Table 2.5 Steady-state error according to system type for selected desired outputs

Desired Output, $y_d(t)$	Type 0 Steady-State Error	Type 1 Steady-State Error	Type 2 Steady-State Error
Unit step, $u_s(t)$	$1/[1 + \lim_{s\to 0} G(s)H(s)]$	0	0
Unit ramp, $t \cdot u_s(t)$	∞	$1/\lim_{s\to 0} sG(s)H(s)$	0
Unit parabola, $t^2 \cdot u_s(t)/2$	∞	∞	$1/\lim_{s\to 0} s^2G(s)H(s)$

classical control parlance that the system type is equal to the *number of pure integrations in the open-loop transfer function*, $G(s)H(s)$. The system of Example 2.16 with $H(s) = K$ is of type 0, while the system with the same plant, $G(s)$, in Example 2.17 becomes of type 1 with $H(s) = 1/s$, and of type 2 with $H(s) = 1/s^2$.

Based on our experience with Examples 2.16–2.18, we can tabulate (and you may verify) the steady-state errors of *stable* closed-loop systems, according to their type and the desired output in Table 2.5.

In Table 2.5 we have introduced a new animal called the *unit parabolic function*, given by $t^2 \cdot u_s(t)/2$ (also $t \cdot r(t)/2$). An example of $y_d(t)$ as a parabolic function is when it is desired to track an object moving with a *constant acceleration* (recall that $y_d(t) = r(t)$ represented an object moving with a *constant velocity*). From Table 2.5, it is evident that to track an object moving with constant acceleration, the closed-loop system must be at least of type 2.

2.8 Stability

As stated previously, one of the most important qualities of a control system is its *stability*. In Example 2.2 we saw that an inverted pendulum is *unstable* about the equilibrium point $\theta = \pi$. In addition, while discussing the transfer function in Section 2.4, we saw that a second order system whose poles have *negative real parts* (or positive damping-ratio, $\varsigma > 0$) exhibits step and impulse responses with exponentially decaying amplitudes, and we called such a system *stable*. While discussing steady-state error, we required that a closed-loop system must be stable before its steady-state error can be calculated, and defined stability tentatively as the property which results in a *bounded* output if the applied input is *bounded*. From Examples 2.2 and 2.12, we have a rough idea about stability, i.e. the tendency of a system (either linear, or nonlinear) to *regain* its equilibrium point once displaced from it. While nonlinear systems can have more than one equilibrium points, their stability must be examined about each equilibrium point. Hence, for nonlinear systems stability is a property of the equilibrium point. The pendulum in Example 2.2 has two equilibrium points, one of which is unstable while the other is stable. We can now define stability (and instability) more precisely for *linear* systems. For simplicity, we will focus on the *initial response* of a system (i.e. response to initial conditions when the applied input is *zero*), and by looking at it, try to determine whether a linear system is

stable. In this manner, we avoid having to classify the stability of a system according to the nature of the applied input, since stability is an intrinsic property of the linear system, independent of the input. There are the following three categories under which all *linear* control systems fall in terms of stability:

1. If the real parts of all the poles (roots of the denominator polynomial of the transfer function) are *negative*, then the initial response to finite initial conditions tends to a *finite* steady-state value in the limit $t \to \infty$. Such linear systems are said to be *asymptotically stable*. The aircraft of Example 2.10 is asymptotically stable, because all four poles have negative real parts.

2. If any pole of the linear system has a *positive* real part, then its initial response to finite initial conditions will be infinite in magnitude in the limit $t \to \infty$. Such a system is said to be *unstable.*

3. If all the poles of a system have real parts *less than or equal to zero*, and all the poles which have *zero* real parts are *simple*, i.e. they are not *repeated* (or *multiple*) poles (recall the discussion following Eq. (2.79)), the initial response of the system to finite initial conditions will keep on oscillating with a *finite* amplitude in the limit $t \to \infty$. Such a system is said to be *stable but not asymptotically stable* (because the response does not tend to an infinite magnitude, but also *does not* approach a constant steady-state value in the limit $t \to \infty$). However, if the poles having *zero* real part are *repeated* (i.e. they are *multiple poles* with the same imaginary part) the initial response of the system to finite initial conditions tends to infinity in the limit $t \to \infty$, and such a system is said to be *unstable*. In physical systems, complex poles occur in conjugate pairs (see Examples 2.10, 2.12). Thus, the only physical possibility of two (or more) repeated poles having zero real parts is that all such poles should be at the *origin* (i.e. their imaginary parts should also be zero).

We can summarize the stability criteria 1–3 by saying that if either the real part of any one pole is positive, or any one repeated pole has zero real part then the linear system is unstable. Otherwise, it is stable. A stable linear system having all poles with negative real parts is asymptotically stable. Using MATLAB you can easily obtain a location of the poles (and zeros) of a system in the Laplace domain with either the intrinsic command *roots(num)* and *roots(den)*, or the Control System Toolbox (CST) commands *pole(sys)*, *zero(sys)*, or *pzmap(sys)*, where *sys* is the transfer function LTI object of the system. From such a plot, it can be seen whether the system is asymptotically stable, stable (but not asymptotically stable), or unstable, from the above stated stability criteria. Another MATLAB (CST) command available for determining the poles of a system is *damp(den)* (see Example 2.10 for use of *damp*). Since the poles of a transfer function can be directly computed using MATLAB, one does not have to perform such mental calisthenics as the *Routh–Hurwitz* stability criteria (D'Azzo and Houpis [2]), which is a method for predicting the *number* of poles in the left and right half planes from the *coefficients* of the denominator polynomial by laboriously constructing a *Routh array* (see D'Azzo and Houpis [2] for details on Routh–Hurwitz stability criteria). Tabular methods such as Routh–Hurwitz were indispensible before the availability of digital computers