Chapter 10
Velocity Control of a Permanent Magnet Brushed Direct Current Motor

An electric motor is a device that is often employed to provide motion to mechanical systems. There are several classes of electric motors: induction motors [1], Chaps. 10 and 11, PM stepper motors [2], PM synchronous motors [3], switched reluctance motors [4, 5], brushless DC motors [6], chapter 10, etc. However, the PM brushed DC motor is the simplest motor to control. Moreover, as its mathematical model is linear and single–input–single–output, it is the ideal motor for using as an experimental prototype in the first stages of education in automatic control. In this chapter, velocity control of a PM brushed DC motor is studied. Position control of this class of motors is studied in the next chapter.

10.1 Mathematical Model

A PM brushed DC motor actuating on a load through a gear box is depicted in Fig. 10.1. The nomenclature employed is the following.

- u is the voltage applied at the motor armature terminals.
- i is the electric current through the armature circuit.
- Θ is the motor angular position.
- θ is the load angular position.
- $e_a = k_e\dot{\Theta}$ is the counter electromotive force where k_e is the counter electromotive force constant.
- $T = k_m i$ is the generated torque, or electromagnetic torque, where k_m is the motor torque constant.
- T_c is the load equivalent torque seen at the motor shaft. This torque opposes the motor movement (it is applied in the opposite sense to the definition of Θ).
- T_L is the torque applied by the motor on the load.

V. M. Hernández-Guzmán, R. Silva-Ortigoza, *Automatic Control with Experiments*,
Advanced Textbooks in Control and Signal Processing,
https://doi.org/10.1007/978-3-319-75804-6_10

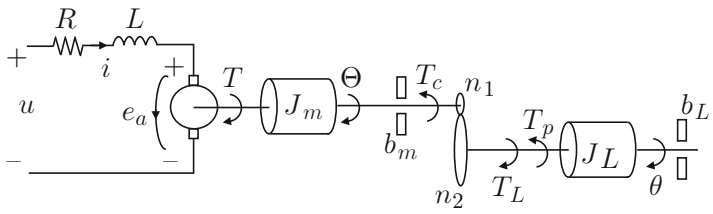

Fig. 10.1 A loaded permanent magnet (PM) brushed direct current (DC) motor

- T_p is the torque due to an external disturbance. It is assumed that this torque opposes the load movement (it is applied in the opposite sense to the definition of θ). If this is not the case, then T_p is negative.
- L is the armature inductance.
- R is the armature resistance.
- J_m is the inertia of the motor's rotor.
- b_m is the motor viscous friction coefficient.
- J_L is the load inertia.
- b_L is the load viscous friction coefficient.
- n_1 and n_2 stand for the teeth number at the motor axis and at the load axis respectively.

Note that a PM brushed DC motor is an electromechanical system with an electrical subsystem and a mechanical subsystem. The corresponding mathematical model is found by analyzing these individual subsystems to join them afterward. The model of each of these parts is obtained in the following.

10.1.1 The Motor Electrical Subsystem Model

Applying Kirchhoff's voltage law (see Fig. 10.1), the following is found:

$$\begin{aligned} \text{applied voltage} &= \sum \text{voltage drops in the mesh,} \\ u &= \text{voltage drop at inductance} + \text{voltage drop at resistance} + \\ &\quad + \text{counter electromotive force,} \\ u &= L\,\frac{di}{dt} + R\,i + e_a, \end{aligned} \tag{10.1}$$

where, according to (2.92) in Chap. 2, the counter electromotive force is given as:

$$e_a = k_e \dot{\Theta},$$

where dot "˙" stands for the first time derivative.

10.1.2 The Motor Mechanical Subsystem Model

Newton's Second Law must be applied separately to each one of the two bodies: the motor's rotor and its load.

10.1.2.1 Motor's Rotor Model

$$\text{inertia} \times \text{angular acceleration} = \sum \text{torques applied to inertia } J_m,$$

$$J_m \ddot{\Theta} = \text{generated torque} - \text{friction torque} - \text{load torque},$$

$$J_m \ddot{\Theta} = T - b_m \dot{\Theta} - T_c, \tag{10.2}$$

where, according to (2.93), Chap. 2, the generated torque is given as:

$$T = k_m \, i.$$

The addition on the right-hand side of (10.2) is algebraic, i.e., the signs indicate the sense of the corresponding applied torque. A negative sign indicates that the torque opposes the motion of the inertia J_m, whereas a positive sign indicates that the torque favors the motion of the inertia J_m.

10.1.2.2 Load Model

Because of the presence of a gear box, it is necessary to use (2.86) and (2.87), Chap. 2, to conclude that:

$$\Theta = n \, \theta, \qquad n = \frac{n_2}{n_1}, \tag{10.3}$$

$$T_L = n \, T_c, \tag{10.4}$$

Applying Newton's Second Law to the load:

$$\text{inertia} \times \text{angular acceleration} = \sum \text{torques on inertia } J_L,$$

$$J_L \ddot{\theta} = T_L - b_L \dot{\theta} - T_p. \tag{10.5}$$

The motor's mechanical subsystem model is obtained using (10.2), (10.5), (10.3), and (10.4) as:

$$J_m\,\ddot{\Theta} = k_m\,i - b_m\dot{\Theta} - \frac{1}{n}T_L,$$

$$J_m\,\ddot{\Theta} = k_m\,i - b_m\dot{\Theta} - \frac{1}{n}\left(J_L\,\ddot{\theta} + b_L\dot{\theta} + T_p\right),$$

$$n\,J_m\,\ddot{\theta} = k_m\,i - b_m\,n\dot{\theta} - \frac{1}{n}\left(J_L\,\ddot{\theta} + b_L\dot{\theta} + T_p\right),$$

$$\left(n^2\,J_m + J_L\right)\ddot{\theta} + \left(n^2 b_m + b_L\right)\dot{\theta} = n\,k_m\,i - T_p. \tag{10.6}$$

Defining:

$$J = n^2\,J_m + J_L, \qquad b = n^2\,b_m + b_L, \tag{10.7}$$

(10.6) can be written as:

$$J\ddot{\theta} + b\dot{\theta} = n\,k_m\,i - T_p. \tag{10.8}$$

Finally, the complete motor mathematical model is given by (10.1) and (10.8), i.e.,

$$L\,\frac{di}{dt} = u - R\,i - n\,k_e\,\dot{\theta}, \tag{10.9}$$

$$J\ddot{\theta} = -b\,\dot{\theta} + n\,k_m\,i - T_p. \tag{10.10}$$

Note that the differential equation in (10.10) is identical to that presented in (2.118) corresponding to the system shown in Fig. 2.29a, Chap. 2. The only differences are because (2.118) is expressed in terms of an applied external torque (which is equal to $k_m\,i$ in a DC motor) and (2.118) does not take into account any external torque applied on body 2 ($T_p = 0$), i.e., on the load.

The differential equations in (10.9) and (10.10) represent the mathematical model of the DC motor-load system because the load position θ can be found as a function of time by solving these differential equations if all the constants involved are known in addition to the voltage applied at the armature terminals. Using the Laplace transform in (10.9) and (10.10) by assuming zero initial conditions, the following is found:

$$I(s) = \frac{1}{sL + R}[U(s) - n\,k_e\,s\theta(s)], \tag{10.11}$$

$$\theta(s) = \frac{1}{s^2 J + sb}[n\,k_m\,I(s) - T_p(s)], \tag{10.12}$$

where $I(s)$, $U(s)$, $\theta(s)$ and $T_p(s)$, stand for Laplace transforms of i, u, θ and T_p respectively.

10.2 Power Amplifier

It is important to stress the necessity of employing a power amplifier. A control system can be divided into two parts: (i) the control part, and (ii) the power part. The control part includes all the hardware devoted to processing information (sensors and control algorithms) to be used to determine the command to be sent to the controlled process. This task requires precise electronic circuits that handle weak electrical signals, i.e., they merely process information but they do not handle high power levels. However, these commands must be capable of producing significant variations of the signal applied at the plant (motor) input. Furthermore, the plant input handles important power levels to be capable of producing the expected changes in the plant. This is the reason for including a power stage in a control system.

The power stage is commonly composed of a power amplifier that receives at its input the weak signal delivered by the control stage (control signal) and it must deliver at its output a strong high-power signal that is directly applied to the plant. For instance, in electromechanical systems a power amplifier must deliver the voltage to be applied directly to electric motors, which convert such an electric signal into torque to produce the mechanism motion. Hence, a power amplifier is placed between the weak signal delivered by the controller and the strong signal applied to the motor. According to this, a power amplifier can be modeled as a constant A_p relating to voltage at its input u_i (control signal) and voltage delivered at its output u (at the motor's armature terminals):

$$u = A_p \, u_i. \tag{10.13}$$

10.3 Electric Current Control

A common technique for motor control in industry [6], pp. 76, [7, 8], is to employ an internal electric current loop. In the simplest version, this internal loop employs a proportional electric current controller. This means that the control signal is computed as:

$$u_i = K(i^* - i), \tag{10.14}$$

where K is a positive constant standing for the proportional electric current controller gain, whereas i^* represents the commanded current that is given as the output of another controller (the external loop) intended to control a mechanical variable, i.e., velocity or position. The design of this external loop is explained later. For now, just consider that i^* is the desired value for the current through the motor, which is specified below. Using (10.13) and (10.14) it is found that voltage applied at the motor armature terminals is given as:

$$u = A_p \, K(i^* - i). \tag{10.15}$$

Applying the Laplace transform to (10.15), the following is found:

$$U(s) = A_p\ K(I^*(s) - I(s)), \tag{10.16}$$

where $I^*(s)$ is the Laplace transform of i^*. Combining (10.11), (10.12), and (10.16), the block diagram in Fig. 10.2a is found. As demonstrated in the Example 4.4, Sect. 4.1, Fig. 10.2a can be simplified as shown in Figs. 10.2b and c. It is also revealed in Example 4.4, Sect. 4.1, that from Fig. 10.2c, the following can be written:

$$\theta(s) = G_1(s)I^*(s) + G_2(s)T_p(s), \tag{10.17}$$

where:

$$G_1(s) = \frac{nk_m}{\left[\left(\frac{sL+R}{KA_p} + 1\right)(sJ + b) + \frac{n^2k_mk_e}{KA_p}\right]s}, \tag{10.18}$$

$$G_2(s) = \frac{-\left(\frac{sL+R}{KA_p} + 1\right)}{\left[\left(\frac{sL+R}{KA_p} + 1\right)(sJ + b) + \frac{n^2k_mk_e}{KA_p}\right]s}. \tag{10.19}$$

If the proportional gain K is chosen to be large enough, then it can be assumed that [6], pp. 76:

$$\frac{sL + R}{KA_p} \approx 0, \qquad \frac{n^2k_mk_e}{KA_p} \approx 0,$$

to approximate (10.18) and (10.19) as:

$$G_1(s) = \frac{nk_m}{s^2J + sb}, \tag{10.20}$$

$$G_2(s) = \frac{-1}{s^2J + bs}. \tag{10.21}$$

Using (10.20) and (10.21) in (10.17), the following is found:

$$\theta(s) = \frac{1}{s^2J + sb}[nk_mI^*(s) - T_p(s)], \tag{10.22}$$

which is represented by the block diagram in Fig. 10.2d. Note that the objective of the electric current control shown in (10.15) is to render the motor's electrical dynamics negligible such that the motor model is now represented only by the motor mechanical subsystem. This can be clearly seen if (10.22) and (10.12) are compared. These expressions indicate that electric current can now be used as the control variable instead of voltage. It is clear that voltage is still the variable that is directly manipulated, as indicated in (10.15); however, for analysis and

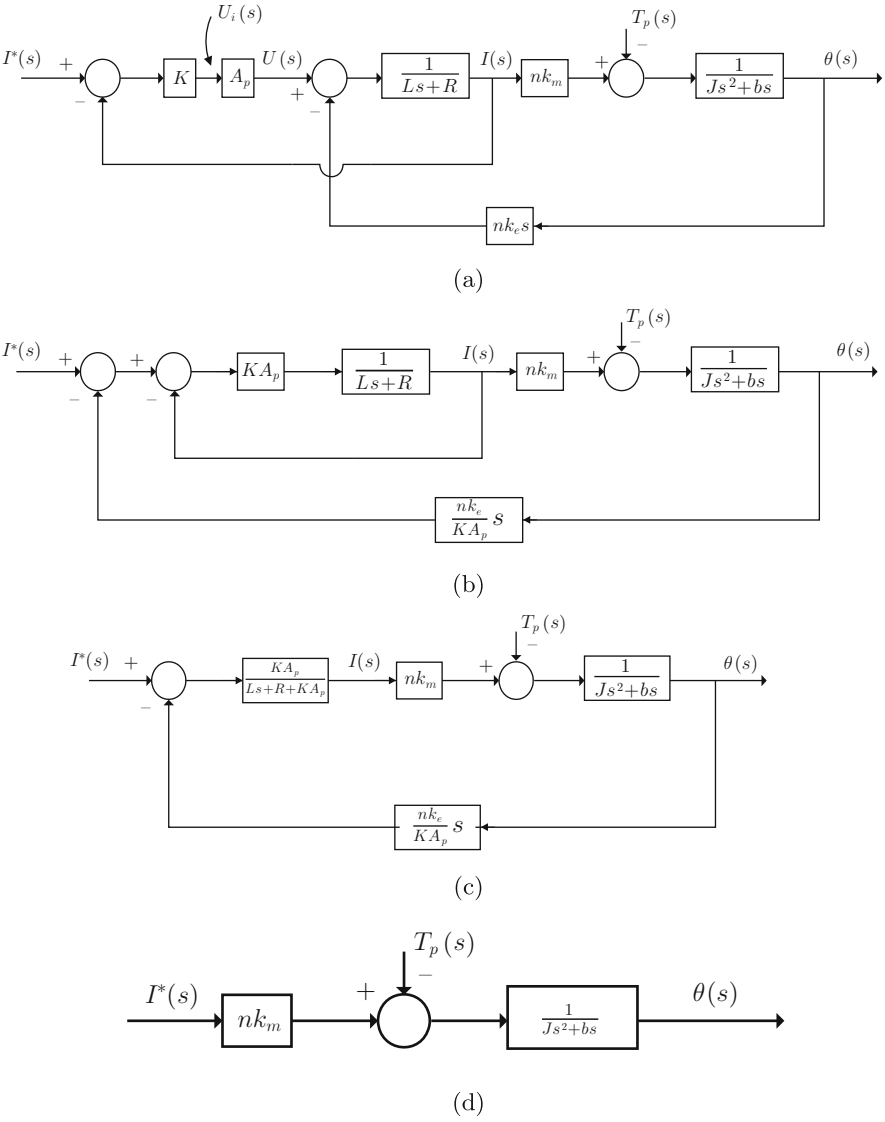

Fig. 10.2 Equivalent block diagrams for a PM brushed DC motor provided with a proportional current loop and a power amplifier

design purposes it can be assumed that electric current is the variable that is directly manipulated through the commanded current i^*. The purpose of the electric current control (10.15) is to accomplish that the actual electric current i tracks the commanded current i^*. A large value of K is very useful for this. Note that the expression in (10.22) can be rewritten as:

$$\theta(s) = \frac{1}{s(s+a)}[kI^*(s) - \frac{1}{J}T_p(s)] \tag{10.23}$$
$$a = \frac{b}{J}, \quad k = \frac{nk_m}{J}.$$

Finally, as velocity ω and position θ are related through $\omega = \frac{d\theta}{dt}$, then the use of the Laplace transform under zero initial conditions yields $\omega(s) = s\theta(s)$. Thus, (10.23) can be rewritten as:

$$\omega(s) = s\theta(s) = \frac{1}{s+a}[kI^*(s) - \frac{1}{J}T_p(s)]. \tag{10.24}$$

In the remainder of this chapter, the model in (10.24) is used to design velocity controllers. Some experimental results obtained with these controllers are also presented.

10.4 Identification

In this section we are interested in performing an experiment that allows us to compute the values of the constants k and a present in the model (10.24). This experiment is described in the following.

Suppose that there is no external disturbance, i.e., $T_p(s) = 0$; hence, the model in (10.24) becomes:

$$\omega(s) = \frac{k}{s+a}I^*(s). \tag{10.25}$$

This is a first-order system such as that studied in Sect. 3.1. Hence, if $i^* = A$ is applied, i.e., $I^*(s) = \frac{A}{s}$, then velocity ω evolves over time as shown in Fig. 10.3, i.e.,

$$\omega(t) = \frac{kA}{a}\left(1 - e^{-at}\right). \tag{10.26}$$

The time constant τ can be measured as shown in Fig. 10.3 and, using $\tau = \frac{1}{a}$, the numerical value of a can be computed. On the other hand, defining the final value of the velocity as $\omega_f = \lim_{t\to\infty}\omega(t) = \frac{kA}{a}$ and, using the value of a just computed, $k = \frac{a\omega_f}{A}$ can be found.

Some results obtained when performing the experiment described above are shown in Fig. 10.4. The noise content in velocity measurements is evidence for a well-known fact in control systems: velocity is a noisy signal. See Sect. 10.6 for an explanation of why ω is measured in volts. In Fig. 10.4, $i^* = A = 0.3$[A] is used and by direct measurement in Fig 10.4, the following is found:

$$\tau = 2.7[\text{s}], \quad \omega_f = 2[\text{V}]. \tag{10.27}$$

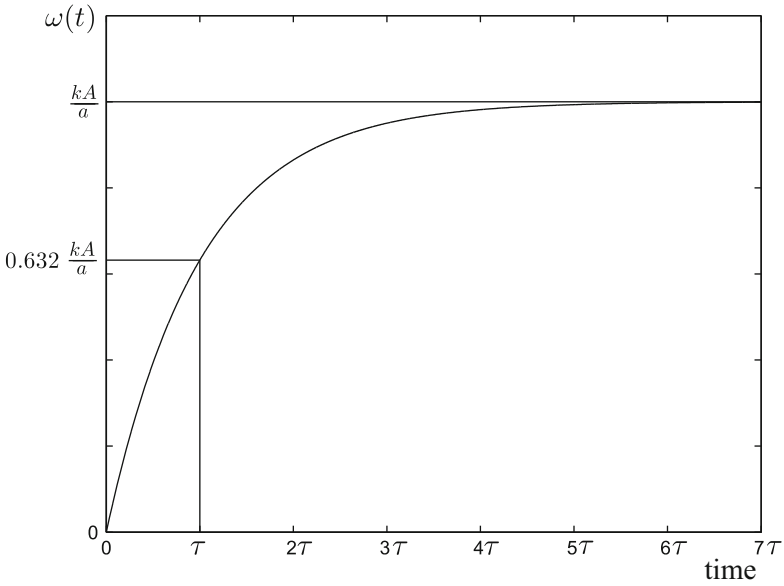

Fig. 10.3 Velocity evolution in a PM brushed DC motor when a constant $i^* = A$ is applied

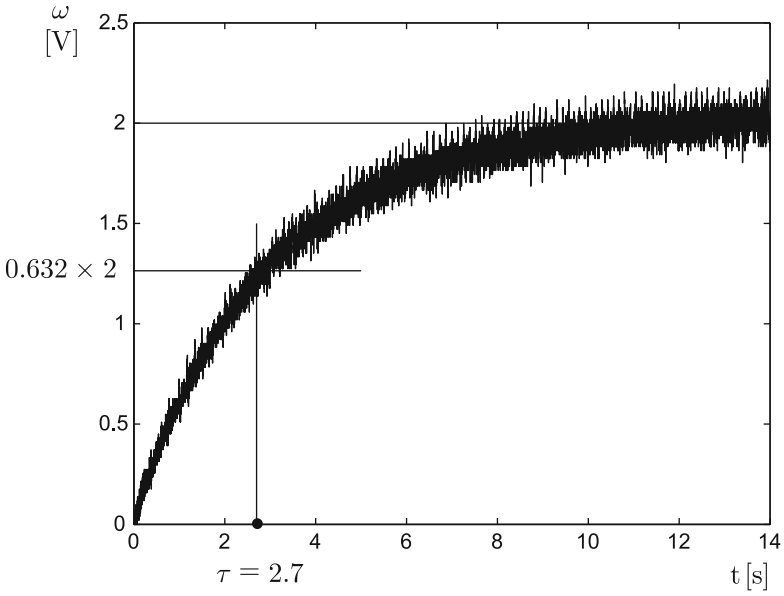

Fig. 10.4 Experimental identification of a PM brushed DC motor

These measurements are simplified and rendered more exact if a computer program is employed to perform them. Using these values and the procedure described above,

the following is found:

$$a = 0.3704, \qquad k = 2.4691. \tag{10.28}$$

Finally, it is not important to compute the numerical value of $1/J$, appearing as the coefficient of the disturbance $T_p(s)$ in (10.24). The reason for this is that T_p is commonly unknown and, despite this, its effect can be compensated for.

10.5 Velocity Control

The main objective of control design is to ensure that the desired transient and steady-state response specifications are achieved. The transient response specifications refer to how fast the closed-loop system is desired to respond. This is achieved by suitably assigning the closed-loop poles. On the other hand, the steady-state specifications refer to the difference that is desired to exist between the measured velocity ω and the desired velocity ω_d when time increases. Moreover, this must stand despite the presence of the external disturbance $T_p(s)$. The controller is the device responsible for ensuring that all these specifications are satisfied.

Suppose that the desired velocity is a constant, i.e., a step function. Then, ω_d is a constant that may be positive, negative or zero. As the plant in (10.24) has no poles at $s = 0$, then the system type is 0. This means that the use of a proportional velocity controller with the form $i^* = k_p(\omega_d - \omega)$, with $k_p > 0$, cannot achieve $\omega = \omega_d$ in a steady state even if no external torque disturbance is present, i.e., even when $T_p = 0$. The solution to this problem is a controller with an integral action to render the system type 1, i.e., a proportional–integral (PI) controller. However, as explained in Sect. 5.2.4, a PI controller cannot simultaneously achieve the desired transient response specifications and a satisfactory rejection of external disturbances $T_p(s)$. It is for this reason that the design of a modified PI velocity controller is presented in the following, which is capable of simultaneously achieving these requirements.

10.5.1 A Modified PI Controller

Applying the inverse Laplace transform to (10.24), the following is found:

$$\dot{\omega} + a\omega = ki^* - \frac{1}{J}T_p. \tag{10.29}$$

In the following, it is shown that the use of the controller:

$$i^* = k_p(\omega_d - \omega) + k_i \int_0^t (\omega_d - \omega(r))dr + \frac{a}{k}\omega_d + k_1(\omega(0) - \omega_d), \tag{10.30}$$

where $\omega(0)$ is the initial velocity, ensures that: 1) The motor velocity ω reaches the desired constant velocity ω_d in a steady state. 2) This is achieved with a time constant that is arbitrarily assigned. 3) The effects of a constant disturbance T_p are compensated for as fast as desired.

Replacing (10.30) in (10.29) and rearranging:

$$\dot{\omega} + (a + k_p k)(\omega - \omega_d) + k_i k \int_0^t (\omega(r) - \omega_d) dr = k_1 k(\omega(0) - \omega_d) - \frac{1}{J} T_p.$$

Defining:

$$k_p = k'_p + k_1,$$

the following can be written:

$$\dot{\omega} + (a + k'_p k)(\omega - \omega_d) + k_i k \int_0^t (\omega(r) - \omega_d) dr$$
$$+ k_1 k[\omega - \omega_d - (\omega(0) - \omega_d)] = -\frac{1}{J} T_p.$$

Note that:

$$\omega - \omega_d - (\omega(0) - \omega_d) = \int_0^t (\dot{\omega}(r) - \dot{\omega}_d) dr,$$

and defining:

$$k_i = k'_i k_1,$$

then:

$$\dot{\omega} + (a + k'_p k)(\omega - \omega_d) + k_1 k \int_0^t \left[k'_i (\omega(r) - \omega_d) + (\dot{\omega}(r) - \dot{\omega}_d) \right] dr$$
$$= -\frac{1}{J} T_p. \tag{10.31}$$

As ω_d is a constant, then $\dot{\omega}_d = 0$. Hence, if:

$$k'_i = a + k'_p k,$$
$$\xi = \int_0^t \left[\dot{\omega}(r) + k'_i (\omega(r) - \omega_d) \right] dr,$$

are defined, then (10.31) can be written as:

$$\dot{\xi} + k_1 k \xi = -\frac{1}{J} T_p.$$

Applying the Laplace transform with zero initial conditions:

$$\xi(s) = \frac{-\frac{1}{J}}{s + k_1 k} T_p(s),$$

or:

$$s\xi(s) = \frac{-\frac{1}{J}s}{s + k_1 k} T_p(s). \tag{10.32}$$

If $T_p = t_d$ is a constant, i.e., $T_p(s) = \frac{t_d}{s}$, then:

$$s\xi(s) = \frac{-\frac{1}{J}s}{s + k_1 k} \frac{t_d}{s}.$$

Applying the final value theorem:

$$\begin{aligned} \lim_{t\to\infty} \dot{\xi}(t) &= \lim_{s\to 0} s\left[s\xi(s)\right], \\ &= \lim_{s\to 0} s \left[\frac{-\frac{1}{J}s}{s + k_1 k} \frac{t_d}{s}\right] = 0. \end{aligned}$$

Hence, if $k_1 k > 0$, the filter in (10.32) is stable and $\lim_{t\to\infty} \dot{\xi}(t) = 0$ is ensured. Furthermore, $\dot{\xi}(t)$ approaches zero faster as $k_1 k > 0$ is larger. Note that $\lim_{t\to\infty} \dot{\xi}(t) = 0$ also implies that:

$$\frac{d}{dt} \int_0^t \left[\dot{\omega}(r) + k_i'(\omega(r) - \omega_d)\right] dr = \dot{\omega} + k_i'(\omega - \omega_d) \to 0.$$

This means that the effect of the constant disturbance disappears as time increases and the remaining differential equation $\dot{\omega} + k_i'\omega = k_i'\omega_d$, which is stable if $k_i' > 0$, possesses the time constant $\tau = \frac{1}{k_i'}$ and has a unit gain in a steady state. This means that:

- If no perturbation exists ($T_p = 0$ and $\dot{\xi}(t) = 0$ for all $t \geq 0$), the velocity response is as that of a first-order system with time constant $\tau = \frac{1}{k_i'}$. The velocity ω reaches the desired velocity ω_d (which is constant) in a steady state.
- If a constant disturbance appears ($T_p = t_d \neq 0$), the deviation produced by such a disturbance vanishes as time increases. Moreover, if a larger k_1 is chosen (such that the product $k_1 k > 0$ is larger), then the deviation due to the disturbance vanishes faster. If once the deviation due to the disturbance is taken to zero (when $\omega = \omega_d$), a step change is commanded in the desired velocity ω_d (a change in ω_d is commanded when the disturbance T_p is present), then the velocity responds as if the disturbance T_p is not present, i.e., as in the previous item: with $\tau = \frac{1}{k_i'}$ as the time constant and velocity ω reaches its desired value ω_d in a steady state.

As the initial conditions are always assumed to be zero in classical control, then $\omega(0) = 0$ can be assumed and the controller in (10.30) becomes:

$$i^* = k_p(\omega_d - \omega) + k_i \int_0^t (\omega_d - \omega(r))dr + \left(\frac{a}{k} - k_1\right)\omega_d, \tag{10.33}$$

which constitutes a simple PI controller with a constant feedforward term. Finally, the tuning rule is summarized by:

$$\begin{aligned} k_p &= k_p' + k_1, \\ k_i &= k_i' k_1, \\ k_i' &= a + k_p' k, \end{aligned} \tag{10.34}$$

where:

- $k_p' > 0$ is chosen to define the desired time constant $\tau = \frac{1}{k_i'} = \frac{1}{a + k_p' k}$, which is always less than the time constant of the plant to be controlled $\frac{1}{a}$, i.e., a very convenient feature.
- $k_1 > 0$ is chosen large for a fast rejection of the disturbance effects. This can be done by trial and error or it can be computed recalling that $\frac{1}{k_1 k}$ is the time constant of the filter in (10.32), which is responsible for the elimination of the deviation due to the disturbance.

Another, better known, way of designing a PI controller with the same advantages found above is by using two-degrees-of-freedom controllers. These are presented in the following section.

10.5.2 A Two-Degrees-of-Freedom Controller

A closed-loop system with a two-degrees-of-freedom controller is shown in Fig. 10.5. The controller has two components with transfer functions $G_{c1}(s)$ and $G_{c2}(s)$, whereas $G_p(s)$ is the plant to be controlled. An external disturbance $D(s)$ applied at the plant input is also considered. In Example 4.5, Sect. 4.1, it is shown that the output of the closed-loop system is given as:

$$\omega(s) = G_1(s)\omega_d(s) + G_2(s)D(s),$$

where:

$$\begin{aligned} G_1(s) &= \frac{G_{c1}(s)G_p(s)}{1 + (G_{c1}(s) + G_{c2}(s))G_p(s)}, \\ G_2(s) &= \frac{G_p(s)}{1 + (G_{c1}(s) + G_{c2}(s))G_p(s)}, \end{aligned} \tag{10.35}$$

are the transfer functions defined as:

$$G_1(s) = \frac{\omega(s)}{\omega_d(s)}, \quad \text{when } D(s) = 0,$$
$$G_2(s) = \frac{\omega(s)}{D(s)}, \quad \text{when } \omega_d(s) = 0.$$

The reason why this control scheme is called *two-degrees-of-freedom control* is that the transfer functions defined in (10.35) can be tuned by maintaining them independent from each other using the two controller components $G_{c1}(s)$ and $G_{c2}(s)$ as independent variables, i.e., as degrees-of-freedom. As a PM brushed DC motor is to be controlled in this chapter, assume that:

$$G_p(s) = \frac{k}{s+a}.$$

Also assume that $G_{c1}(s)$ and $G_{c2}(s)$ are PI controllers, i.e.,

$$G_{c1}(s) = \frac{k_{p1}s + k_{i1}}{s}, \quad G_{c2}(s) = \frac{k_{p2}s + k_{i2}}{s},$$
$$G_c(s) = G_{c1}(s) + G_{c2}(s) = \frac{k_p s + k_i}{s}, \quad k_p = k_{p1} + k_{p2}, \quad k_i = k_{i1} + k_{i2}.$$

Hence, the first transfer function in (10.35) can be written as:

$$\begin{aligned}\frac{\omega(s)}{\omega_d(s)} &= \frac{G_{c1}(s)\frac{k}{s+a}}{1 + \frac{k_p s + k_i}{s}\frac{k}{s+a}}, \\ &= \frac{skG_{c1}(s)}{s^2 + (a + k_p k)s + k_i k}. \end{aligned} \tag{10.36}$$

Suppose that the closed-loop response is desired to be as that of a first- order with a pole at $s = -p_1$, with $p_1 > 0$. Then, the characteristic polynomial in the last expression must satisfy:

$$\begin{aligned}s^2 + (a + k_p k)s + k_i k &= (s + p_1)(s + f), \\ &= s^2 + (p_1 + f)s + p_1 f,\end{aligned}$$

where the root at $s = -f$, $f > 0$, is introduced merely to obtain a second-degree polynomial on the right-hand side of this expression. Equating the coefficients on both sides, the following is found:

$$a + k_p k = p_1 + f, \quad k_i k = p_1 f,$$

i.e.,

$$k_p = \frac{p_1 + f - a}{k}, \quad k_i = \frac{p_1 f}{k}.$$

On the other hand, to obtain a response of a first order with a pole at $s = -p_1$, the transfer function in (10.36) must satisfy:

$$\frac{\omega(s)}{\omega_d(s)} = \frac{p_1(s+f)}{s^2 + (a + k_p k)s + k_i k} = \frac{p_1(s+f)}{(s+p_1)(s+f)} = \frac{p_1}{s+p_1}. \quad (10.37)$$

Thus, equating (10.36) and (10.37), the following is found:

$$skG_{c1}(s) = p_1(s+f).$$

It is concluded that $G_{c1}(s)$ is a PI controller:

$$G_{c1}(s) = \frac{k_{p1}s + k_{i1}}{s}, \quad k_{p1} = \frac{p_1}{k}, \quad k_{i1} = \frac{p_1 f}{k}.$$

Based on this result, $G_{c2}(s)$ can be computed as:

$$G_{c2}(s) = G_c(s) - G_{c1}(s) = \frac{k_{p2}s + k_{i2}}{s}, \quad k_{p2} = \frac{f-a}{k}, \quad k_{i2} = 0.$$

This means that $G_{c2}(s) = k_{p2}$ is a proportional controller. On the other hand, the second transfer function in (10.35) becomes:

$$\begin{aligned} \frac{\omega(s)}{D(s)} &= \frac{\frac{k}{s+a}}{1 + \frac{k_p s + k_i}{s}\frac{k}{s+a}}, \\ &= \frac{ks}{s^2 + (a + k_p k)s + k_i k}, \\ &= \frac{ks}{(s+p_1)(s+f)}. \end{aligned} \quad (10.38)$$

Using this expression and the final value theorem, it can be verified that the effect of a constant external disturbance $D(s) = -\frac{t_d}{Jks}$ vanishes as time increases. Furthermore, this is achieved faster as f and p_1 are larger. As p_1 is fixed by the desired response to a constant velocity command, then f is the free parameter. Thus, f must be chosen to be large. According to Fig. 10.5, the controller is given as:

$$\begin{aligned} I^*(s) &= G_{c1}(s)(\omega_d(s) - \omega(s)) - G_{c2}(s)\omega(s), \\ &= k_{p1}(\omega_d(s) - \omega(s)) + k_{i1}\frac{(\omega_d(s) - \omega(s))}{s} - k_{p2}\omega(s), \end{aligned}$$

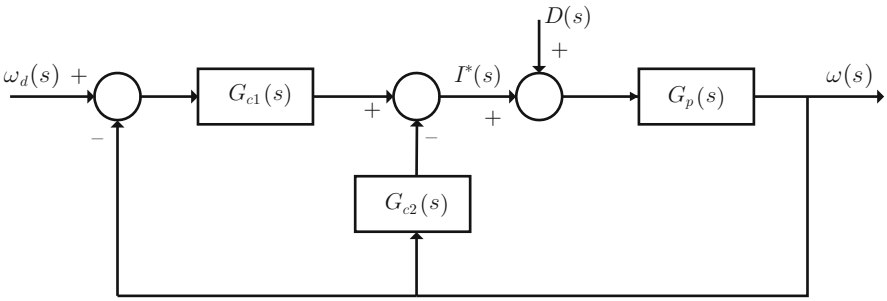

Fig. 10.5 Closed-loop system with a two-degrees-of-freedom controller

$$= (k_{p1} + k_{p2})(\omega_d(s) - \omega(s)) + k_{i1}\frac{(\omega_d(s) - \omega(s))}{s} - k_{p2}\omega_d(s),$$

$$= k_p(\omega_d(s) - \omega(s)) + k_{i1}\frac{(\omega_d(s) - \omega(s))}{s} - k_{p2}\omega_d(s). \tag{10.39}$$

Using the inverse Laplace transform, the following is finally obtained:

$$i^* = k_p(\omega_d - \omega) + k_{i1}\int_0^t (\omega_d - \omega(r))dr - k_{p2}\omega_d. \tag{10.40}$$

Note that controllers in (10.40) and (10.33) are identical if:

$$k_1 = \frac{f}{k}, \quad p_1 = k_i',$$

are defined. Thus, it is up to the reader to decide which of the approaches presented in Sect. 10.5.1 or 10.5.2 is preferred.

10.6 Experimental Prototype

The following are the main components of the control system.

- Microcontroller PIC16F877A [9].
- Digital/analog converter, DAC0800LCN, 8 bits.
- 2 PM brushed DC motors. Nominal voltage 24[V], nominal current 2.3[A].

Both motor shafts are coupled such that one motor is employed as a generator. The generated voltage is used as measurement of the other motor's velocity which is controlled according to the control law in (10.33).

The electric diagram of the complete velocity control system is shown in Fig. 10.6. The control algorithm in (10.33) is computed by a microcontroller

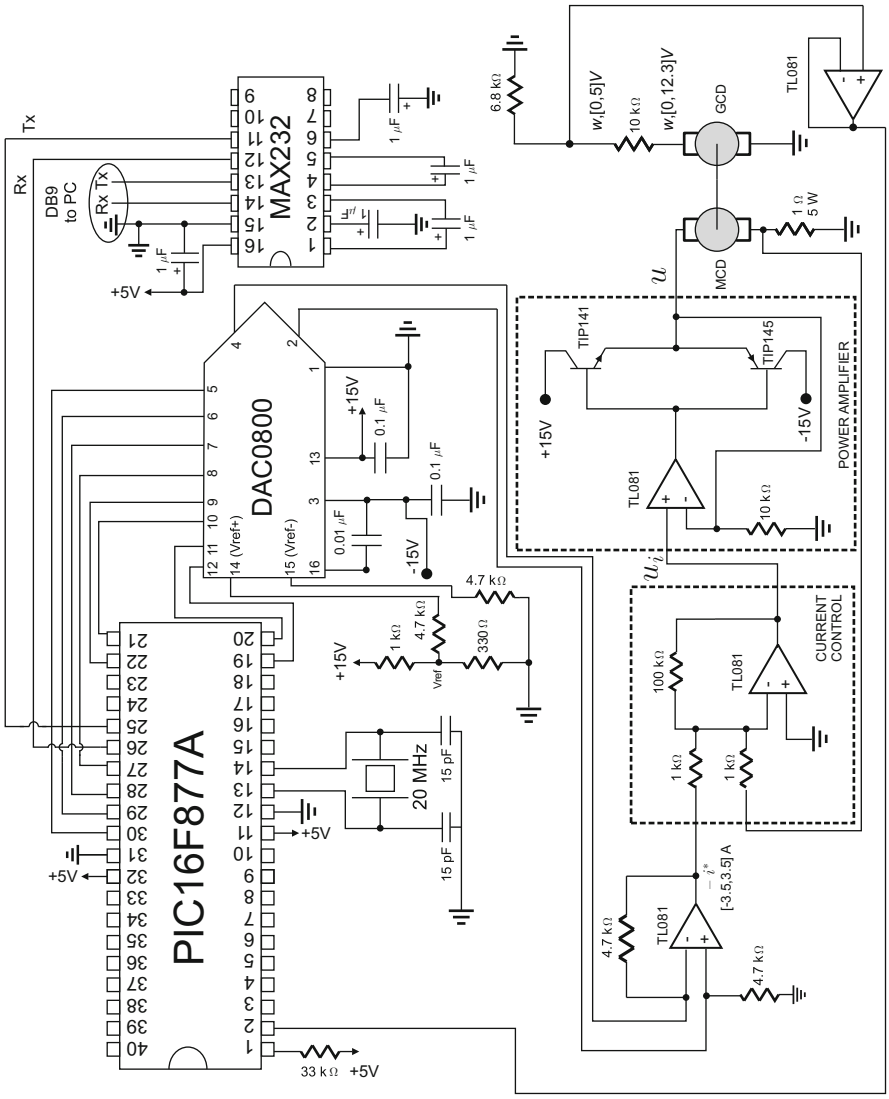

Fig. 10.6 Electric diagram of the complete velocity control system for a PM brushed DC motor

PIC16F877A. Once the desired velocity ω_d and the velocity measurement ω are known, the microcontroller computes the control signal i^* in (10.33). The microcontroller program is listed in Sect. 10.9.

In the following, the way in which the microcontroller PIC16F877A can be used as a controller is described. It is stressed that it is not the purpose to present detailed information on the microcontroller operation. A simple description is presented on how the components of this microcontroller are employed to implement the control law being tested.

The motor velocity is measured through canal 0 of the microcontroller analog/digital converter. Only the eight highest bits of the converter are employed. The input analog range of the converter is [0, +5][V]. Hence, the datum delivered by the converter t_0 is used to compute w=0.0196*t_0 (where 0.0196 = 5/255) to retrieve, as the variable w, voltage at the analog input of the analog/digital converter. This voltage is delivered by the motor used as a generator and it is proportional to the velocity of the motor to be controlled. Hence, it is assumed that this voltage represents the measured velocity. Note that before entering the analog/digital converter (through microcontroller pin 2), the generator voltage passes through a tension divisor used to adapt the maximal generated voltage (12.3[V] at maximal velocity) to 5[V]. This is the upper limit of the analog input of the analog/digital converter. The operational amplifier TL081 is used to protect the microcontroller.

The controller in (10.33) is computed with the instructions "iast=kp*error+ki*ie +cte*wd; ie=ie+0.002*error;," where the second expression represents the integral of the velocity error. See Appendix F for an explanation of why this expression represents a discrete iterative version of the integral on the velocity error. As indicated in Fig. 10.6, $-i^*$ must appear at the input of the operational amplifier TL081 inside the part entitled "current control." This is accomplished as follows. First, iast=-iast is computed in the program listed in Sect. 10.9, to change the sign of i^*. The microcontroller must deliver an 8-bit digital code (iastd) to the 8-bit digital/analog converter DAC0800LCN. Hence, the microcontroller must compute:

$$\text{iastd} = 36.4286 * \text{iast} + 127,$$

which results from the graphical relation presented in Fig. 10.7. Once "iastd" is computed, it is delivered to the digital/analog converter through port D (PORTD=iastd). The digital/analog converter works together with an operational amplifier TL081. These devices are connected following the manufacturer's suggestions [10]. This ensures that, at the operational amplifier output, a voltage is obtained whose numerical value corresponds to $-i^*$. This voltage is received by another operational amplifier TL081 devoted to implementing the current controller, i.e., to compute:

$$u_i = 100(i^* - i).$$

The reason why $-i^*$ must appear at the indicated place in Fig. 10.6 has to do with the sign operations performed by the operational amplifiers in this figure. On the other hand, a unit gain power amplifier is composed of a third operational amplifier TL081 together with two power transistors TIP141 and TIP145 in a complementary symmetry connection.

The circuit MAX232 is devoted to sending the measured velocity ω and the control signal i^* to a portable computer (through a USB to a series adapter) whose unique purpose is the graphical representation of those variables. This is performed through the variables "cuentaH=0x00" and "cuentaL=t_0" (to send velocity) or "cuentaH=0x00" and "cuentaL=(iastd) &(0xFF)" (to send the control

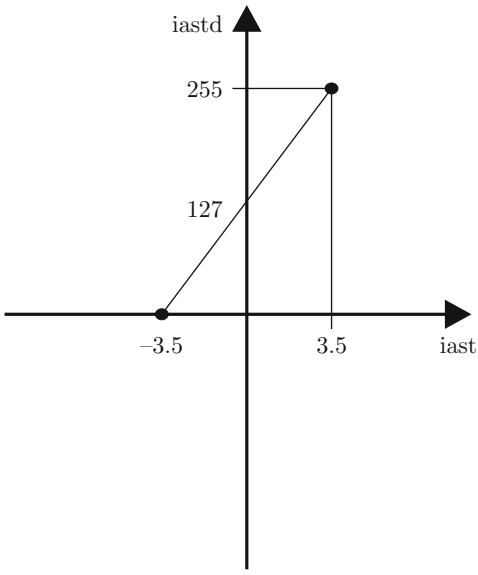

Fig. 10.7 Conditioning of the signal i^*, which must be sent to the digital/analog converter

signal). These data are sent to the portable computer through the instructions: putc(0xAA); putc(cuentaH); putc(cuentaL);. Finally, the timer TMR0 is employed to fix the sample time $T = 0.002$[seg] (40 counts of timer 0).

10.6.1 Electric Current Control

The electrical loop shown in (10.14) is implemented as follows. A 1[Ohm] power resistance is series connected to the motor armature. Hence, voltage between terminals of this resistance is numerically equal to the electric current i flowing through the motor. Then, the operational amplifier at the block "current control" in Fig. 10.6 performs the operation indicated in (10.14) with $K = 100$.

10.6.2 Power Amplifier

According to Sect. 9.1.2, the block labeled "power amplifier," in Fig. 10.6, allows us to reduce the dead zone introduced by the power transistors in a complementary symmetry connection. Moreover, with the indicated values $A_p = 1$ is accomplished in (10.13) and (10.15). This results in $A_pK = 100$. This value has been chosen because a good performance was obtained. It is important to stress that large values such as $A_pK = 700$ are employed in commercial drivers for industrial applications [8].

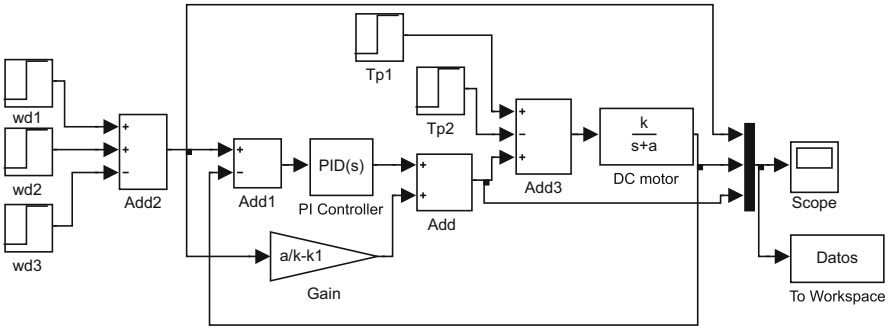

Fig. 10.8 MATLAB/Simulink diagram used to obtain the results in Fig. 10.9

10.7 Simulation Results

Some simulation results obtained with the controller in (10.33) (or, equivalently, with the controller in (10.40)), employing the MATLAB/Simulink diagram in Fig. 10.8 are presented in Fig. 10.9. Starting from a zero initial velocity, the following desired velocity is commanded:

$$\omega_d = \begin{cases} 1.5, & 0 \leq t < 4 \\ 2.5, & 4 \leq t < 12 \\ 1.5, & t \geq 12 \end{cases} . \tag{10.41}$$

Also, the constant disturbance $T_p = 2.5$ is applied at $t = 8$[s], via software, which disappears at $t = 17$[s]. This means that T_p appears as an electric current signal, which adds to the electric current commanded by controller. The DC motor parameters are the same as those in (10.28). The gains of the controller in (10.33) are computed according to (10.34) together with numerical values in (10.28) and:

$$k_p' = 0.5, \quad k_1 = 4.0,$$

, which fixes the desired time constant as $\tau = 0.6231$[s]. The reader can verify that the labels in Fig. 10.9 prove that the desired time constant is accomplished when applying the desired velocity steps. Also note that the effects of the disturbance T_p are kept small and disappear very fast.

The wd and Tp blocks in Fig. 10.8 are suitably programmed to produce the desired velocity defined in (10.41) and the disturbance T_p defined above respectively. The PI controller block has kp and ki as the proportional and integral gains respectively, whereas the To workspace block is programmed to save data as an array with 0.001 as sample time. The graphical results in Fig. 10.9 were obtained by executing the following MATLAB code in an m-file:

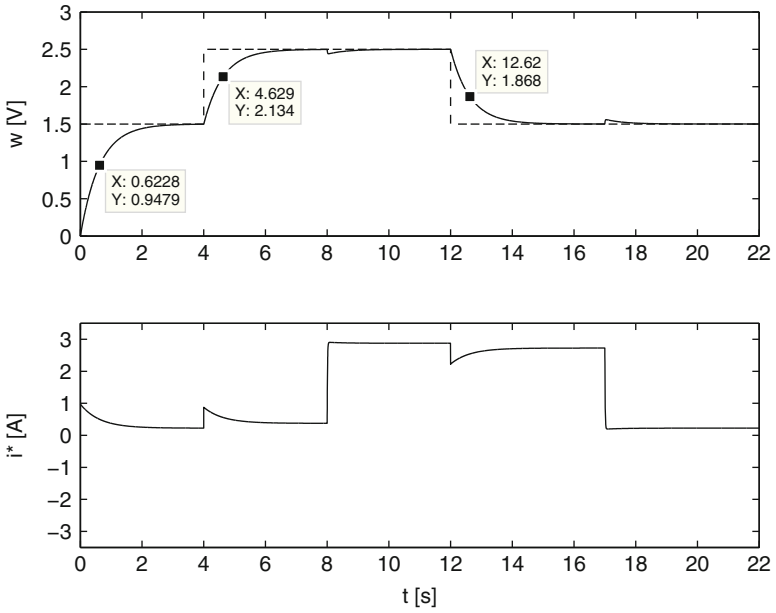

Fig. 10.9 Simulation results when using the controller in (10.33)

```
a=0.3704;
k=2.4691;
T=0.6231; %time constant, sec
kpp=(1/T-a)/k
k1=40;
kp=kpp+k1;
kip=a+kpp*k;
ki=kip*k1;
Tp=2.5;
```

then simulation in Fig. 10.8 is run, and, finally, the following code is executed in an m-file:

```
Ts=0.001;
t=0:Ts:22;
figure(1)
subplot(2,1,1)
plot(t,Datos(:,1),'b--');
hold on
plot(t,Datos(:,2),'b-');
hold off
axis([0 22 0 3])
ylabel('w [V]')
```

```
subplot(2,1,2)
plot(t,Datos(:,3),'b-');
axis([0 22 -3.5 3.5])
xlabel('t [s]')
ylabel('i* [A]')
```

10.8 Experimental Results

Some experimental results obtained with the controller in (10.33) (or, equivalently, with the controller in (10.40)), employing the experimental prototype described in Sect. 10.6, are presented in Fig. 10.10. Starting from a zero initial velocity, the desired velocity in (10.41) is commanded. Also, disturbance T_p and the controller gains are the same as those described in Sect. 10.7. The velocity measured in the experiment is shown in Fig. 10.10. The white line represents the response obtained in simulation for the system without disturbances:

$$\dot{\omega} + k_i'\omega = k_i'\omega_d, \tag{10.42}$$

with $k_i' = \frac{1}{0.6231}$, i.e., the value computed according to (10.34), and ω_d given in (10.41). Note that the velocity experimental response exactly tracks the theoretical response defined by (10.42) and this is still true during the transition from $\omega_d = 2.5$ to $\omega_d = 1.5$ at $t = 12$[s] despite the constant disturbance $T_p = 2.5$ being present during this transition. This corroborates the conclusions stated just before (10.33). Moreover, it is important to stress that the desired transient and steady-state response specifications have been exactly achieved and the deviations observed when an external disturbance appears and disappears vanish very fast. Note that the simulation results in Fig. 10.9 and the experimental results in Fig. 10.10 are very similar.

Next, the results obtained with a classical PI controller are presented to compare with results obtained in Fig. 10.10. The results obtained with a classical PI controller are presented in Fig. 10.11. The controller gains are computed as:

$$\frac{k_i}{k_p} = a, \quad \tau = \frac{1}{k_p k} = 0.6231[\text{s}], \tag{10.43}$$

according to the discussion in Sect. 5.2.4, with τ the desired closed-loop system time constant. The desired velocity ω_d is identical to that shown in (10.41) and a disturbance is applied that is identical to that used in Fig. 10.10, i.e., it is applied via software with a constant value $T_p = 2.5$ at $t = 8$[s] and disappears at $t = 17$[s]. The following is observed:

- Although the transient response is close to the desired one, it is not as good as that in Fig. 10.10 (the thin line represents the desired transient response).

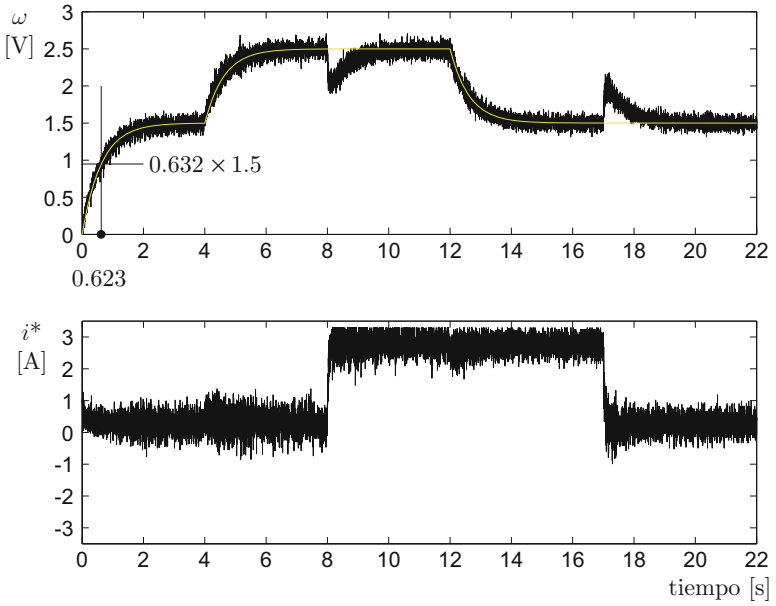

Fig. 10.10 Velocity control using the modified PI controller shown in (10.33) (or, equivalently, with the controller in (10.40))

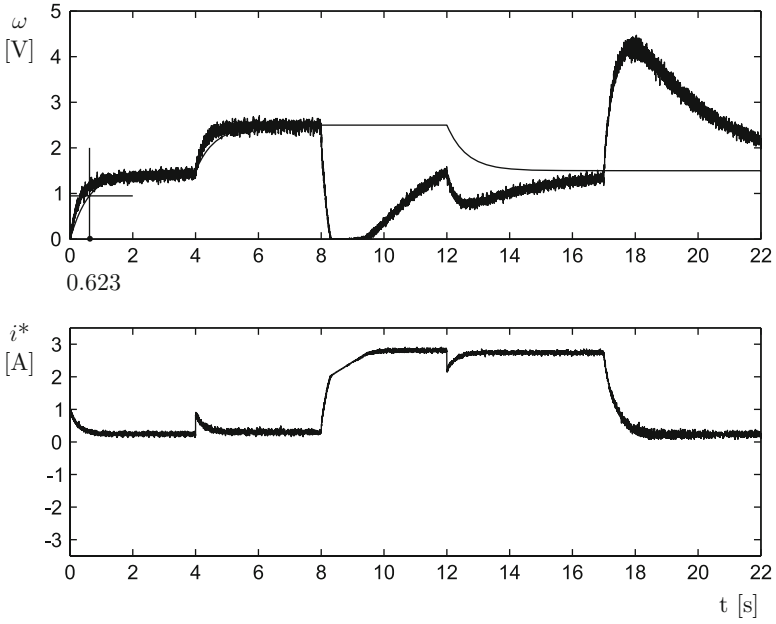

Fig. 10.11 Classical proportional–integral (PI) velocity control using the tuning rule in (10.43)

- The measured velocity reaches the desired velocity in a steady state when no external disturbance is applied, i.e., before $t = 8$[s].
- The deviation produced by the external disturbance is much larger than in Fig. 10.10 and it vanishes much more slowly. In fact, this deviation is not compensated for before a new change in the desired velocity appears at $t = 12$[s]. The same is observed when the disturbance disappears at $t = 17$[s]. According to Sect. 5.2.4, this is because the dynamics of the disturbance rejection has a dominant time constant that is equal to the motor open-loop time constant, $\frac{1}{a}$, which is 2.7[s] as indicated in (10.27).

The experimental results shown in Fig. 10.12 are obtained using a classical PI controller whose gains are chosen according to:

$$\frac{k_i}{k_p} = d,\ d = 1.4 < p_1,\ k_p = \frac{l_2 l_3}{l_1 k}, \tag{10.44}$$

$$l_1 = abs(-p_1 + d),\ l_2 = abs(-p_1),\ l_3 = abs(-p_1 + a),\ p_1 = \frac{1}{0.6231} = 1.6.$$

This tuning criterion is introduced in (5.25), Sect. 5.2.4, where $\tau = 0.6231$[s] is the slowest time constant present in the disturbance rejection dynamics. This time constant is chosen to be equal to the desired time constant when the system responds to the desired velocity because, according to (10.38), this is the slowest time constant that is present in the disturbance rejection shown in Fig. 10.10 ($f > p_1$ for the numerical values employed). The desired velocity ω_d is identical to that shown in (10.41) and a disturbance is applied that is identical to that used in Fig. 10.10, i.e., it is constant with value $T_p = 2.5$, it is applied via software at $t = 8$[s], and disappears at $t = 17$[s]. The following is observed:

- The deviation due to the disturbance vanishes almost as fast as in Fig. 10.10.
- Although the transient response to the reference of velocity is very fast, it does not correspond to the desired transient response represented by the thin line in Fig. 10.12. As explained in Sect. 5.2.4, this is because the closed-loop system has two poles, one assigned at $s = -p_1$, but the other is far to the left of this point. Moreover, the pole located at $s = -\frac{1}{0.6231} = -1.6 = -p_1$ is very close to the zero at $s = -d = -1.4$, i.e., its effects are significantly compensated for (cancelled) and only the dynamics of the fast pole located far to the left is observed.

The experimental results in Figs. 10.11 and 10.12 corroborate the observations presented in Sect. 5.2.4 on classical PI velocity control: there is no tuning rule allowing us to compute the proportional and integral gains to simultaneously achieve the desired specifications for the transient response to a velocity reference and for a satisfactory disturbance rejection. This motivates the design of the modified PI controller presented in (10.33) whose experimental results are shown in Fig. 10.10.

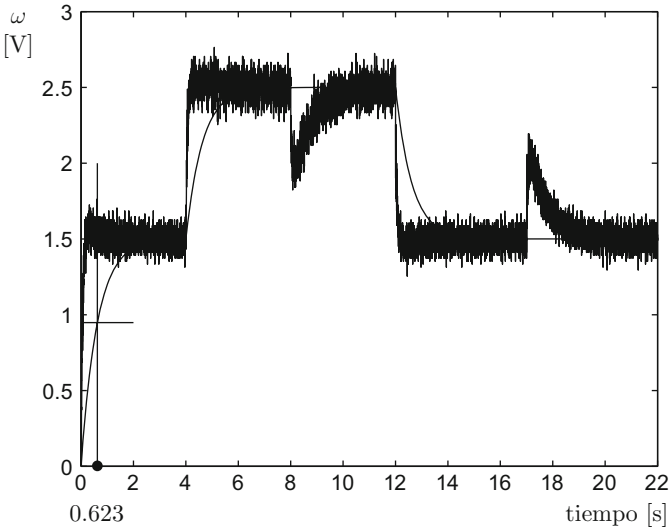

Fig. 10.12 Classical PI velocity control using the tuning rule in (10.44)

10.9 Microcontroller PIC16F877A Programming

The flow diagram for the Microcontroller PIC16F877A programming when controlling velocity in a PM brushed DC motor is presented in Fig. 10.13. The complete code is listed in the following. The reader is referred to [11] for a precise explanation of each one of the instructions in the program.

```
// PM brushed DC motor velocity control microcontroller program
#include<16f877A.h>
#include<stdlib.h>
#include<math.h>

#fuses HS,NOWDT,PUT,NOBROWNOUT,NOLVP,NOWRT,NOPROTECT,NOCPD

#use delay(clock=20000000) //time basis
// Config. series port
#use rs232(baud=115200,XMIT=PIN_C6,RCV=PIN_C7,BITS=8,PARITY=N)
// Ports and registers addresses
#byte OPTION= 0x81
#byte TMR0 = 0x01
#byte PORTA = 0x05

#byte PORTB = 0x06

#byte PORTC = 0x07
#byte PORTD = 0x08
#byte PORTE = 0x09
```

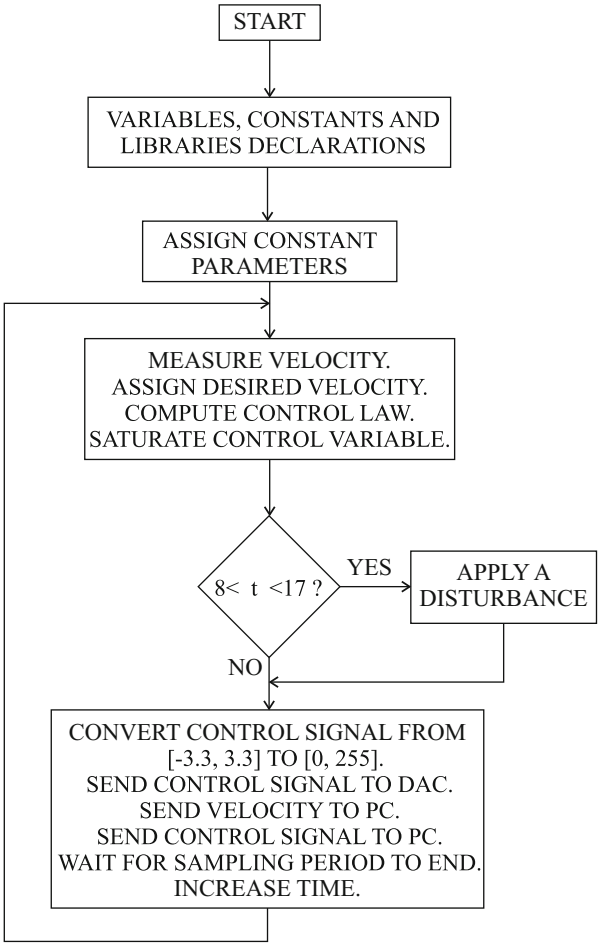

Fig. 10.13 Flow diagram for microcontroller PIC16F877A programming when controlling velocity in a PM brushed DC motor

```
#byte ADCON0= 0x1F
#byte ADCON1= 0x9F
#bit PC0 = 0x07.0

#bit PC1 = 0x07.1
//------------Variables declaration------------//
int16 inter,cuenta;

int8 cuentaH,cuentaL,puerto,AB,AB_1,aux;

int8 cont,iastd,cont2,cont3,cont4,t_0;
```

```
float w,error,iast,wm1,kp,ki,ie,k,a,wd,tiempo,kpp,k3,kib,cte;
unsigned int u,i;
//------------Interrupt subroutine------------//
#int_rb void rb_isr() {
puerto=PORTB;
AB=((puerto)&(0x30))>>4;
aux=AB^AB_1;
if(aux!=0)
if(aux!=3)
if(((AB_1<<1)^AB)&(0x02))
cuenta--;
else
cuenta++;
AB_1=AB;
}
//------------Main program------------//
void main(void) {
setup_adc(ADC_CLOCK_INTERNAL ); //ADC (internal clock)
set_tris_a(0b11111111);
set_tris_b(0b11111111);
set_tris_c(0b10000000); //Serial communication
set_tris_d(0b00000000);
set_tris_e(0b11111111);
OPTION=0x07; //pre_scaler timer0, 1:256
PORTC=0;
TMR0=0;
cuenta=0;
AB=0;
AB_1=0;
enable_interrupts(global);
enable_interrupts(int_rb);
cont=4;
cont2=0;
cont3=0;
cont4=0;
PORTD=127;
k=2.4691;
a=1/2.7;
kpp=0.5;
k3=4.0;
kp=kpp+k3;
kib=a+k*kpp;
ki=kib*k3;
cte=a/k-k3;
tiempo=0.0;
wm1=0.0;
ie=0.0;
ADCON1=0x04;
ADCON0=0x81;
while(cont2<3)
{
while(cont3<255)
{
TMR0=0;
```

```
while(TMR0<255)
{
}
cont3++;
}
cont2++;
}
TMR0=0;
while(TRUE)
{
cont++;
ADCON0=0x81; //Changing to CH0
for(i=0;i<37;i++); //Stabilizing internal capacitor
t_0=read_adc(); //Read ADC
w=0.0196*t_0;
if(tiempo<4.0)
wd=1.5;
else
if(tiempo<12.0)
wd=2.5;
else
wd=1.5;
// wd=2.5;
error=wd-w;
/* Controller */
iast=kp*error+ki*ie+cte*wd;
ie=ie+0.002*error;
/* _____________________________________________ */
if(iast<-3.3)
iast=-3.3;
if(iast>3.3)
iast=3.3;
// iast=0.3;
iast=-iast;
if(tiempo>8.0)
{
if(tiempo<17.0)
iast=iast+2.5;
}
iastd=(unsigned int)(36.4286*iast+127);
PORTD=iastd;
wm1=w;
/*
// inter=(cuenta)&(0xFF00);
cuentaH=0x00; //inter>>8;
cuentaL=t_0;//(cuenta)&(0x00FF);
putc(0xAA); //Recognition serial port
putc(cuentaH); //Sending to serial port
putc(cuentaL);
if(cont==5)
{
cont=0;
cont4++;
if(cont4==255)
```

```
cont4=0;
}
*/
if(tiempo>8.0)
{
if(tiempo<17.0)
iast=iast-2.5;
}
iast=-iast;
iastd=(unsigned int)(36.4286*iast+127);
// inter=(cuenta)&(0xFF00);
cuentaH=0x00; //inter>>8;
cuentaL=(iastd)&(0xFF);
putc(0xAA); //Recognition serial port
putc(cuentaH); //sending to serial port
putc(cuentaL);
if(cont==5)
{
cont=0;
cont4++;
if(cont4==255)
cont4=0;
}
PC1=1; //waiting for sampling time
while(TMR0<40) // 1 count=(4/FXtal)*256 seg, 40=2 ms
{
}
PC1=0; //sampling time has arrived
TMR0=0;
tiempo=tiempo+0.002;
} //closing the infinite while
} //closing mian
```

10.10 Frequency Response-Based Design

10.10.1 Model Identification

Another experimental prototype for velocity control in a PM brushed DC motor has been built. The identification of the corresponding model has been performed using a frequency response experiment. The details on the steps that have been followed to perform this task are given in the following.

1. The following voltage signal has been applied at the motor armature terminals $u = 2 + A\sin(\bar{\omega}t)$[V] and the measured velocity of the motor has been observed to have the following form $\omega = \omega_0 + B\sin(\bar{\omega}t + \phi)$[rad/s], where ω_0 is a constant. The variable ω is used to designate the motor angular velocity and $\bar{\omega}$ to designate the angular frequency. ω is used to designate any of these variables when it is evident from the context which of these variables is referred to.

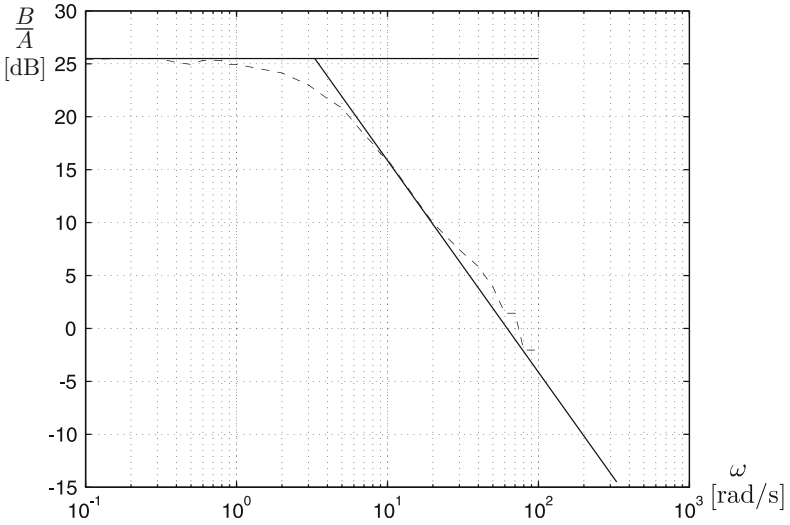

Fig. 10.14 Bode diagrams from the experimental data used for model identification. Continuous: asymptotes. Dashed: experimental data (Table 10.1)

2. This experiment has been performed for several values of frequency ω, and the results are presented in Table 10.1. Note that only the amplitude of the sinusoidal functions is taken into account. Moreover, the phase ϕ is not considered.
3. The Bode diagrams obtained from Table 10.1 are presented in Fig. 10.14. One 0[dB/dec] asymptote and one -20[dB/dec] asymptote fit these experimental data well. This means that the velocity motor model must have the following structure:

$$\frac{\omega(s)}{U(s)} = \frac{k}{s+a}. \tag{10.45}$$

4. From the intersection of these asymptotes, it is concluded that:

$$a = 3.3 \tag{10.46}$$

5. From the zero frequency response magnitude: $20\log(k/a) = 25.5$[dB], i.e.,

$$k = 3.3 \times 10^{\frac{25.5}{20}} = 62.1604. \tag{10.47}$$

Table 10.1 Frequency response experimental data for model identification of a PM brushed DC motor

$\bar{\omega}$[rad/s]	$2A$ [V]	$2B$ [rad/s]
0.1	1	18.68
0.2	1	18.85
0.3	1	18.85
0.4	1	18.07
0.5	1	17.68
0.6	1	18.46
0.7	1	18.46
0.8	1	18.46
0.9	1	17.67
1	1	17.67
2	1	16.1
3	1	14.14
4	1	12.17
5	1	11
6	1	9.42
7	1	8.25
8	1	7.46
9	1	6.68
10	1	6.29
20	1	3.15
30	1	2.35
40	1	1.96
50	1	1.57
60	1	1.18
70	1	1.18
80	1	0.79
90	1	0.79
100	1	0.79

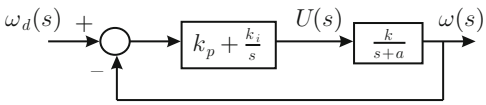

Fig. 10.15 Proportional–integral control of velocity

10.10.2 Proportional–Integral Control Design

As the motor model in 10.45 has no pole at $s = 0$, then a controller with integral action is required to render the system type 1. This ensures a zero steady-state error if the desired velocity is constant. Hence, given the first-order structure of the motor model, the design of a proportional–integral (PI) controller is well suited (see Fig. 10.15).

This means that the open loop transfer function is given as:

$$G(s)H(s) = \frac{k_p s + k_i}{s} \frac{k}{s + a}.$$

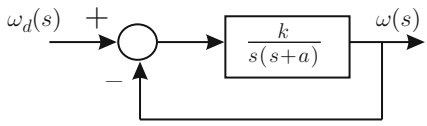

Fig. 10.16 Integral control of velocity $k_i = 1, k_p = 0$

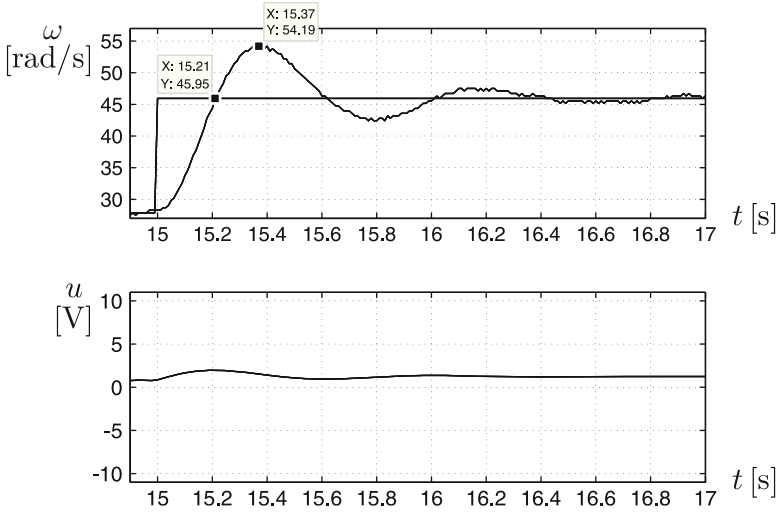

Fig. 10.17 Experimental response of the control system in Fig. 10.16, i.e., an integral controller of velocity with $k_i = 1, k_p = 0$

Proceeding as in Sect. 6.7.1, this transfer function can be rewritten as:

$$G(s)H(s) = k_p b(z+1)\frac{k}{s(s+a)}, \quad z = \frac{s}{b}, \quad b = \frac{k_i}{k_p}.$$

Hence, the factor $\frac{k}{s(s+a)}$ is to be considered as the "plant to be controlled" and the factor $k_p b(z+1)$ as the "controller." Thus, it is of interest to know how a closed system such as that shown in Fig. 10.16 would respond. Note that this can be achieved in practice by implementing an integral controller with a unit gain, i.e., $k_i = 1, k_p = 0$. In Fig. 10.17, we present the experimental results obtained when testing the response of such a control system. We observe that 45.6% overshoot and a rise time of $t_r = 0.21$[s] are produced.

It is chosen to maintain the same rise time and to reduce overshoot to 15%. Hence, the crossover frequency of the factor $\frac{k}{s(s+a)}$ alone must be kept, i.e., the magnitude of the factor $k_p b(z+1)$ must be 0[dB] at the crossover frequency of the factor $\frac{k}{s(s+a)}$. On the other hand, to achieve a 15% overshoot, we can use this value and:

$$\zeta = \sqrt{\frac{\ln^2(M_p/100)}{\ln^2(M_p/100) + \pi^2}},$$

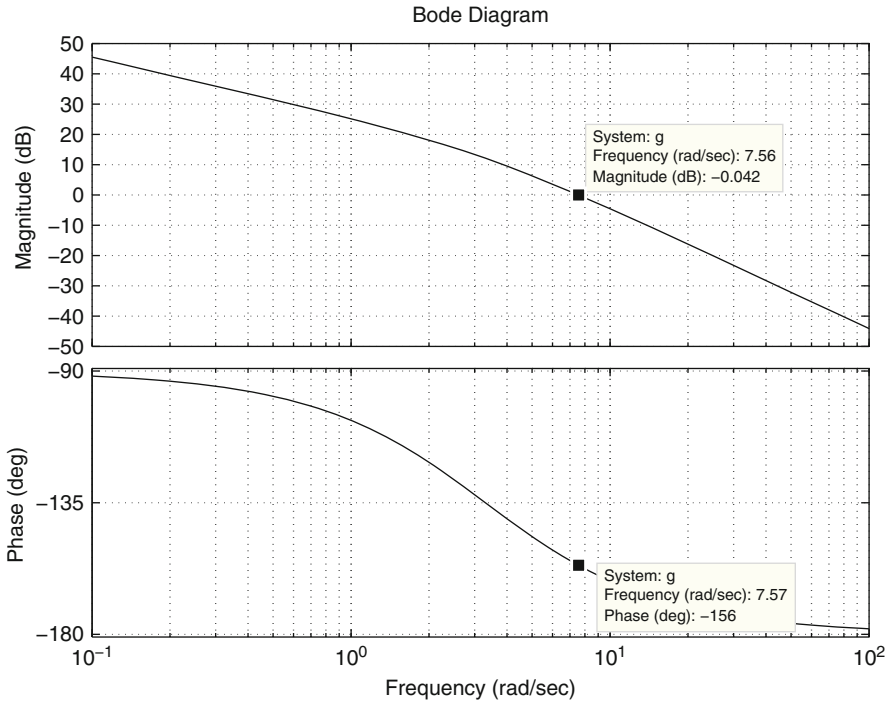

Fig. 10.18 Bode diagrams of the factor $\frac{k}{s(s+a)}$, where (10.46) and (10.47) are used

to find $\zeta = 0.5169$. According to Table 6.5, this requires a 52° phase margin. In Fig. 10.18 the Bode diagrams of the factor $\frac{k}{s(s+a)}$ are presented where (10.46) and (10.47) are used. It is observed that the crossover frequency is $\omega_1 = 7.55$[rad/s] (to be kept) and the phase is $-156°$. This means that a 28° phase lead has to be contributed by the controller, i.e., by the factor $z + 1$, when $\omega = \omega_1 = 7.55$[rad/s]. In Fig. 10.19, it is found that a 28° phase lead is obtained when $\frac{\omega}{b} = 0.53$. Hence, using $\omega = \omega_1 = 7.55$[rad/s] $b = 14.2453$ is obtained. On the other hand, from the magnitude condition:

$$|k_p b(z+1)|_{\text{dB}} = 20\log(k_p b) + |z+1|_{\text{dB}} = 0[\text{dB}],$$

where $|z + 1|_{\text{dB}} = 1.09$[dB] according to Fig. 10.19. Thus:

$$k_p = \frac{1}{b} 10^{\frac{-1.09}{20}} = 0.0619.$$

Finally, from $k_i = bk_p$, the following is found:

$$k_i = 0.8821.$$

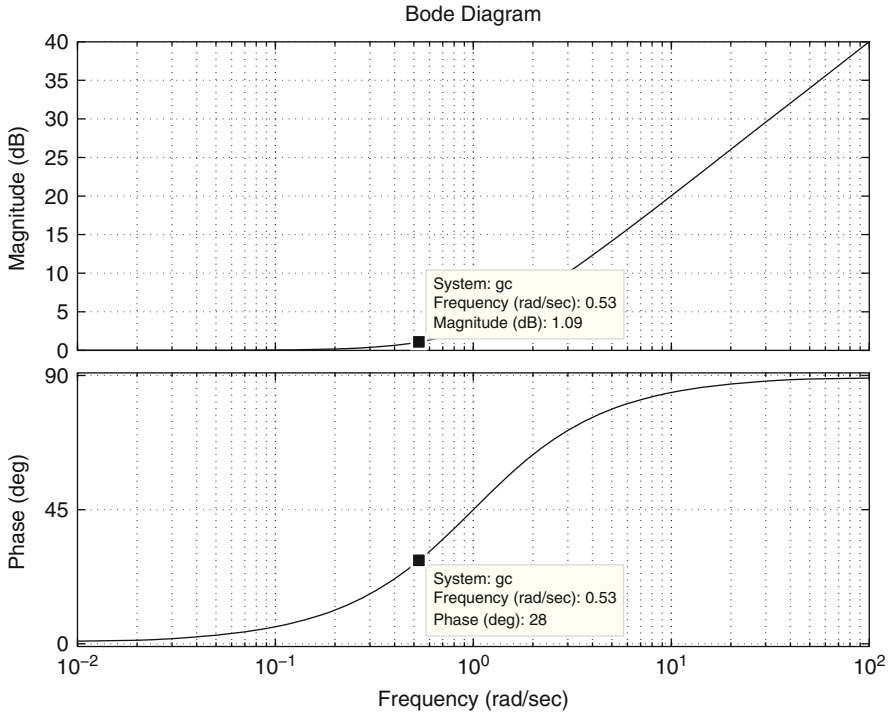

Fig. 10.19 Bode diagrams of the factor $z + 1$. The horizontal axis represents $\frac{\omega}{b}$

In Fig. 10.20, a 7.55[rad/s] crossover frequency and a 52° phase margin is observed. This means that the frequency response design is correct. On the other hand, the simulations presented in Fig. 10.21 show a 21% overshoot, instead of the desired 15%, and 0.235[s] as rise time, instead of the desired value $t_r = 0.21$[s]. In Fig. 10.22, some experimental results are presented where 15.22% is obtained as overshoot and 0.21[s] is the value of the rise time. Note that these values are very close to the desired values. Finally, according to Sect. 6.5.2, an additional open-loop phase lag of $-\omega T$[rad] must be considered to take into account the time required for controller computation. For experiments in Fig. 10.22 a sampling period of $T = 0.01$[s] was employed, which represents the worst case time delay. Thus, the additional phase lag at the crossover frequency $\omega = 7.57$[rad/s] is $-\omega T \times 180°/\pi = -4.33°$. As this phase lag is small, this explains why the response in Fig. 10.22 is not deteriorated by the time delay induced by the controller digital implementation.

In Fig. 10.23 this PI controller is compared with the PI control scheme presented (10.33). The controller gains were computed using:

$$k_1 = 0.1, \quad k_i' = 1/0.1,$$

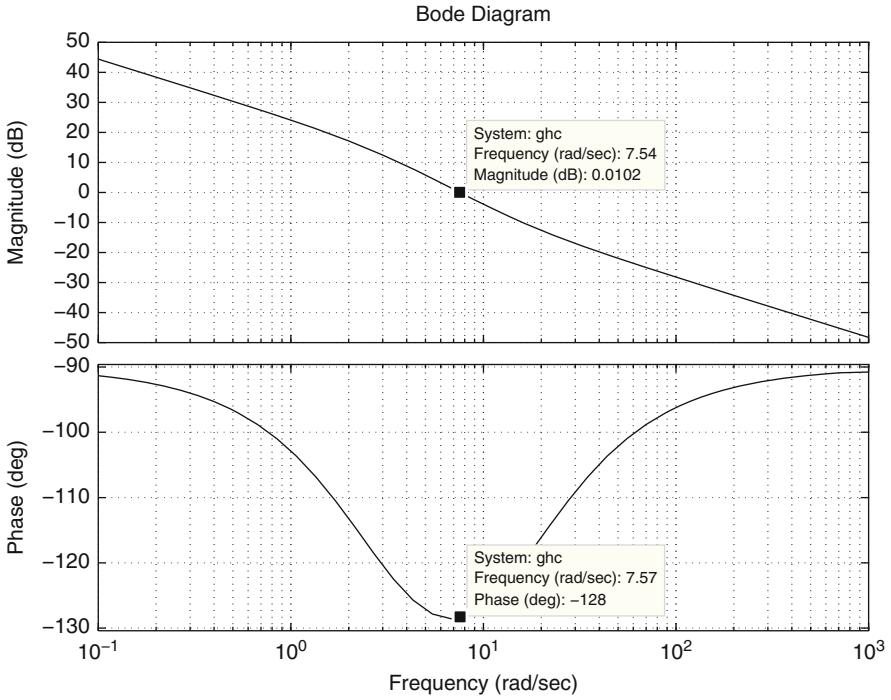

Fig. 10.20 Bode diagrams of the transfer function $\frac{k_p s+k_i}{s}\frac{k}{s+a}$. $k_p = 0.0619$, $k_i = 0.8821$, and (10.46), (10.47) are used

and the relations in (10.34). These gains were selected such that the time responses are approximately the same. Note that the controller in (10.33) forces a first-order response when step changes in the desired velocity are commanded. A constant disturbance is applied for $t \geq 17$[s] via software. This is done by adding the constant 2.5[V] to the voltage that is delivered by the controller, which has to be applied to the motor. We realize that the effects of the disturbance are a little smaller for the controller in (10.33), although the rejection time is almost the same.

Note that the disturbance rejection with both controllers is not so different than what happens with the responses obtained in Sect. 10.8 for a classical PI controller and controller in (10.33). The key to understanding this is to recall that the PI controller in Sect. 10.8 is designed by imposing $\frac{k_i}{k_p} = a$, which results in a slow rejection of external disturbances, as explained in Sect. 5.2.4. On the other hand, the classical PI controller in the present section is designed without imposing such a constraint. Thus, a faster disturbance rejection is accomplished. Note, however, that this design has been performed without taking into consideration the desired specifications for the disturbance rejection response.

Figures 10.14, and 10.18 to 10.21, in addition to all computations involved in controller design, were performed using the following MATLAB code in an m-file:

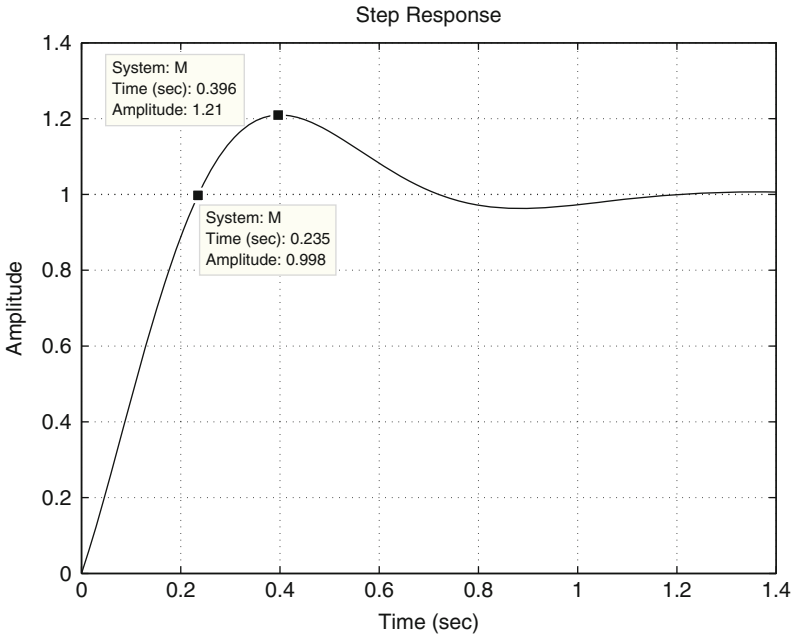

Fig. 10.21 Simulation response of the control system in Fig. 10.15 when a unit step in the desired velocity is applied. $k_p = 0.0619$, $k_i = 0.8821$, (10.46), and (10.47) are used

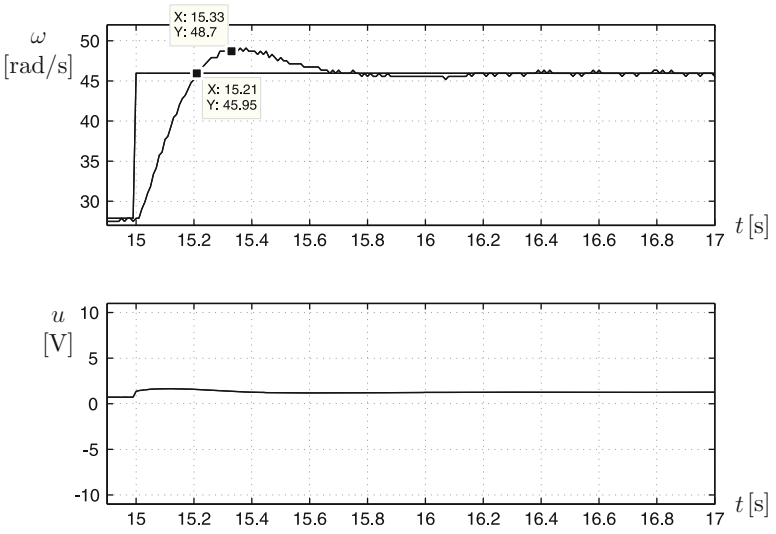

Fig. 10.22 Experimental response when using the control scheme in Fig. 10.15, with gains $k_p = 0.0619$ and $k_i = 0.8821$

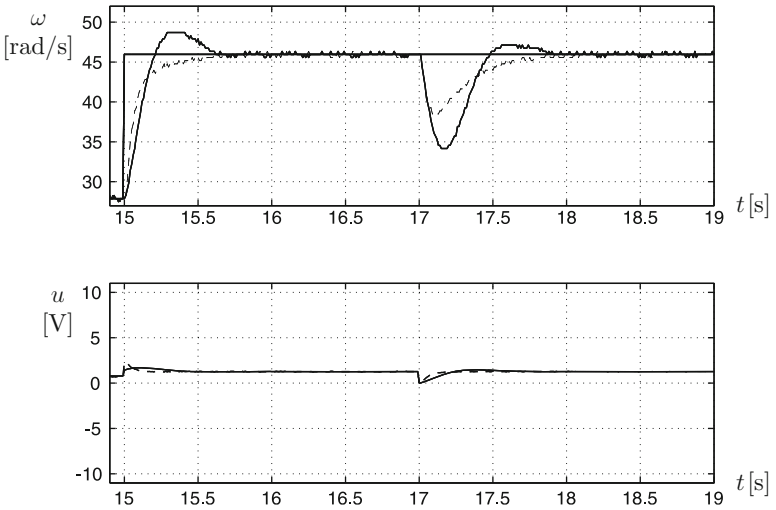

Fig. 10.23 Experimental comparison of the control scheme in Fig. 10.15 (continuous) and the controller in (10.33) (dashed)

```
clc
W=[0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1 2 3 4 5 6 7 8 9 10
20 30 40 50 60 70 80 90 100];
B=[18.68 18.85 18.85 18.07 17.68 18.46 18.46 18.46 17.67
17.67 16.1 14.14 12.17 11 9.42 8.25 7.46 6.68 6.29 3.15
2.35 1.96 1.57 1.18 1.18 0.79 0.79 0.79];
A=[1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1];
magnitude=20*log10(B./A);
figure(1)
semilogx(W,magnitude,'r-')
grid on
line([0.1 100],[25.5 25.5])
line([3.3 330],[25.5 -14.5])
a=3.3;
k=a*(10^(25.5/20))
trd=0.21; % tr desired
g=tf(k,[1 a 0]); % estimate DC motor model + integral ctrl
figure(2)
bode(g)
grid on
w1=7.55;
phase=-156;
Mpd=15;
zd=sqrt(log(Mpd/100)^2/(log(Mpd/100)^2+pi^2))
Kfd=52;
phased=-(180-52);
phase_lead=(156+phased) % 28 degrees
gc=tf([1 1],1)
figure(3)
```

```
bode(gc)
grid on
b=w1/0.53;
kp=10^(-1.09/20)/b
ki=kp*b
ctrl=tf([kp ki],[1 0]);
motor=tf(k,[1 a]);
ghc=ctrl*motor;
figure(4)
bode(ghc)
grid on
M=feedback(ghc,1,-1)
figure(5)
step(M)
grid on
```

10.10.3 Prototype Construction

The electric diagram, in addition to the computer program used to evaluate control algorithms, is identical to the electric diagram and the Builder 6 C++ code listed in Chap. 12. The only difference is that in Chap. 12, two encoders have to be read, whereas in the present chapter only one encoder is to be read. The sampling period used in the present section was 0.01[s]. Also, a microcontroller PI16F877A is used in Chap. 12 and in the present chapter. The code for programming this microcontroller is identical in both chapters. Again, in the present chapter, only one encoder has to be read. Thus, the code of the program is simpler in the present case.

In particular, the frequency response experiments performed for model identification in Sect. 10.10.1 require the following voltage $u = 2 + A\sin(\bar{\omega}t)$[V] to be applied at the armature terminals of the motor. This is accomplished by writing the program line "iast=2.0+A*sin(w t);," provided that A=0.5 and w are defined as constants in the non-executable part of the program. This is all we need, because, according to Chap. 12, the control program and hardware are so contrived that the numerical value of the variable "iast" appears as a voltage signal at the motor armature terminals.

This prototype has been built because it presents some advantages when performing the frequency response experiments described in Sect. 10.10.1. When using the prototype depicted in Fig. 10.6 for frequency response identification, the velocity of the motor has a large drift, which poses difficulties for the measurement of the velocity amplitude. Such a problem is eliminated with the new prototype.

10.11 Summary

Two velocity controllers for a PM brushed DC motor have been experimentally tested: a classical PI controller and a modified PI controller. The latter controller has been motivated by some of the drawbacks of the former and the experimental results have corroborated the good performance of the new controller. Clear explanations have also been provided on how to build the interfaces required by the complete control system and how a microcontroller can be programmed to implement the designed controller.

It is important to stress that it is common in the motor control literature to assume that either torque or electric current are the motor input signal, despite it the voltage being the actual motor input signal and torque being a variable that results from the evolution of the motor's electrical dynamics. It has been explained in this chapter that use of an internal electric current loop makes it possible to use electric current, or the generated torque, as the input variable.

A significant problem that must be solved before designing a controller is to determine the numerical values of the plant parameters. Two ways of solving this problem in a PM brushed DC motor are presented in this chapter. The fundamental idea in the first approach is to apply a step input to the motor and to observe its response. This response is found to be similar to that of a first-order system. As this response is well known, then the parameters of a first-order system are identified from such a time response. In the second approach, a frequency response experiment is performed: (1) Several sinusoidal voltages are applied at the motor terminals, each one with a different frequency. (2) The ratio between the output and the input amplitudes are plotted with respect to frequency. (3) From this magnitude Bode diagram, it is possible to estimate the numerical parameters of the DC motor model.

10.12 Review Questions

1. What is an electric current loop and what is it used for?
2. Why does a PI controller suffice to regulate velocity in a PM brushed DC motor? Why not use a proportional–integral–derivative (PID) controller?
3. Is a PI velocity controller useful for taking the steady-state error to zero when the desired velocity is not constant? What happens if the disturbance is not constant? Explain.
4. How would you use the ideas in the present chapter to regulate voltage produced by an electric generator that is actuated by a PM brushed DC motor?
5. How would you use the ideas in the present chapter to control a temperature control system and a level control system?
6. What controllers would you use for the control systems in the previous question? Would you use a PID controller? Why?

7. If a classical PI controller slowly compensates for the effect of a disturbance, what changes would you make to the controller gains?
8. What are the effects of the proportional and the integral gains in a classical PI controller?

References

1. R. Ortega, A. Loría, P. J. Nicklasson, and H. Sira-Ramírez, *Passivity-based control of Euler-Lagrange Systems*, Springer, London, 1998.
2. V. M Hernández Guzmán and R. V. Carrillo Serrano, Global PID position control of PM stepper motors and PM synchronous motors, *International Journal of Control*, vol. 84, no. 11, pp. 1807–1816, 2011.
3. V. M Hernández Guzmán and R. Silva-Ortigoza, PI control plus electric current loops for PM synchronous motors, *IEEE Transactions on Control Systems Technology*, vol. 19, no. 4, pp. 868–873, 2011.
4. G. Espinosa-Pérez, P. Maya-Ortiz, M. Velasco-Villa, and H. Sira-Ramírez, Passivity-based control of switched reluctance motors with nonlinear magnetic circuits, *IEEE Transactions on Control Systems Technology*, Vol. 12, pp. 439–448, 2004.
5. V. M. Hernández-Guzmán, J. Orrante-Sakanassi, and F. Mendoza-Mondragón, Velocity regulation in switched reluctance motors under magnetic flux saturation conditions, *Mathematical Problems in Engineering*, Hindawi, Vol. 2018, 13 pp., Article ID 3280468.
6. J. Chiasson, *Modeling and high-performance control of electric machines*, IEEE Press-Wiley Interscience, New Jersey, 2005.
7. Parker Automation, Position systems and Controls, Training and Product Catalog DC-ROM, *Compumotor's Virtual Classroom*, 1998.
8. R. Campa, E. Torres, V. Santibáñez, and R. Vargas, Electromechanical dynamics characterization of a brushless direct-drive servomotor. *Proc. VII Mexican Congress on Robotics, COMRob 2005*, México, D.F., October 27–28, 2005.
9. PIC16F877A Enhanced Flash Microcontroller, Data sheet, Microchip Technology Inc., 2003.
10. DAC0800 8-bit Digital-to-Analog Converter, Data sheet, National Semiconductor Corporation, 1995.
11. Custom Computer Services Incorporated, CCS C Compiler Reference Manual, 2003.