(t) - 1,\ 0), & \text{otherwise.} \end{cases} \tag{4}$$

Think of $r(t)$ and 1 as *rates* of work added and removed, respectively. Note that if $x(0) \geq 0$, then $x(t) \geq 0$ for all t.

As before, this model is an instance of the general canonical model (1). More interestingly, (4) is a system that is *naturally* discrete time but has been approximated by a continuous-time model. Such a system is sometimes called a *fluid* queue and arises in a variety of applications, including fluid approximations for Internet congestion control [4], calculating buffer overflow probabilities in network routers [5], and fluid-queue models of battery life [6]. Indeed, such *continuum* models of discrete systems lead to interesting models with advantageous features [7].

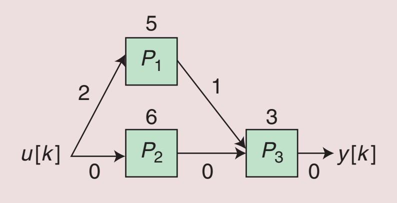

FIGURE 1 A simple production system [8].

State-Space Model with Nonstandard Linearity

Linear state-space models come with a rich set of tools for analysis, such as eigenvalues in the study of stability. Here is an example from [8] of a state-space model that is linear but with respect to a nonstandard algebra. Consider the production system illustrated in Figure 1, composed of three processing units $P_1, P_2,$ and P_3. Processing units P_1 and P_2 process the raw materials and send them to P_3, which then assembles the finished product, subject to the constraints that neither P_1 nor P_2 can start processing new raw material until it has finished processing what it has started. Likewise, P_3 cannot start assembly until it has finished assembling what it has started and received new processed materials from both P_1 and P_2.

The processing times at each of the processing units are shown above each respective unit in Figure 1. Each link is labeled with a number representing the time it takes the raw material to traverse the link. For example, it takes 2 s for the raw material to get from the input to P_1. The symbol $u[k]$ is the time instant at which raw material is fed to the system for the kth time, $x_i[k]$ is the time instant at which the ith processing unit starts working for the k th time, and $y[k]$ is the time instant at which the kth finished product leaves the system. Given this notation and the constraints described above, a mathematical description of the system is

$$\begin{aligned} x_1[k] &= \max(x_1[k-1] + 5,\ u[k] + 2), \\ x_2[k] &= \max(x_2[k-1] + 6,\ u[k] + 0), \\ x_3[k] &= \max(x_1[k] + 5 + 1, \\ &\qquad x_2[k] + 6 + 0, \\ &\qquad x_3[k-1] + 3\ ), \\ y[k] &= x_3[k] + 3 + 0. \end{aligned} \tag{5}$$

This system can be put in the general nonlinear canonical form, but it looks unwieldy, and the right-hand side is nonsmooth because of the max function.

The 2016 IEEE Control Systems Society Executive Committee celebrating a fruitful year at the 2016 Conference on Decision and Control.

Now, suppose a new algebra called the *max-plus algebra* is defined that has two binary operators $\oplus$ and $\otimes$, where $x \oplus y = \max(x,y)$ and $x \otimes y = x + y$. What are these odd symbols? According to [8]:

> The reason for using these symbols is that there is a remarkable analogy between $\oplus$ and conventional addition, and between $\otimes$ and conventional multiplication: many concepts and properties from linear algebra (such as the Cayley-Hamilton theorem, eigenvectors and eigenvalues, Cramer's rule, …) can be translated to the max-plus algebra by replacing $+$ by $\oplus$ and $\times$ by $\otimes$.

Indeed, in the max-plus algebra, (5) can be written succinctly in matrix form as

$$x[k] = A \otimes x[k-1] \oplus B \otimes u[k],$$
$$y[k] = C \otimes x[k],$$

Edwin Chong (center) having a cheerful discussion with Anna Stefanopoulou and Massoud Amin at the 2016 Conference on Decision and Control.

which, interestingly, is now in *linear* state-space form. The familiar tools for linear state-space systems can now be applied (subject, of course, to certain provisos).

PARTING THOUGHTS

The foundations of systems theory are so widely applicable that they go well beyond application to control; they also apply to signal processing and information theory, to name two related fields. What I have tried to illustrate here is that the models and tools so familiar to us are also useful in a wider class of applications, namely, those that come from the abstracted, man-made, nonnatural world. These include multiagent systems, manufacturing, the Internet, computer systems, logistics, social networks, supply chains, project management, and finance.

REFERENCES

[1] E. A. Lee and P. Varaiya, *Structure and Interpretation of Signals and Systems*. 2nd ed., 2011.

[2] C. G. Cassandras and S. Lafortune, *Introduction to Discrete Event Systems*. New York: Springer-Verlag, 2008.

[3] R. Goebel, R. G. Sanfelice, and A. R. Teel, *Hybrid Dynamical Systems: Modeling, Stability, and Robustness*. Princeton, NJ: Princeton Univ. Press, 2012.

[4] R. Srikant, *The Mathematics of Internet Congestion Control*. New York: Springer-Verlag, 2004.

[5] N. Hohn, D. Veitch, K. Papagiannaki, and C. Diot, "Bridging router performance and queuing theory," in *Proc. Joint Int. Conf. Measurement and Modeling of Computer Systems*, 2004, p. 355.

[6] G. L. Jones, P. G. Harrison, U. Harder, and T. Field, "Fluid queue models of battery life," in *Proc. 2011 IEEE 19th Int. Symp. Modeling, Analysis and Simulation of Computer and Telecommunication Systems*, July 25–27, 2011. doi: 10.1109/MASCOTS.2011.61.

[7] Y. Zhang, E. K. P. Chong, J. Hannig, and D. Estep, "Approximating extremely large networks via continuum limits," *IEEE Access*, vol. 1, pp. 577–595, Sept. 2013.

[8] B. De Schutter and T. van den Boom, "Max-plus algebra and max-plus linear discrete event systems: An introduction," in *Proc. 9th Int. Workshop on Discrete Event Systems*, Göteborg, Sweden, May 2008, pp. 36–42.

Edwin K. P. Chong