CHAPTER 11

BASIC CONTROL SYSTEMS DESIGN

William J. Palm III
Department of Mechanical Engineering
University of Rhode Island
Kingston, Rhode Island

1 INTRODUCTION **383**

2 CONTROL SYSTEM STRUCTURE **386**
2.1 A Standard Diagram 386
2.2 Transfer Functions 388
2.3 System-Type Number and Error Coefficients 388

3 TRANSDUCERS AND ERROR DETECTORS **389**
3.1 Displacement and Velocity Transducers 389
3.2 Temperature Transducers 391
3.3 Flow Transducers 391
3.4 Error Detectors 392
3.5 Dynamic Response of Sensors 392

4 ACTUATORS **392**
4.1 Electromechanical Actuators 392
4.2 Hydraulic Actuators 394
4.3 Pneumatic Actuators 396

5 CONTROL LAWS **399**
5.1 Proportional Control 399
5.2 Integral Control 401
5.3 Proportional-plus-Integral Control 403
5.4 Derivative Control 404
5.5 PID Control 405

6 CONTROLLER HARDWARE **405**
6.1 Feedback Compensation and Controller Design 405
6.2 Electronic Controllers 406
6.3 Pneumatic Controllers 407
6.4 Hydraulic Controllers 407

7 FURTHER CRITERIA FOR GAIN SELECTION **409**
7.1 Performance Indices 410
7.2 Optimal-Control Methods 411
7.3 The Ziegler–Nichols Rules 412
7.4 Nonlinearities and Controller Performance 413
7.5 Reset Windup 414

8 COMPENSATION AND ALTERNATIVE CONTROL STRUCTURES **414**
8.1 Series Compensation 415
8.2 Feedback Compensation and Cascade Control 415
8.3 Feedforward Compensation 415
8.4 State-Variable Feedback 417
8.5 Pseudoderivative Feedback 418

9 GRAPHICAL DESIGN METHODS **418**
9.1 The Nyquist Stability Theorem 419
9.2 Systems with Dead-Time Elements 420
9.3 Open-Loop Design for PID Control 420
9.4 Design with the Root Locus 421

10 PRINCIPLES OF DIGITAL CONTROL **424**
10.1 Digital Controller Structure 425
10.2 Digital Forms of PID Control 425

11 UNIQUELY DIGITAL ALGORITHMS **427**
11.1 Digital Feedforward Compensation 427
11.2 Control Design in the z-Plane 428
11.3 Direct Design of Digital Algorithms 432

12 HARDWARE AND SOFTWARE FOR DIGITAL CONTROL **433**

Revised from William J. Palm III, *Modeling, Analysis and Control of Dynamic Systems,* 2nd ed., Wiley, 2000, by permission of the publisher.

Copyrighted Material

Copyright © 2006 John Wiley & Sons Retrieved from: www.knovel.com

12.1 Digital Control Hardware 434
12.2 Software for Digital Control 436
12.3 Embedded Control Systems and Hardware-in-the-Loop Testing 438

13 SOFTWARE SUPPORT FOR CONTROL SYSTEM DESIGN 438
13.1 Software for Graphical Design Methods 438
13.2 Software for Control Systems Simulation 438

14 FUTURE TRENDS IN CONTROL SYSTEMS 439
14.1 Fuzzy Logic Control 441
14.2 Nonlinear Control 441
14.3 Adaptive Control 441
14.4 Optimal Control 442

REFERENCES 442

1 INTRODUCTION

The purpose of a *control system* is to produce a desired *output.* This output is usually specified by the command *input* and is often a function of time. For simple applications in well-structured situations, *sequencing* devices like timers can be used as the control system. But most systems are not that easy to control, and the controller must have the capability of reacting to disturbances, changes in its environment, and new input commands. The key element that allows a control system to do this is *feedback,* which is the process by which a system's output is used to influence its behavior. Feedback in the form of the room-temperature measurement is used to control the furnace in a thermostatically controlled heating system. Figure 1 shows the *feedback loop* in the system's *block diagram,* which is a graphical representation of the system's control structure and logic. Another commonly found control system is the pressure regulator shown in Fig. 2.

Feedback has several useful properties. A system whose individual elements are nonlinear can often be modeled as a linear one over a wider range of its variables with the proper use of feedback. This is because feedback tends to keep the system near its reference operation condition. Systems that can maintain the output near its desired value despite changes in the environment are said to have good *disturbance rejection.* Often we do not have accurate values for some system parameter or these values might change with age. Feedback can be used to minimize the effects of parameter changes and uncertainties. A system that has both good disturbance rejection and low sensitivity to parameter variation is *robust.* The application that resulted in the general understanding of the properties of feedback is shown in Fig. 3. The electronic amplifier gain A is large, but we are uncertain of its exact value. We use the resistors R_1 and R_2 to create a feedback loop around the amplifier and pick R_1 and R_2 to create a feedback loop around the amplifier and R_1 and R_2 so that $AR_2/R_1 >> 1$. Then the input–output relation becomes $e_o \approx R_1 e_i/R_2$, which is independent

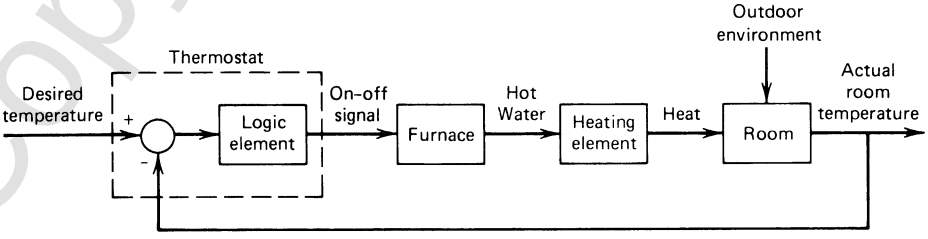


Figure 1 Block diagram of the thermostat system for temperature control.[1]

Copyright © 2006 John Wiley & Sons Retrieved from: www.knovel.com

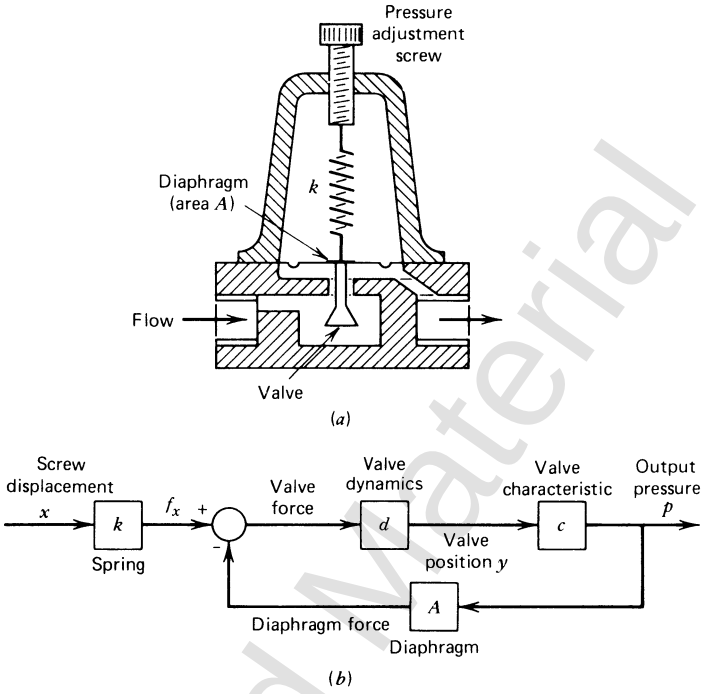


Figure 2 Pressure regulator: (*a*) cutaway view; (*b*) block diagram.[1]

of A as long as A remains large. If R_1 and R_2 are known accurately, then the system gain is now reliable.

Figure 4 shows the block diagram of a *closed-loop* system, which is a system with feedback. An *open-loop* system, such as a timer, has no feedback. Figure 4 serves as a focus for outlining the prerequisites for this chapter. The reader should be familiar with the *transfer function* concept based on the Laplace transform, the *pulse transfer* function based on the z-transform, for digital control, and the differential equation modeling techniques needed to obtain them. It is also necessary to understand block diagram algebra, characteristic roots, the final-value theorem, and their use in evaluating system response for common inputs like the step function. Also required are stability analysis techniques such as the Routh criterion

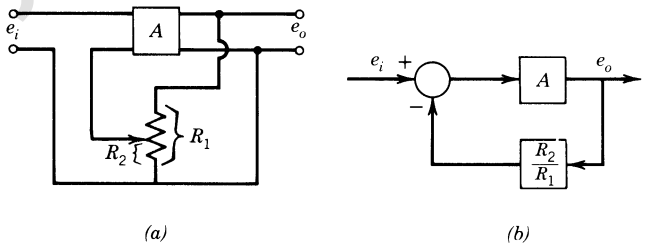


Figure 3 A closed-loop system.

Copyright © 2006 John Wiley & Sons Retrieved from: www.knovel.com

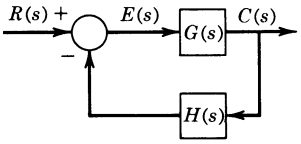


Figure 4 Feedback compensation of an amplifier.

and transient performance specifications such as the damping ratio ζ, natural frequency ω_n, dominant time constant τ, maximum overshoot, settling time, and bandwidth. The above material is reviewed in the previous chapter. Treatment in depth is given in Refs. 1–4.

2 CONTROL SYSTEM STRUCTURE

The electromechanical position control system shown in Fig. 5 illustrates the structure of a typical control system. A load with an inertia I is to be positioned at some desired angle θ_r. A dc motor is provided for this purpose. The system contains viscous damping, and a disturbance torque T_d acts on the load, in addition to the motor torque T. Because of the disturbance, the angular position θ of the load will not necessarily equal the desired value θ_r. For this reason, a potentiometer, or some other sensor such as an encoder, is used to measure the displacement θ. The potentiometer voltage representing the controlled position θ is compared to the voltage generated by the command potentiometer. This device enables the operator to dial in the desired angle θ_r. The amplifier sees the difference e between the two potentiometer voltages. The basic function of the amplifier is to increase the small error voltage e up to the voltage level required by the motor and to supply enough current required by the motor to drive the load. In addition, the amplifier may shape the voltage signal in certain ways to improve the performance of the system.

The control system is seen to provide two basic functions: (1) to respond to a command input that specifies a new desired value for the controlled variable and (2) to keep the controlled variable near the desired value in spite of disturbances. The presence of the feedback loop is vital to both functions. A block diagram of this system is shown in Fig. 6. The power supplies required for the potentiometers and the amplifier are not shown in block diagrams of control system logic because they do not contribute to the control logic.

2.1 A Standard Diagram

The electromechanical positioning system fits the general structure of a control system (Fig. 7). This figure also gives some standard terminology. Not all systems can be forced into this format, but it serves as a reference for discussion.

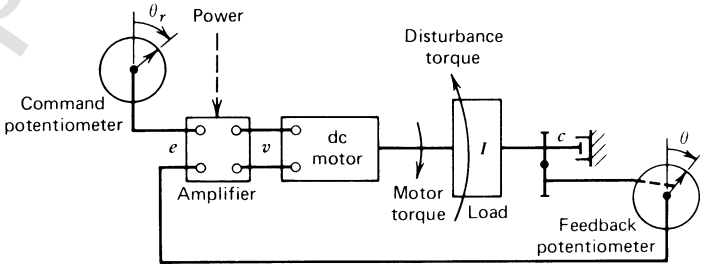


Figure 5 Position control system using a dc motor.[1]

Copyright © 2006 John Wiley & Sons Retrieved from: www.knovel.com

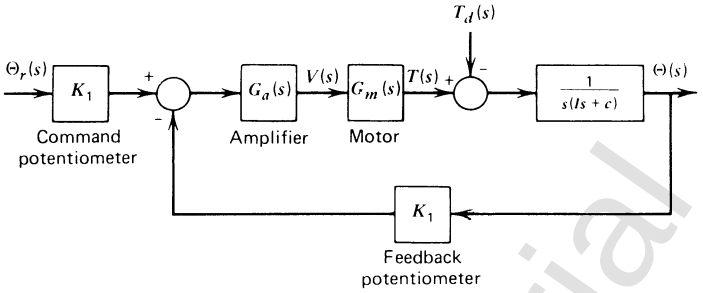
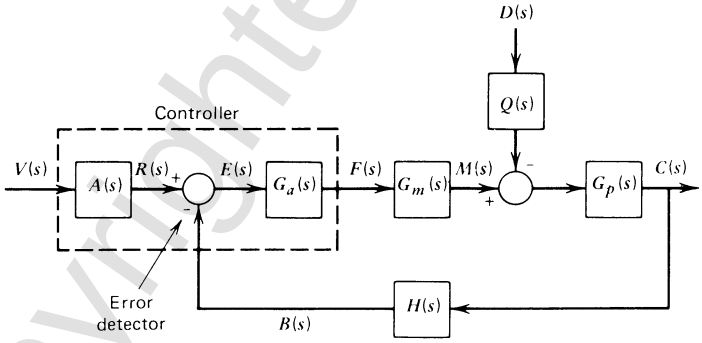


Figure 6 Block diagram of the position control system shown in Fig. 5.[1]

The controller is generally thought of as a logic element that compares the command with the measurement of the output and decides what should be done. The input and feedback elements are transducers for converting one type of signal into another type. This allows the error detector directly to compare two signals of the same type (e.g., two voltages). Not all functions show up as separate physical elements. The error detector in Fig. 5 is simply the input terminals of the amplifier.

The control logic elements produce the control signal, which is sent to the *final control elements.* These are the devices that develop enough torque, pressure, heat, and so on to influence the elements under control. Thus, the final control elements are the "muscle" of the system, while the control logic elements are the "brain." Here we are primarily concerned with the design of the logic to be used by this brain.



Elements

$A(s)$	Input elements
$G_a(s)$	Control logic elements
$G_m(s)$	Final control elements
$G_p(s)$	Plant elements
$H(s)$	Feedback elements
$Q(s)$	Disturbance elements

Signals

$B(s)$	Feedback signal
$C(s)$	Controlled variable or output
$D(s)$	Disturbance input
$E(s)$	Error or actuating signal
$F(s)$	Control signal
$M(s)$	Manipulated variable
$R(s)$	Reference input
$V(s)$	Command input

Figure 7 Terminology and basic structure of a feedback control system.[1]

Copyrighted Material

Copyright © 2006 John Wiley & Sons Retrieved from: www.knovel.com

The object to be controlled is the *plant.* The *manipulated* variable is generated by the final control elements for this purpose. The disturbance input also acts on the plant. This is an input over which the designer has no influence and perhaps for which little information is available as to the magnitude, functional form, or time of occurrence. The disturbance can be a random input, such as wind gust on a radar antenna, or deterministic, such as Coulomb friction effects. In the latter case, we can include the friction force in the system model by using a nominal value for the coefficient of friction. The disturbance input would then be the deviation of the friction force from this estimated value and would represent the uncertainty in our estimate.

Several control system classifications can be made with reference to Fig. 7. A *regulator* is a control system in which the controlled variable is to be kept constant in spite of disturbances. The command input for a regulator is its *set point.* A *follow-up system* is supposed to keep the control variable near a command value that is changing with time. An example of a follow-up system is a machine tool in which a cutting head must trace a specific path in order to shape the product properly. This is also an example of a *servomechanism,* which is a control system whose controlled variable is a mechanical position, velocity, or acceleration. A thermostat system is not a servomechanism, but a *process control system,* where the controlled variable describes a thermodynamic process. Typically, such variables are temperature, pressure, flow rate, liquid level, chemical concentration, and so on.

2.2 Transfer Functions

A transfer function is defined for each input–output pair of the system. A specific transfer function is found by setting all other inputs to zero and reducing the block diagram. The *primary* or *command* transfer function for Fig. 7 is

$$\frac{C(s)}{V(s)} = \frac{A(s)G_a(s)G_m(s)G_p(s)}{1 + G_a(s)G_m(s)G_p(s)H(s)} \tag{1}$$

The *disturbance* transfer function is

$$\frac{C(s)}{D(s)} = \frac{-Q(s)G_p(s)}{1 + G_a(s)G_m(s)G_p(s)H(s)} \tag{2}$$

The transfer functions of a given system all have the same denominator.

2.3 System-Type Number and Error Coefficients

The error signal in Fig. 4 is related to the input as

$$E(s) = \frac{1}{1 + G(s)H(s)} R(s) \tag{3}$$

If the final-value theorem can be applied, the steady-state error is

$$e_{ss} = \lim_{s \to 0} \frac{sR(s)}{1 + G(s)H(s)} \tag{4}$$

The *static error coefficient* c_i is defined as

$$c_i = \lim_{s \to 0} s^i G(s)H(s) \tag{5}$$

Copyrighted Material

Copyright © 2006 John Wiley & Sons Retrieved from: www.knovel.com

A system is of *type n* if $G(s)H(s)$ can be written as $s^n F(s)$. Table 1 relates the steady-state error to the system type for three common inputs and can be used to design systems for minimum error. The higher the system type, the better the system is able to follow a rapidly changing input. But higher type systems are more difficult to stabilize, so a compromise must be made in the design. The coefficients c_0, c_1, and c_2 are called the *position, velocity,* and *acceleration error coefficients.*

3 TRANSDUCERS AND ERROR DETECTORS

The control system structure shown in Fig. 7 indicates a need for physical devices to perform several types of functions. Here we present a brief overview of some available transducers and error detectors. Actuators and devices used to implement the control logic are discussed in Sections 4 and 5.

3.1 Displacement and Velocity Transducers

A *transducer* is a device that converts one type of signal into another type. An example is the potentiometer, which converts displacement into voltage, as in Fig. 8. In addition to this conversion, the transducer can be used to make measurements. In such applications, the term *sensor* is more appropriate. Displacement can also be measured electrically with a *linear variable differential transformer* (LVDT) or a *synchro.* An LVDT measures the linear displacement of a movable magnetic core through a primary winding and two secondary windings (Fig. 9). An ac voltage is applied to the primary. The secondaries are connected together and also to a detector that measures the voltage and phase difference. A phase difference of 0° corresponds to a positive core displacement, while 180° indicates a negative displacement. The amount of displacement is indicated by the amplitude of the ac voltage in the secondary. The detector converts this information into a dc voltage e_o, such that $e_o = Kx$. The LVDT is sensitive to small displacements. Two of them can be wired together to form an error detector.

A synchro is a rotary differential transformer, with angular displacement as either the input or output. They are often used in pairs (a *transmitter* and a *receiver*) where a remote indication of angular displacement is needed. When a transmitter is used with a synchro *control transformer,* two angular displacements can be measured and compared (Fig. 10). The output voltage e_o is approximately linear with angular difference within $\pm 70°$, so that $e_o = K(\theta_1 - \theta_2)$.

Table 1 Steady-State Error e_{ss} for Different System-Type Numbers

	System-Type Number n			
$R(s)$	0	1	2	3
Step $1/s$	$\dfrac{1}{1 + C_0}$	0	0	0
Ramp $1/s^2$	∞	$\dfrac{1}{C_1}$	0	0
Parabola $1/s^3$	∞	∞	$\dfrac{1}{C_2}$	0

Copyright © 2006 John Wiley & Sons Retrieved from: www.knovel.com

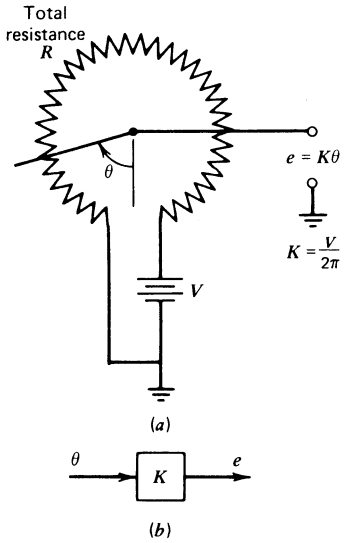


Figure 8 Rotary potentiometer.[1]

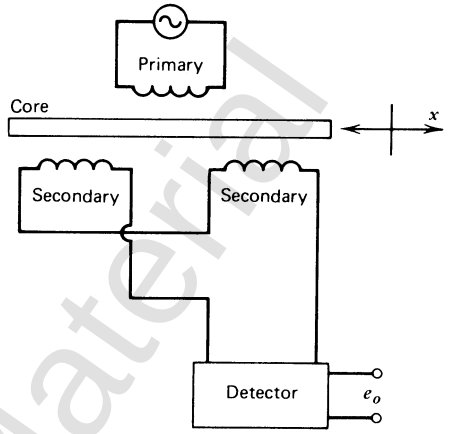


Figure 9 Linear variable differential transformer.[1]

Displacement measurements can be used to obtain forces and accelerations. For example, the displacement of a calibrated spring indicates the applied force. The accelerometer is another example. Still another is the strain gage used for force measurement. It is based on the fact that the resistance of a fine wire changes as it is stretched. The change in resistance is detected by a circuit that can be calibrated to indicate the applied force. Sensors utilizing piezoelectric elements are also available.

Velocity measurements in control systems are most commonly obtained with a *tachometer.* This is essentially a dc generator (the reverse of a dc motor). The input is mechanical (a velocity). The output is a generated voltage proportional to the velocity. Translational

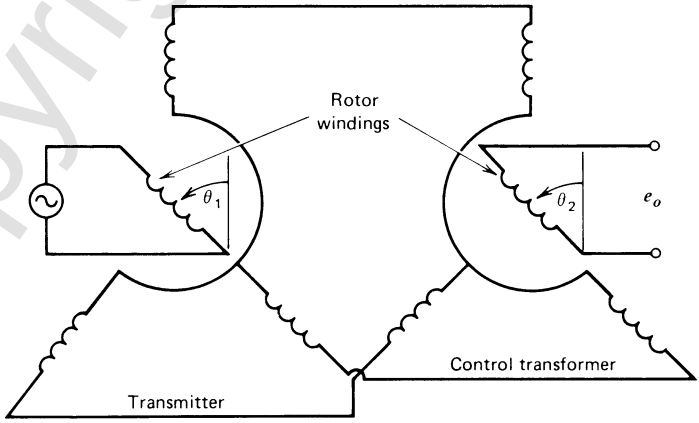


Figure 10 Synchro transmitter control transformer.[1]

Copyrighted Material

Copyright © 2006 John Wiley & Sons Retrieved from: www.knovel.com

velocity can be measured by converting it to angular velocity with gears, for example. Tachometers using ac signals are also available.

Other velocity transducers include a magnetic pickup that generates a pulse every time a gear tooth passes. If the number of gear teeth is known, a pulse counter and timer can be used to compute the angular velocity. This principle is also employed in turbine flowmeters.

A similar principle is employed by *optical encoders,* which are especially suitable for digital control purposes. These devices use a rotating disk with alternating transparent and opaque elements whose passage is sensed by light beams and a photosensor array, which generates a binary (on–off) train of pulses. There are two basic types: the absolute encoder and the incremental encoder. By counting the number of pulses in a given time interval, the incremental encoder can measure the rotational speed of the disk. By using multiple tracks of elements, the absolute encoder can produce a binary digit that indicates the amount of rotation. Hence, it can be used as a position sensor.

Most encoders generate a train of transistor–transistor logic (TTL) voltage level pulses for each channel. The incremental encoder output contains two channels that each produce N pulses every revolution. The encoder is mechanically constructed so that pulses from one channel are shifted relative to the other channel by a quarter of a pulse width. Thus, each pulse pair can be divided into four segments called *quadratures.* The encoder output consists of $4N$ *quadrature counts per revolution.* The pulse shift also allows the direction of rotation to be determined by detecting which channel leads the other. The encoder might contain a third channel, known as the zero, index, or marker channel, that produces a pulse once per revolution. This is used for initialization.

The gain of such an incremental encoder is $4N/2\pi$. Thus, an encoder with 1000 pulses per channel per revolution has a gain of 636 counts per radian. If an absolute encoder produces a binary signal with n bits, the maximum number of positions it can represent is $2n$, and its gain is $2^n/2\pi$. Thus, a 16-bit absolute encoder has a gain of $2^{16}/2\pi = 10{,}435$ counts per radian.

3.2 Temperature Transducers

When two wires of dissimilar metals are joined together, a voltage is generated if the junctions are at different temperatures. If the reference junction is kept at a fixed, known temperature, the thermocouple can be calibrated to indicate the temperature at the other junction in terms of the voltage v. Electrical resistance changes with temperature. Platinum gives a linear relation between resistance and temperature, while nickel is less expensive and gives a large resistance change for a given temperature change. Seminconductors designed with this property are called *thermistors.* Different metals expand at different rates when the temperature is increased. This fact is used in the bimetallic strip transducer found in most home thermostats. Two dissimilar metals are bonded together to form the strip. As the temperature rises, the strip curls, breaking contact and shutting off the furnace. The temperature gap can be adjusted by changing the distance between the contacts. The motion also moves a pointer on the temperature scale of the thermostat. Finally, the pressure of a fluid inside a bulb will change as its temperature changes. If the bulb fluid is air, the device is suitable for use in pneumatic temperature controllers.

3.3 Flow Transducers

A flow rate q can be measured by introducing a flow restriction, such as an orifice plate, and measuring the pressure drop Δp across the restriction. The relation is $\Delta p = Rq^2$, where

Copyright © 2006 John Wiley & Sons Retrieved from: www.knovel.com

R can be found from calibration of the device. The pressure drop can be sensed by converting it into the motion of a diaphragm. Figure 11 illustrates a related technique. The Venturi-type flowmeter measures the static pressures in the constricted and unconstricted flow regions. Bernoulli's principle relates the pressure difference to the flow rate. This pressure difference produces the diaphragm displacement. Other types of flowmeters are available, such as turbine meters.

3.4 Error Detectors

The error detector is simply a device for finding the difference between two signals. This function is sometimes an integral feature of sensors, such as with the synchro transmitter–transformer combination. This concept is used with the diaphragm element shown in Fig. 11. A detector for voltage difference can be obtained, as with the position control system shown in Fig. 5. An amplifier intended for this purpose is a *differential amplifier.* Its output is proportional to the difference between the two inputs. In order to detect differences in other types of signals, such as temperature, they are usually converted to a displacement or pressure. One of the detectors mentioned previously can then be used.

3.5 Dynamic Response of Sensors

The usual transducer and detector models are static models and as such imply that the components respond instantaneously to the variable being sensed. Of course, any real component has a dynamic response of some sort, and this response time must be considered in relation to the controlled process when a sensor is selected. If the controlled process has a time constant at least 10 times greater than that of the sensor, we often would be justified in using a static sensor model.

4 ACTUATORS

An *actuator* is the final control element that operates on the low-level control signal to produce a signal containing enough power to drive the plant for the intended purpose. The armature-controlled dc motor, the hydraulic servomotor, and the pneumatic diaphragm and piston are common examples of actuators.

4.1 Electromechanical Actuators

Figure 12 shows an electromechanical system consisting of an armature-controlled dc motor driving a load inertia. The rotating armature consists of a wire conductor wrapped around

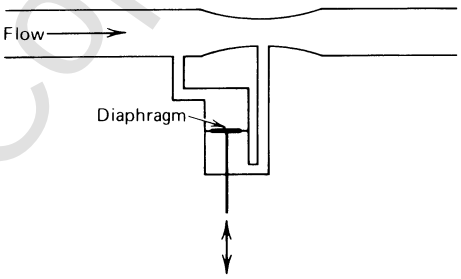


Figure 11 Venturi-type flowmeter. The diaphragm displacement indicates the flow rate.[1]

Copyrighted Material

Copyright © 2006 John Wiley & Sons Retrieved from: www.knovel.com

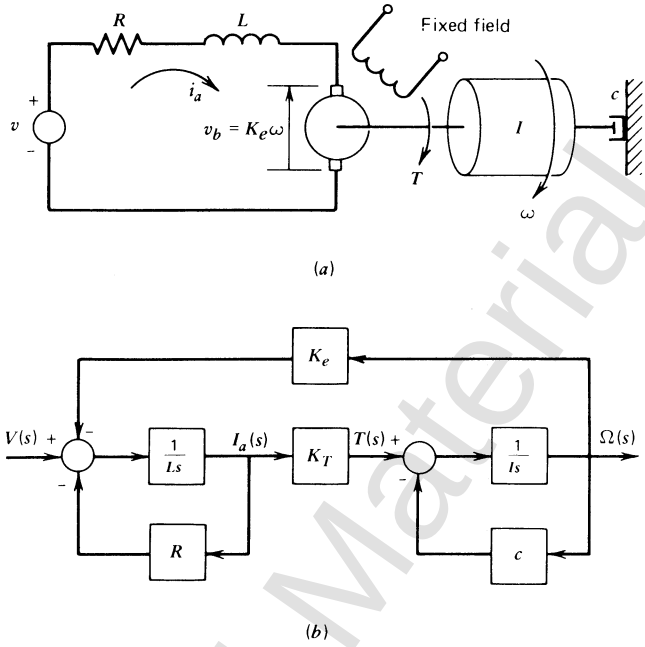


Figure 12 Armature-controlled dc motor with a load and the system's block diagram.[1]

an iron core. This winding has an inductance L. The resistance R represents the lumped value of the armature resistance and any external resistance deliberately introduced to change the motor's behavior. The armature is surrounded by a magnetic field. The reaction of this field with the armature current produces a torque that causes the armature to rotate. If the armature voltage v is used to control the motor, the motor is said to be *armature controlled.* In this case, the field is produced by an electromagnet supplied with a constant voltage or by a permanent magnet. This motor type produces a torque T that is proportional to the armature current i_a:

$$T = K_T i_a \tag{6}$$

The torque constant K_T depends on the strength of the field and other details of the motor's construction. The motion of a current-carrying conductor in a field produces a voltage in the conductor that opposes the current. This voltage is called the *back emf* (electromotive force). Its magnitude is proportional to the speed and is given by

$$e_b = K_e\omega \tag{7}$$

The transfer function for the armature-controlled dc motor is

$$\frac{\Omega(s)}{V(s)} = \frac{K_T}{LIs^2 + (RI + cL)s + cR + K_eK_T} \tag{8}$$

Another motor configuration is the *field-controlled* dc motor. In this case, the armature current is kept constant and the field voltage v is used to control the motor. The transfer function is

Copyrighted Material

Copyright © 2006 John Wiley & Sons Retrieved from: www.knovel.com

$$\frac{\Omega(s)}{V(s)} = \frac{K_T}{(Ls + R)(Is + c)} \tag{9}$$

where R and L are the resistance and inductance of the field circuit and K_T is the torque constant. No back emf exists in this motor to act as a self-braking mechanism.

Two-phase ac motors can be used to provide a low-power, variable-speed actuator. This motor type can accept the ac signals directly from LVDTs and synchros without demodulation. However, it is difficult to design ac amplifier circuitry to do other than proportional action. For this reason, the ac motor is not found in control systems as often as dc motors. The transfer function for this type is of the form of Eq. (9).

An actuator especially suitable for digital systems is the *stepper motor,* a special dc motor that takes a train of electrical input pulses and converts each pulse into an angular displacement of a fixed amount. Motors are available with resolutions ranging from about 4 steps per revolution to more than 800 steps per revolution. For 36 steps per revolution, the motor will rotate by 10° for each pulse received. When not being pulsed, the motors lock in place. Thus, they are excellent for precise positioning applications, such as required with printers and computer tape drives. A disadvantage is that they are low-torque devices. If the input pulse frequency is not near the resonant frequency of the motor, we can take the output rotation to be directly related to the number of input pulses and use that description as the motor model.

4.2 Hydraulic Actuators

Machine tools are one application of the hydraulic system shown in Fig. 13. The applied force f is supplied by the servomotor. The mass m represents that of a cutting tool and the power piston, while k represents the combined effects of the elasticity naturally present in the structure and that introduced by the designer to achieve proper performance. A similar statement applies to the damping c. The valve displacement z is generated by another control system in order to move the tool through its prescribed motion. The spool valve shown in

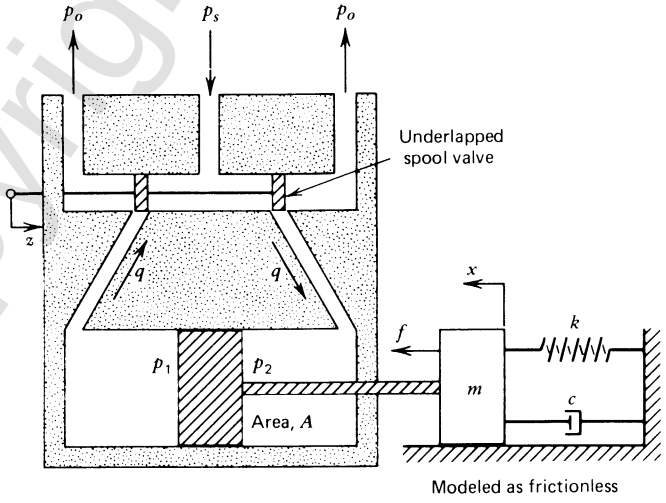


Figure 13 Hydraulic servomotor with a load.[1]

Copyrighted Material

Copyright © 2006 John Wiley & Sons Retrieved from: www.knovel.com

Fig. 13 had two *lands.* If the width of the land is greater than the port width, the valve is said to be *overlapped.* In this case, a dead zone exists in which a slight change in the displacement z produces no power piston motion. Such dead zones create control difficulties and are avoided by designing the valve to be *underlapped* (the land width is less the port width). For such valves there will be a small flow opening even when the valve is in the neutral position at $z = 0$. This gives it a higher sensitivity than an overlapped valve.

The variables z and $\Delta p = p_2 - p_1$ determine the volume flow rate, as

$$q = f(z,\Delta p)$$

For the reference equilibrium condition ($z = 0$, $\Delta p = 0$, $q = 0$), a linearization gives

$$q = C_1 z - C_2 \Delta p \tag{10}$$

The linearization constants are available from theoretical and experimental results.[5] The transfer function for the system is[1,2]

$$T(s) = \frac{X(s)}{Z(s)} = \frac{C_1}{(C_2 m/A)\, s^2 + (cC_2/A + A)\, s + C_2 k/A} \tag{11}$$

The development of the steam engine led to the requirement for a speed control device to maintain constant speed in the presence of changes in load torque or steam pressure. In 1788, James Watt of Glasgow developed his now-famous flyball governor for this purpose (Fig. 14). Watt took the principle of sensing speed with the centrifugal pendulum of Thomas Mead and used it in a feedback loop on a steam engine. As the motor speed increases, the flyballs move outward and pull the slider upward. The upward motion of the slider closes the steam valve, thus causing the engine to slow down. If the engine speed is too slow, the spring force overcomes that due to the flyballs, and the slider moves down to open the steam valve. The desired speed can be set by moving the plate to change the compression in the spring. The principle of the flyball governor is still used for speed control applications. Typically, the pilot valve of a hydraulic servomotor is connected to the slider to provide the high forces required to move large supply valves.

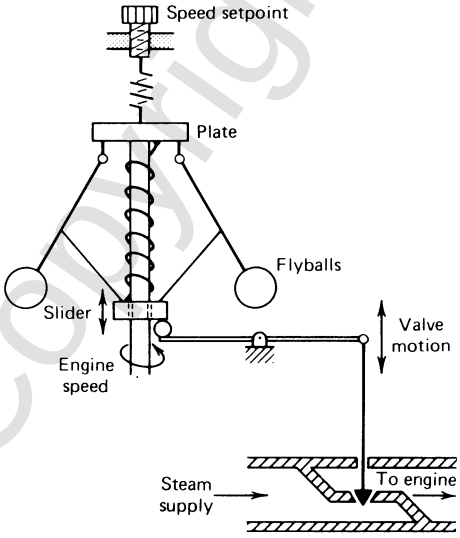


Figure 14 James Watt's flyball governor for speed control of a steam engine.[1]

Copyrighted Material

Copyright © 2006 John Wiley & Sons Retrieved from: www.knovel.com

Many hydraulic servomotors use multistage valves to obtain finer control and higher forces. A *two-stage valve* has a *slave valve,* similar to the pilot valve but situated between the pilot valve and the power piston.

Rotational motion can be obtained with a *hydraulic motor,* which is, in principle, a pump acting in reverse (fluid input and mechanical rotation output). Such motors can achieve higher torque levels than electric motors. A hydraulic pump driving a hydraulic motor constitutes a *hydraulic transmission.*

A popular actuator choice is the *electrohydraulic* system, which uses an electric actuator to control a hydraulic servomotor or transmission by moving the pilot valve or the swashplate angle of the pump. Such systems combine the power of hydraulics with the advantages of electrical systems. Figure 15 shows a hydraulic motor whose pilot valve motion is caused by an armature-controlled dc motor. The transfer function between the motor voltage and the piston displacement is

$$\frac{X(s)}{V(s)} = \frac{K_1 K_2 C_1}{As^2(\tau s + 1)} \tag{12}$$

If the rotational inertia of the electric motor is small, then $\tau \approx 0$.

4.3 Pneumatic Actuators

Pneumatic actuators are commonly used because they are simple to maintain and use a readily available working medium. Compressed air supplies with the pressures required are commonly available in factories and laboratories. No flammable fluids or electrical sparks are present, so these devices are considered the safest to use with chemical processes. Their power output is less than that of hydraulic systems but greater than that of electric motors.

A device for converting pneumatic pressure into displacement is the bellows shown in Fig. 16. The transfer function for a linearized model of the bellows is of the form

$$\frac{X(s)}{P(s)} = \frac{K}{\tau s + 1} \tag{13}$$

where x and p are deviations of the bellows displacement and input pressure from nominal values.

In many control applications, a device is needed to convert small displacements into relatively large pressure changes. The nozzle–flapper serves this purpose (Fig. 17*a*). The input displacement y moves the flapper, with little effort required. This changes the opening at the nozzle orifice. For a large enough opening, the nozzle back pressure is approximately the same as atmospheric pressure p_a. At the other extreme position with the flapper completely blocking the orifice, the back pressure equals the supply pressure p_s. This variation is shown in Fig. 17*b*. Typical supply pressures are between 30 and 100 psia. The orifice

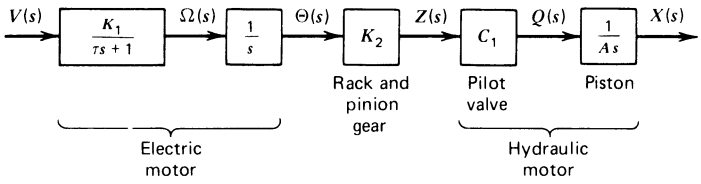


Figure 15 Electrohydraulic system for translation.[1]

Copyrighted Material

Copyright © 2006 John Wiley & Sons Retrieved from: www.knovel.com

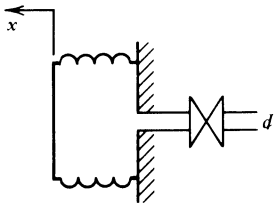


Figure 16 Pneumatic bellows.[1]

diameter is approximately 0.01 in. Flapper displacement is usually less than one orifice diameter.

The nozzle–flapper is operated in the linear portion of the back-pressure curve. The linearized back pressure relation is

$$p = -K_f x \tag{14}$$

where $-K_f$ is the slope of the curve and is a very large number. From the geometry of similar triangles, we have

$$p = -\frac{aK_f}{a+b}\,y \tag{15}$$

In its operating region, the nozzle–flapper's back pressure is well below the supply pressure.

The output pressure from a pneumatic device can be used to drive a final control element like the pneumatic actuating valve shown in Fig. 18. The pneumatic pressure acts on the upper side of the diaphragm and is opposed by the return spring.

Formerly, many control systems utilized pneumatic devices to implement the control law in analog form. Although the overall, or higher level, control algorithm is now usually implemented in digital form, pneumatic devices are still frequently used for final control corrections at the actuator level, where the control action must eventually be supplied by a mechanical device. An example of this is the electropneumatic valve positioner used in Valtek valves and illustrated in Fig. 19. The heart of the unit is a pilot valve capsule that moves up and down according to the pressure difference across its two supporting diaphragms. The capsule has a plunger at its top and at its bottom. Each plunger has an exhaust seat at one

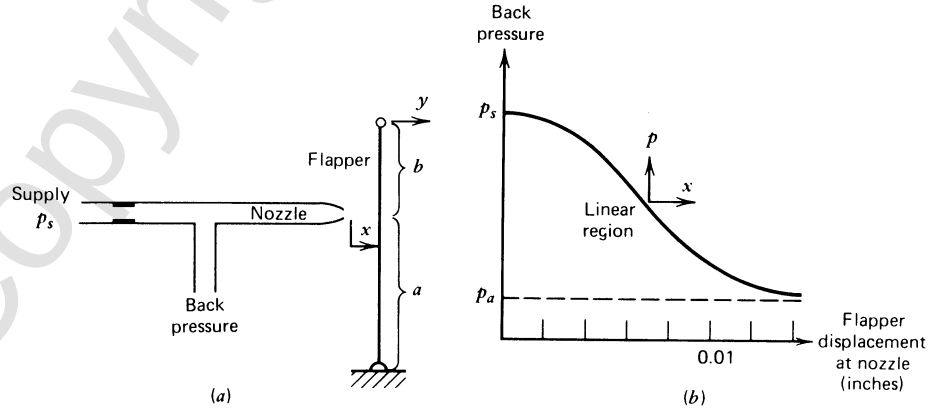


Figure 17 Pneumatic nozzle–flapper amplifier and its characteristic curve.[1]

Copyright © 2006 John Wiley & Sons Retrieved from: www.knovel.com

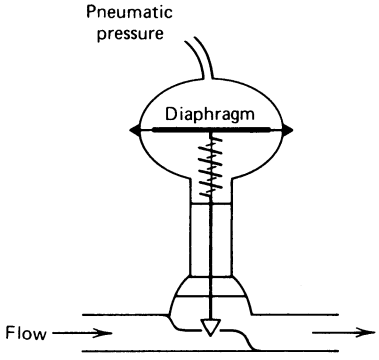


Figure 18 Pneumatic flow control valve.[1]

end and a supply seat at the other. When the capsule is in its equilibrium position, no air is supplied to or exhausted from the valve cylinder, so the valve does not move.

The process controller commands a change in the valve stem position by sending the 4–20-mA dc input signal to the positioner. Increasing this signal causes the electromagnetic actuator to rotate the lever counterclockwise about the pivot. This increases the air gap between the nozzle and flapper. This decreases the back pressure on top of the upper diaphragm and causes the capsule to move up. This motion lifts the upper plunger from its supply seat and allows the supply air to flow to the bottom of the valve cylinder. The lower plunger's exhaust seat is uncovered, thus decreasing the air pressure on top of the valve piston, and the valve stem moves upward. This motion causes the lever arm to rotate, increasing the tension in the feedback spring and decreasing the nozzle–flapper gap. The valve

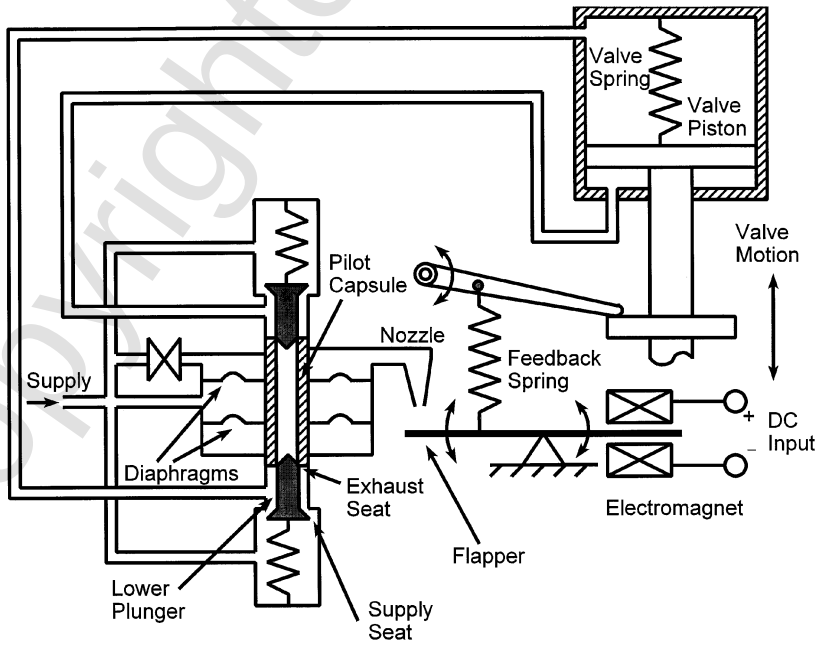


Figure 19 An electropneumatic valve positioner.

Copyrighted Material

Copyright © 2006 John Wiley & Sons Retrieved from: www.knovel.com

continues to move upward until the tension in the feedback spring counteracts the force produced by the electromagnetic actuator, thus returning the capsule to its equilibrium position.

A decrease in the dc input signal causes the opposite actions to occur, and the valve moves downward.

5 CONTROL LAWS

The control logic elements are designed to act on the error signal to produce the control signal. The algorithm that is used for this purpose is called the *control law,* the *control action,* or the *control algorithm.* A nonzero error signal results from either a change in command or a disturbance. The general function of the controller is to keep the controlled variable near its desired value when these occur. More specifically, the control objectives might be stated as follows:

1. Minimize the steady-state error.
2. Minimize the settling time.
3. Achieve other transient specifications, such as minimizing the overshoot.

In practice, the design specifications for a controller are more detailed. For example, the bandwidth might also be specified along with a safety margin for stability. We never know the numerical values of the system's parameters with true certainty, and some controller designs can be more sensitive to such parameter uncertainties than other designs. So a parameter sensitivity specification might also be included.

The following control laws form the basis of most control systems.

5.1 Proportional Control

Two-position control is the most familiar type, perhaps because of its use in home thermostats. The control output takes on one of two values. With the *on–off controller,* the controller output is either on or off (e.g., fully open or fully closed). Two-position control is acceptable for many applications in which the requirements are not too severe. However, many situations require finer control.

Consider a liquid-level system in which the input flow rate is controlled by a valve. We might try setting the control valve manually to achieve a flow rate that balances the system at the desired level. We might then add a controller that adjusts this setting in proportion to the deviation of the level from the desired value. This is *proportional control,* the algorithm in which the change in the control signal is proportional to the error. Block diagrams for controllers are often drawn in terms of the deviations from a zero-error equilibrium condition. Applying this convention to the general terminology of Fig. 6, we see that proportional control is described by

$$F(s) = K_P E(s)$$

where $F(s)$ is the deviation in the control signal and K_P is the *proportional gain.* If the total valve displacement is $y(t)$ and the manually created displacement is x, then

$$y(t) = K_P e(t) + x$$

The percent change in error needed to move the valve full scale is the *proportional band.* It is related to the gain as

Copyright © 2006 John Wiley & Sons Retrieved from: www.knovel.com

$$K_P = \frac{100}{\text{band } \%}$$

The zero-error valve displacement x is the *manual reset.*

Proportional Control of a First-Order System

To investigate the behavior of proportional control, consider the speed control system shown in Fig. 20; it is identical to the position controller shown in Fig. 6, except that a tachometer replaces the feedback potentiometer. We can combine the amplifier gains into one, denoted K_P. The system is thus seen to have proportional control. We assume the motor is field controlled and has a negligible electrical time constant. The disturbance is a torque T_d, for example, resulting from friction. Choose the reference equilibrium condition to be $T_d = T = 0$ and $\omega_r = w = 0$. The block diagram is shown in Fig. 21. For a meaningful error signal to be generated, K_1 and K_2 should be chosen to be equal. With this simplification the diagram becomes that shown in Fig. 22, where $G(s) = K = K_1 K_P K_T/R$. A change in desired speed can be simulated by a unit step input for ω_r. For $\Omega_r(s) = 1/s$, the velocity approaches the steady-state value $\omega_{ss} = K/(c + K) < 1$. Thus, the final value is less than the desired value of 1, but it might be close enough if the damping c is small. The time required to reach this value is approximately four time constants, or $4\tau = 4I/(c + K)$. A sudden change in load torque can also be modeled by a unit step function $T_d(s) = 1/s$. The steady-state response due solely to the disturbance is $-1/(c + K)$. If $c + K$ is large, this error will be small.

The performance of the proportional control law thus far can be summarized as follows. For a first-order plant with step function inputs:

1. The output never reaches its desired value if damping is present ($c \neq 0$), although it can be made arbitrarily close by choosing the gain K large enough. This is called *offset error.*
2. The output approaches its final value without oscillation. The time to reach this value is inversely proportional to K.
3. The output deviation due to the disturbance at steady state is inversely proportional to the gain K. This error is present even in the absence of damping ($c = 0$).

As the gain K is increased, the time constant becomes smaller and the response faster. Thus, the chief disadvantage of proportional control is that it results in steady-state errors and can only be used when the gain can be selected large enough to reduce the effect of the largest expected disturbance. Since proportional control gives zero error only for one load condition (the reference equilibrium), the operator must change the manual reset by hand

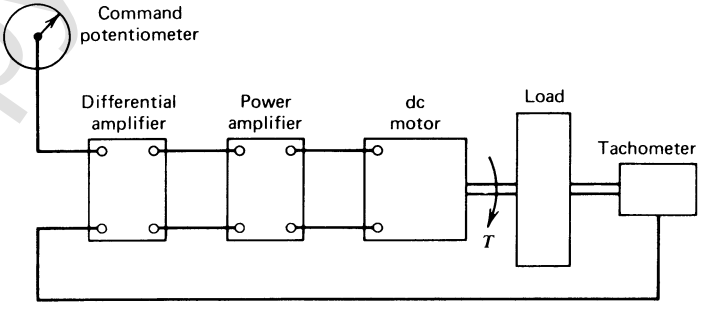


Figure 20 Velocity control system using a dc motor.[1]

Copyright © 2006 John Wiley & Sons Retrieved from: www.knovel.com

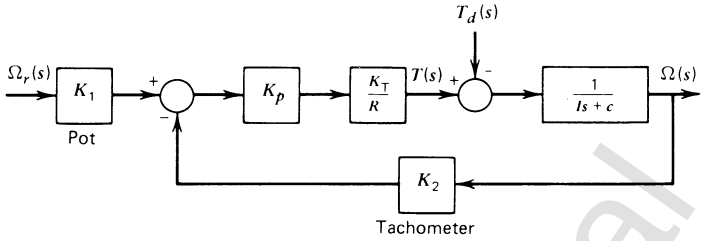


Figure 21 Block diagram of the velocity control system of Fig. 20.[1]

(hence the name). An advantage to proportional control is that the control signal responds to the error instantaneously (in theory at least). It is used in applications requiring rapid action. Processes with time constants too small for the use of two-position control are likely candidates for proportional control. The results of this analysis can be applied to any type of first-order system (e.g., liquid level, thermal, etc.) having the form in Fig. 22.

Proportional Control of a Second-Order System

Proportional control of a neutrally stable second-order plant is represented by the position controller of Fig. 6 if the amplifier transfer function is a constant $G_a(s) = K_a$. Let the motor transfer function be $G_m(s) = K_T/R$, as before. The modified block diagram is given in Fig. 23 with $G(s) = K = K_1K_aK_T/R$. The closed-loop system is stable if I, c, and K are positive. For no damping ($c = 0$), the closed-loop system is neutrally stable. With no disturbance and a unit step command, $\Theta_r(s) = 1/s$, the steady-state output is $\omega_{ss} = 1$. The offset error is thus zero if the system is stable ($c > 0$, $K > 0$). The steady-state output deviation due to a unit step disturbance is $-1/K$. This deviation can be reduced by choosing K large. The transient behavior is indicated by the damping ratio, $\zeta = c/2\sqrt{IK}$.

For slight damping, the response to a step input will be very oscillatory and the overshoot large. The situation is aggravated if the gain K is made large to reduce the deviation due to the disturbance. We conclude, therefore, that proportional control of this type of second-order plant is not a good choice unless the damping constant c is large. We will see shortly how to improve the design.

5.2 Integral Control

The offset error that occurs with proportional control is a result of the system reaching an equilibrium in which the control signal no longer changes. This allows a constant error to exist. If the controller is modified to produce an increasing signal as long as the error is nonzero, the offset might be eliminated. This is the principle of *integral control.* In this mode

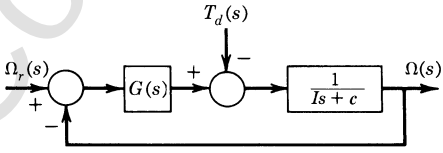


Figure 22 Simplified form of Fig. 21 for the case $K_1 = K_2$.

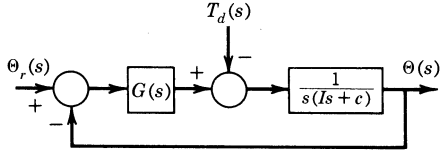


Figure 23 Position servo.

Copyright © 2006 John Wiley & Sons Retrieved from: www.knovel.com

the change in the control signal is proportional to the *integral* of the error. In the terminology of Fig. 7, this gives

$$F(s) = \frac{K_I}{s} E(s) \tag{16}$$

where $F(s)$ is the deviation in the control signal and K_I is the *integral gain.* In the time domain, the relation is

$$f(t) = K_I \int_0^t e(t)\, dt \tag{17}$$

if $f(0) = 0$. In this form, it can be seen that the integration cannot continue indefinitely because it would theoretically produce an infinite value of $f(t)$ if $e(t)$ does not change sign. This implies that special care must be taken to reinitialize a controller that uses integral action.

Integral Control of a First-Order System

Integral control of the velocity in the system of Fig. 20 has the block diagram shown in Fig. 22, where $G(s) = K/s$, $K = K_1 K_I K_T/R$. The integrating action of the amplifier is physically obtained by the techniques to be presented in Section 6 or by the digital methods presented in Section 10. The control system is stable if I, c, and K are positive. For a unit step command input, $\omega_{ss} = 1$; so the offset error is zero. For a unit step disturbance, the steady-state deviation is zero if the system is stable. Thus, the steady-state performance using integral control is excellent for this plant with step inputs. The damping ratio is $\zeta = c/2\sqrt{IK}$. For slight damping, the response will be oscillatory rather than exponential as with proportional control. Improved steady-state performance has thus been obtained at the expense of degraded transient performance. The conflict between steady-state and transient specifications is a common theme in control system design. As long as the system is underdamped, the time constant is $\tau = 2I/c$ and is not affected by the gain K, which only influences the oscillation frequency in this case. It might by physically possible to make K small enough so that $\zeta >> 1$, and the nonoscillatory feature of proportional control recovered, but the response would tend to be sluggish. Transient specifications for fast response generally require that $\zeta < 1$. The difficulty with using $\zeta < 1$ is that τ is fixed by c and I. If c and I are such that $\zeta < 1$, then τ is large if $I >> c$.

Integral Control of a Second-Order System

Proportional control of the position servomechanism in Fig. 23 gives a nonzero steady-state deviation due to the disturbance. Integral control [$G(s) = K/s$] applied to this system results in the command transfer function

$$\frac{\Theta(s)}{\Theta_r(s)} = \frac{K}{Is^3 + cs^2 + K} \tag{18}$$

With the Routh criterion, we immediately see that the system is not stable because of the missing s term. Integral control is useful in improving steady-state performance, but in general it does not improve and may even degrade transient performance. Improperly applied, it can produce an unstable control system. It is best used in conjunction with other control modes.

Copyrighted Material

Copyright © 2006 John Wiley & Sons Retrieved from: www.knovel.com

5.3 Proportional-plus-Integral Control

Integral control raised the order of the system by 1 in the preceding examples but did not give a characteristic equation with enough flexibility to achieve acceptable transient behavior. The instantaneous response of proportional control action might introduce enough variability into the coefficients of the characteristic equation to allow both steady-state and transient specifications to be satisfied. This is the basis for using *proportional-plus-integral control* (PI control). The algorithm for this two-mode control is

$$F(s) = K_P E(s) + \frac{K_I}{s} E(s) \tag{19}$$

The integral action provides an automatic, not manual, reset of the controller in the presence of a disturbance. For this reason, it is often called *reset action.*

The algorithm is sometimes expressed as

$$F(s) = K_P \left(1 + \frac{1}{T_I s}\right) E(s) \tag{20}$$

where T_I is the *reset time.* The reset time is the time required for the integral action signal to equal that of the proportional term if a constant error exists (a hypothetical situation). The reciprocal of reset time is expressed as repeats per minute and is the frequency with which the integral action repeats the proportional correction signal.

The proportional control gain must be reduced when used with integral action. The integral term does not react instantaneously to a zero-error signal but continues to correct, which tends to cause oscillations if the designer does not take this effect into account.

PI Control of a First-Order System

PI action applied to the speed controller of Fig. 20 gives the diagram shown in Fig. 21 with $G(s) = K_P + K_I/s$. The gains K_P and K_I are related to the component gains, as before. The system is stable for positive values of K_P and K_I. For $\Omega_r(s) = 1/s$, $\omega_{ss} = 1$, and the offset error is zero, as with integral action only. Similarly, the deviation due to a unit step disturbance is zero at steady state. The damping ratio is $\zeta = (c + K_P)/2\sqrt{IK_I}$. The presence of K_P allows the damping ratio to be selected without fixing the value of the dominant time constant. For example, if the system is underdamped ($\zeta < 1$), the time constant is $\tau = 2I/(c + K_P)$. The gain K_P can be picked to obtain the desired time constant, while K_I is used to set the damping ratio. A similar flexibility exists if $\zeta = 1$. Complete description of the transient response requires that the numerator dynamics present in the transfer functions be accounted for.[1,2]

PI Control of a Second-Order System

Integral control for the position servomechanism of Fig. 23 resulted in a third-order system that is unstable. With proportional action, the diagram becomes that of Fig. 22, with $G(s) = K_P + K_I/s$. The steady-state performance is acceptable, as before, if the system is assumed to be stable. This is true if the Routh criterion is satisfied, that is, if I, c, K_P, and K_I are positive and $cK_P - IK_I > 0$. The difficulty here occurs when the damping is slight. For small c, the gain K_P must be large in order to satisfy the last condition, and this can be difficult to implement physically. Such a condition can also result in an unsatisfactory time constant. The root-locus method of Section 9 provides the tools for analyzing this design further.

Copyrighted Material

Copyright © 2006 John Wiley & Sons Retrieved from: www.knovel.com

5.4 Derivative Control

Integral action tends to produce a control signal even after the error has vanished, which suggests that the controller be made aware that the error is approaching zero. One way to accomplish this is to design the controller to react to the derivative of the error with *derivative control* action, which is

$$F(s) = K_D s E(s) \tag{21}$$

where K_D is the *derivative gain.* This algorithm is also called *rate action.* It is used to damp out oscillations. Since it depends only on the error rate, derivative control should never be used alone. When used with proportional action, the following PD control algorithm results:

$$F(s) = (K_P + K_D s)E(s) = K_P(1 + T_D s)E(s) \tag{22}$$

where T_D is the *rate time* or *derivative time.* With integral action included, the *proportional-plus-integral-plus-derivative* (*PID*) *control law* is obtained:

$$F(s) = \left(K_P + \frac{K_I}{s} + K_D s\right) E(s) \tag{23}$$

This is called a *three-mode controller.*

PD Control of a Second-Order System

The presence of integral action reduces steady-state error but tends to make the system less stable. There are applications of the position servomechanism in which a nonzero derivation resulting from the disturbance can be tolerated but an improvement in transient response over the proportional control result is desired. Integral action would not be required, but rate action can be added to improve the transient response. Application of PD control to this system gives the block diagram of Fig. 23 with $G(s) = K_P + K_D s$.

The system is stable for positive values of K_D and K_P. The presence of rate action does not affect the steady-state response, and the steady-state results are identical to those with proportional control; namely, zero offset error and a deviation of $-1/K_P$, due to the disturbance. The damping ratio is $\zeta = (c + K_D)/2\sqrt{IK_P}$. For proportional control, $\zeta = c/2\sqrt{IK_P}$. Introduction of rate action allows the proportional gain K_P to be selected large to reduce the steady-state deviation, while K_D can be used to achieve an acceptable damping ratio. The rate action also helps to stabilize the system by adding damping (if $c = 0$, the system with proportional control is not stable).

The equivalent of derivative action can be obtained by using a tachometer to measure the angular velocity of the load. The block diagram is shown in Fig. 24. The gain of the amplifier–motor–potentiometer combination is K_1, and K_2 is the tachometer gain. The advantage of this system is that it does not require signal differentiation, which is difficult to

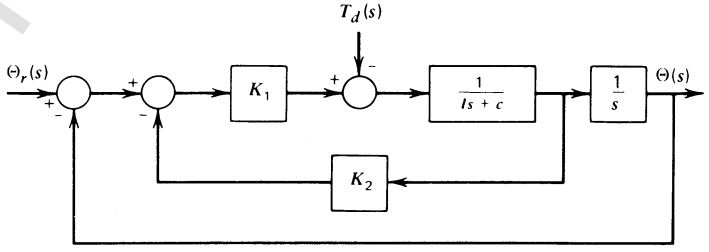


Figure 24 Tachometer feedback arrangement to replace PD control for the position servo.[1]

Copyrighted Material

Copyright © 2006 John Wiley & Sons Retrieved from: www.knovel.com

implement if signal noise is present. The gains K_1 and K_2 can be chosen to yield the desired damping ratio and steady-state deviation, as was done with K_P and K_I.

5.5 PID Control

The position servomechanism design with PI control is not completely satisfactory because of the difficulties encountered when the damping c is small. This problem can be solved by the use of the full PID control law, as shown in Fig. 23 with $G(s) = K_P + K_D s + K_I/s$.

A stable system results if all gains are positive and if $(c + K_D)K_P - IK_I > 0$. The presence of K_D relaxes somewhat the requirement that K_P be large to achieve stability. The steady-state errors are zero, and the transient response can be improved because three of the coefficients of the characteristic equation can be selected. To make further statements requires the root-locus technique presented in Section 9.

Proportional, integral, and derivative actions and their various combinations are not the only control laws possible, but they are the most common. PID controllers will remain for some time the standard against which any new designs must compete.

The conclusions reached concerning the performance of the various control laws are strictly true only for the plant model forms considered. These are the first-order model without numerator dynamics and the second-order model with a root at $s = 0$ and no numerator zeros. The analysis of a control law for any other linear system follows the preceding pattern. The overall system transfer functions are obtained, and all of the linear system analysis techniques can be applied to predict the system's performance. If the performance is unsatisfactory, a new control law is tried and the process repeated. When this process fails to achieve an acceptable design, more systematic methods of altering the system's structure are needed; they are discussed in later sections. We have used step functions as the test signals because they are the most common and perhaps represent the severest test of system performance. Impulse, ramp, and sinusoidal test signals are also employed. The type to use should be made clear in the design specifications.

6 CONTROLLER HARDWARE

The control law must be implemented by a physical device before the control engineer's task is complete. The earliest devices were purely kinematic and were mechanical elements such as gears, levers, and diaphragms that usually obtained their power from the controlled variable. Most controllers now are analog electronic, hydraulic, pneumatic, or digital electronic devices. We now consider the analog type. Digital controllers are covered starting in Section 10.

6.1 Feedback Compensation and Controller Design

Most controllers that implement versions of the PID algorithm are based on the following feedback principle. Consider the single-loop system shown in Fig. 1. If the open-loop transfer function is large enough that $|G(s)H(s)| >> 1$, the closed-loop transfer function is approximately given by

$$T(s) = \frac{G(s)}{1 + G(s)H(s)} \approx \frac{G(s)}{G(s)H(s)} = \frac{1}{H(s)} \tag{24}$$

The principle states that a power unit $G(s)$ can be used with a feedback element $H(s)$ to create a desired transfer function $T(s)$. The power unit must have a gain high enough that

Copyright © 2006 John Wiley & Sons Retrieved from: www.knovel.com

$|G(s)H(s)| >> 1$, and the feedback elements must be selected so that $H(s) = 1/T(s)$. This principle was used in Section 1 to explain the design of a feedback amplifier.

6.2 Electronic Controllers

The *operational amplifier* (*op amp*) is a high-gain amplifier with a high input impedance. A diagram of an op amp with feedback and input elements with impedances $T_f(s)$ and $T_i(s)$ is shown in Fig. 25. An approximate relation is

$$\frac{E_o(s)}{E_i(s)} = -\frac{T_f(s)}{T_i(s)}$$

The various control modes can be obtained by proper selection of the impedances. A proportional controller can be constructed with a *multiplier,* which uses two resistors, as shown in Fig. 26. An *inverter* is a multiplier circuit with $R_f = R_i$. It is sometimes needed because of the sign reversal property of the op amp. The multiplier circuit can be modified to act as an adder (Fig. 27).

PI control can be implemented with the circuit of Fig. 28. Figure 29 shows a complete system using op amps for PI control. The inverter is needed to create an error detector. Many industrial controllers provide the operator with a choice of control modes, and the operator can switch from one mode to another when the process characteristics or control objectives change. When a switch occurs, it is necessary to provide any integrators with the proper initial voltages or else undesirable transients will occur when the integrator is switched into the system. Commercially available controllers usually have built-in circuits for this purpose.

In theory, a differentiator can be created by interchanging the resistance and capacitance in the integrating op amp. The difficulty with this design is that no electrical signal is "pure." Contamination always exists as a result of voltage spikes, ripple, and other transients generally categorized as "noise." These high-frequency signals have large slopes compared with the more slowly varying primary signal, and thus they will dominate the output of the differentiator. In practice, this problem is solved by filtering out high-frequency signals, either with a low-pass filter inserted in cascade with the differentiator or by using a redesigned differentiator such as the one shown in Fig. 30. For the ideal PD controller, $R_1 = 0$. The attenuation curve for the ideal controller breaks upward at $\omega = 1/R_2C$ with a slope of 20 dB/decade. The curve for the practical controller does the same but then becomes flat for $\omega > (R_1 + R_2)/R_1R_2C$. This provides the required limiting effect at high frequencies.

PID control can be implemented by joining the PI and PD controllers in parallel, but this is expensive because of the number of op amps and power supplies required. Instead, the usual implementation is that shown in Fig. 31. The circuit limits the effect of frequencies above $\omega = 1/\beta R_1C_1$. When $R_1 = 0$, ideal PID control results. This is sometimes called the *noninteractive* algorithm because the effect of each of the three modes is additive, and they do not interfere with one another. The form given for $R_1 \neq 0$ is the *real* or *interactive*

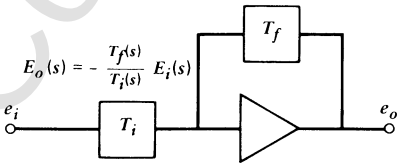


Figure 25 Operational amplifier (op amp).[1]

Copyrighted Material

Copyright © 2006 John Wiley & Sons Retrieved from: www.knovel.com

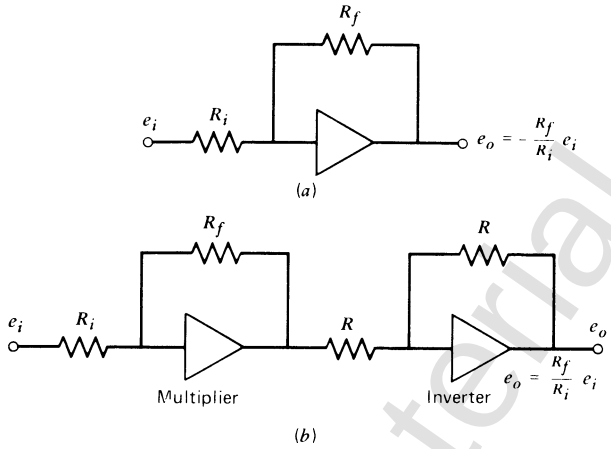


Figure 26 Op-amp implementation of proportional control.[1]

algorithm. This name results from the fact that historically it was difficult to implement noninteractive PID control with mechanical or pneumatic devices.

6.3 Pneumatic Controllers

The nozzle–flapper introduced in Section 4 is a high-gain device that is difficult to use without modification. The gain K_f is known only imprecisely and is sensitive to changes induced by temperature and other environmental factors. Also, the linear region over which Eq. (14) applies is very small. However, the device can be made useful by compensating it with feedback elements, as was illustrated with the electropneumatic valve positioner shown in Fig. 19.

6.4 Hydraulic Controllers

The basic unit for synthesis of hydraulic controllers is the hydraulic servomotor. The nozzle–flapper concept is also used in hydraulic controllers.[5] A PI controller is shown in Fig. 32. It can be modified for proportional action. Derivative action has not seen much use in hydraulic controllers. This action supplies damping to the system, but hydraulic systems are usually

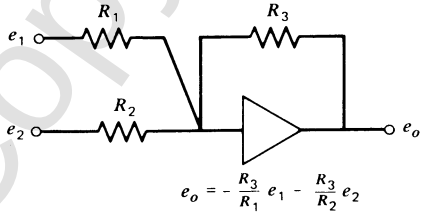


Figure 27 Op-amp adder circuit.[1]

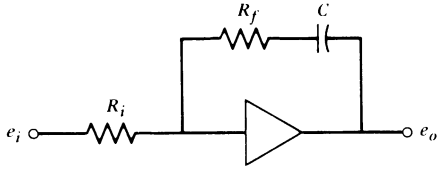


Figure 28 Op-amp implementation of PI control.[1]

Copyrighted Material

Copyright © 2006 John Wiley & Sons Retrieved from: www.knovel.com

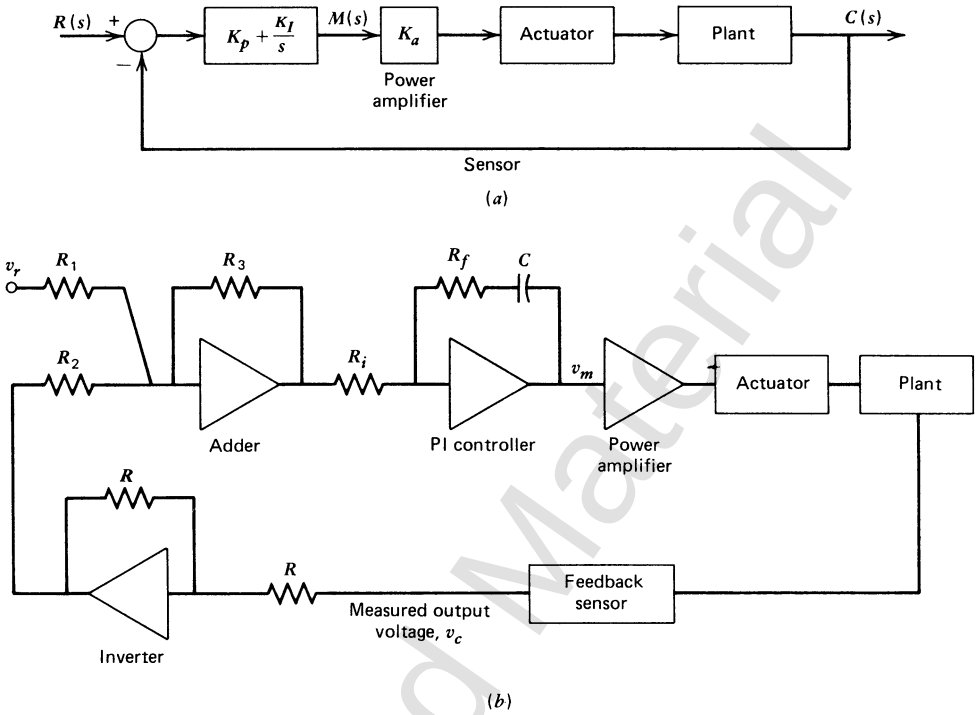
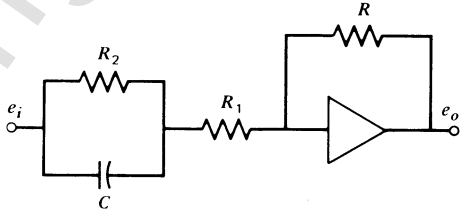


Figure 29 Implementation of a PI controller using op amps. (*a*) Diagram of the system. (*b*) Diagram showing how the op amps are connected.[2]



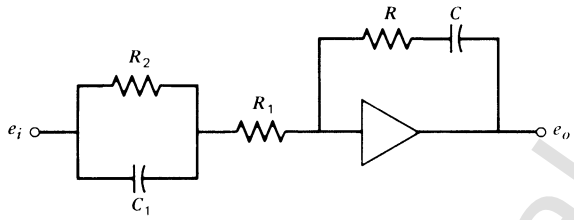
$$e_o = -K_p\left(e_i + T_D\frac{de_i}{dt}\right) - \alpha T_D\frac{de_o}{dt}$$

$$K_p = \frac{R}{R_1 + R_2} \qquad T_D = R_2C \qquad \alpha = \frac{R_1}{R_1 + R_2}$$

Figure 30 Practical op-amp implementation of PD control.[1]

Copyrighted Material

Copyright © 2006 John Wiley & Sons Retrieved from: www.knovel.com



$$e_o = -\left(K_p e_i + K_I \int_0^t e_i\,dt + K_D \frac{de_i}{dt}\right) - \beta R_1 C_1 \frac{de_o}{dt}$$

$$\beta = \frac{R_2}{R_1 + R_2} \qquad K_p = \beta \frac{RC + R_2 C_1}{R_2 C}$$

$$K_I = \frac{\beta}{R_2 C} \qquad K_D = \beta R C_1$$

Figure 31 Practical op-amp implementation of PID control.[1]

highly damped intrinsically because of the viscous working fluid. PI control is the algorithm most commonly implemented with hydraulics.

7 FURTHER CRITERIA FOR GAIN SELECTION

Once the form of the control law has been selected, the gains must be computed in light of the performance specifications. In the examples of the PID family of control laws in Section 5, the damping ratio, dominant time constant, and steady-state error were taken to be the primary indicators of system performance in the interest of simplicity. In practice, the criteria are usually more detailed. For example, the rise time and maximum overshoot, as well as the other transient response specifications of the previous chapter, may be encountered. Requirements can also be stated in terms of frequency response characteristics, such as bandwidth, resonant frequency, and peak amplitude. Whatever specific form they take, a complete

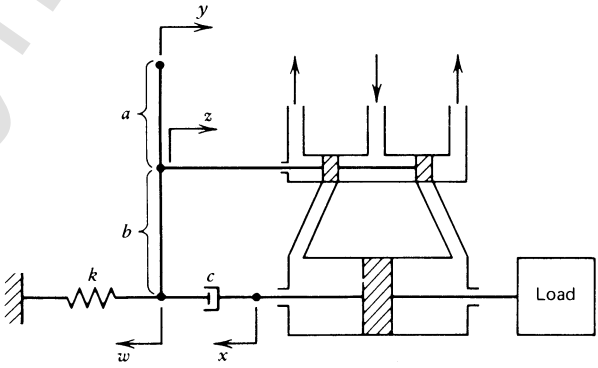


Figure 32 Hydraulic implementation of PI control.[1]

Copyrighted Material

Copyright © 2006 John Wiley & Sons Retrieved from: www.knovel.com

set of specifications for control system performance generally should include the following considerations for given forms of the command and disturbance inputs:

1. Equilibrium specifications
   - **(a)** Stability
   - **(b)** Steady-state error
2. Transient specifications
   - **(a)** Speed of response
   - **(b)** Form of response
3. Sensitivity specifications
   - **(a)** Sensitivity to parameter variations
   - **(b)** Sensitivity to model inaccuracies
   - **(c)** Noise rejection (bandwidth, etc.)

In addition to these performance stipulations, the usual engineering considerations of initial cost, weight, maintainability, and so on must be taken into account. The considerations are highly specific to the chosen hardware, and it is difficult to deal with such issues in a general way.

Two approaches exist for designing the controller. The proper one depends on the quality of the analytical description of the plant to be controlled. If an accurate model of the plant is easily developed, we can design a specialized controller for the particular application. The range of adjustment of controller gains in this case can usually be made small because the accurate plant model allows the gains to be precomputed with confidence. This technique reduces the cost of the controller and can often be applied to electromechanical systems.

The second approach is used when the plant is relatively difficult to model, which is often the case in process control. A standard controller with several control modes and wide ranges of gains is used, and the proper mode and gain settings are obtained by testing the controller on the process in the field. This approach should be considered when the cost of developing an accurate plant model might exceed the cost of controller tuning in the field. Of course, the plant must be available for testing for this approach to be feasible.

7.1 Performance Indices

The performance criteria encountered thus far require a set of conditions to be specified—for example, one for steady-state error, one for damping ratio, and one for the dominant time constant. If there are many such conditions, and if the system is of high order with several gains to be selected, the design process can get quite complicated because transient and steady-state criteria tend to drive the design in different directions. An alternative approach is to specify the system's desired performance by means of one analytical expression called a *performance index.* Powerful analytical and numerical methods are available that allow the gains to be systematically computed by minimizing (or maximizing) this index.

To be useful, a performance index must be selective. The index must have a sharply defined extremum in the vicinity of the gain values that give the desired performance. If the numerical value of the index does not change very much for large changes in the gains from their optimal values, the index will not be selective.

Any practical choice of a performance index must be easily computed, either analytically, numerically, or experimentally. Four common choices for an index are the following:

Copyright © 2006 John Wiley & Sons Retrieved from: www.knovel.com

$$J = \int_0^\infty |e(t)|\, dt \qquad \text{(IAE Index)} \tag{25}$$

$$J = \int_0^\infty t|e(t)|\, dt \qquad \text{(ITAE Index)} \tag{26}$$

$$J = \int_0^\infty [e(t)]^2\, dt \qquad \text{(ISE Index)} \tag{27}$$

$$J = \int_0^\infty t[e(t)]^2\, dt \qquad \text{(ITSE Index)} \tag{28}$$

where $e(t)$ is the system error. This error usually is the difference between the desired and the actual values of the output. However, if $e(t)$ does not approach zero as $t \to \infty$, the preceding indices will not have finite values. In this case, $e(t)$ can be defined as $e(t) = c(\infty) - c(t)$, where $c(t)$ is the output variable. If the index is to be computed numerically or experimentally, the infinite upper limit can be replaced by a time t_f large enough that $e(t)$ is negligible for $t > t_f$.

The *integral absolute-error* (IAE) criterion (25) expresses mathematically that the designer is not concerned with the sign of the error, only its magnitude. In some applications, the IAE criterion describes the fuel consumption of the system. The index says nothing about the relative importance of an error occurring late in the response versus an error occurring early. Because of this, the index is not as selective as the *integral-of-time-multiplied absolute-error* (ITAE) criterion (26). Since the multiplier t is small in the early stages of the response, this index weights early errors less heavily than later errors. This makes sense physically. No system can respond instantaneously, and the index is lenient accordingly, while penalizing any design that allows a nonzero error to remain for a long time. Neither criterion allows highly underdamped or highly overdamped systems to be optimum. The ITAE criterion usually results in a system whose step response has a slight overshoot and well-damped oscillations.

The *integral squared-error* (ISE) and *integral-of-time-multiplied squared-error* (ITSE) criteria are analogous to the IAE and ITAE criteria, except that the square of the error is employed for three reasons: (1) in some applications, the squared error represents the system's power consumption; (2) squaring the error weights large errors much more heavily than small errors; (3) the squared error is much easier to handle analytically. The derivative of a squared term is easier to compute than that of an absolute value and does not have a discontinuity at $e = 0$. These differences are important when the system is of high order with multiple error terms.

The closed-form solution for the response is not required to evaluate a performance index. For a given set of parameter values, the response and the resulting index value can be computed numerically. The optimum solution can be obtained using systematic computer search procedures; this makes this approach suitable for use with nonlinear systems.

7.2 Optimal-Control Methods

Optimal-control theory includes a number of algorithms for systematic design of a control law to minimize a performance index, such as the following generalization of the ISE index, called the *quadratic* index:

$$J = \int_0^\infty (\mathbf{x}^\mathbf{T}\mathbf{Q}\mathbf{x} + \mathbf{u}^\mathbf{T}\mathbf{R}\mathbf{u})\, dt \tag{29}$$

Copyrighted Material

Copyright © 2006 John Wiley & Sons Retrieved from: www.knovel.com

where **x** and **u** are the deviations of the state and control vectors from the desired reference values. For example, in a servomechanism, the state vector might consist of the position and velocity, and the control vector might be a scalar—the force or torque produced by the actuator. The matrices **Q** and **R** are chosen by the designer to provide relative weighting for the elements of **x** and **u**. If the plant can be described by the linear state-variable model

$$\dot{\mathbf{x}} = \mathbf{Ax} + \mathbf{Bu} \tag{30}$$

$$\mathbf{y} = \mathbf{Cx} + \mathbf{Du} \tag{31}$$

where **y** is the vector of outputs—for example, position and velocity—then the solution of this *linear-quadratic* control problem is the linear control law:

$$\mathbf{u} = \mathbf{Ky} \tag{32}$$

where **K** is a matrix of gains that can be found by several algorithms.[1,6,7] A valid solution is guaranteed to yield a stable closed-loop system, a major benefit of this method.

Even if it is possible to formulate the control problem in this way, several practical difficulties arise. Some of the terms in (29) might be beyond the influence of the control vector **u**; the system is then *uncontrollable.* Also, there might not be enough information in the output equation (31) to achieve control, and the system is then *unobservable.* Several tests are available to check controllability and observability. Not all of the necessary state variables might be available for feedback or the feedback measurements might be noisy or biased. Algorithms known as *observers, state reconstructors, estimators,* and *digital filters* are available to compensate for the missing information. Another source of error is the uncertainty in the values of the coefficient matrices **A**, **B**, **C**, and **D**. Identification schemes can be used to compare the predicted and the actual system performance and to adjust the coefficient values ''on-line.''

7.3 The Ziegler–Nichols Rules

The difficulty of obtaining accurate transfer function models for some processes has led to the development of empirically based rules of thumb for computing the optimum gain values for a controller. Commonly used guidelines are the *Ziegler–Nichols rules,* which have proved so helpful that they are still in use 50 years after their development. The rules actually consist of two separate methods. The first method requires the open-loop step response of the plant, while the second uses the results of experiments performed with the controller already installed. While primarily intended for use with systems for which no analytical model is available, the rules are also helpful even when a model can be developed.

Ziegler and Nichols developed their rules from experiments and analysis of various industrial processes. Using the IAE criterion with a unit step response, they found that controllers adjusted according to the following rules usually had a step response that was oscillatory but with enough damping so that the second overshoot was less than 25% of the first (peak) overshoot. This is the *quarter-decay* criterion and is sometimes used as a specification.

The first method is the *process reaction* method and relies on the fact that many processes have an open-loop step response like that shown in Fig. 33. This is the *process signature* and is characterized by two parameters, R and L, where R is the slope of a line tangent to the steepest part of the response curve and L is the time at which this line intersects the time axis. First- and second-order linear systems do not yield positive values for L, and so the method cannot be applied to such systems. However, third- and higher order linear systems with sufficient damping do yield such a response. If so, the Ziegler–Nichols rules recommend the controller settings given in Table 2.

Copyright © 2006 John Wiley & Sons Retrieved from: www.knovel.com

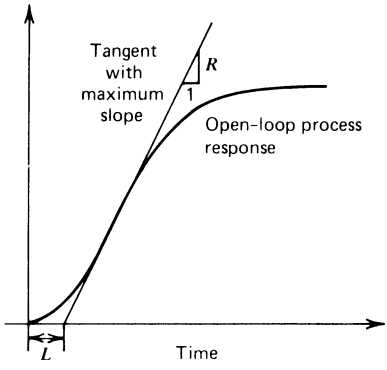


Figure 33 Process signature for a unit step input.[1]

The *ultimate-cycle* method uses experiments with the controller in place. All control modes except proportional are turned off, and the process is started with the proportional gain K_P set at a low value. The gain is slowly increased until the process begins to exhibit sustained oscillations. Denote the period of this oscillation by P_u and the corresponding *ultimate gain* by K_{Pu}. The Ziegler–Nichols recommendations are given in Table 2 in terms of these parameters. The proportional gain is lower for PI control than for proportional control and is higher for PID control because integral action increases the order of the system and thus tends to destabilize it; thus, a lower gain is needed. On the other hand, derivative action tends to stabilize the system; hence, the proportional gain can be increased without degrading the stability characteristics. Because the rules were developed for a typical case out of many types of processes, final tuning of the gains in the field is usually necessary.

7.4 Nonlinearities and Controller Performance

All physical systems have nonlinear characteristics of some sort, although they can often be modeled as linear systems provided the deviations from the linearization reference condition are not too great. Under certain conditions, however, the nonlinearities have significant effects

Table 2 The Ziegler–Nichols Rules

Controller transfer function $G(s) = K_p\left(1 + \frac{1}{T_I s} + T_D s\right)$		
Control Mode	Process Reaction Method	Ultimate-Cycle Method
P control	$K_p = \frac{1}{RL}$	$K_p = 0.5K_{pu}$
PI control	$K_p = \frac{0.9}{RL}$	$K_p = 0.45K_{pu}$
	$T_I = 3.3L$	$T_I = 0.83P_u$
PID control	$K_p = \frac{1.2}{RL}$	$K_p = 0.6K_{pu}$
	$T_I = 2L$	$T_I = 0.5P_u$
	$T_D = 0.5L$	$T_D = 0.125P_u$

Copyrighted Material

Copyright © 2006 John Wiley & Sons Retrieved from: www.knovel.com

on the system's performance. One such situation can occur during the start-up of a controller if the initial conditions are much different from the reference condition for linearization. The linearized model is then not accurate, and nonlinearities govern the behavior. If the nonlinearities are mild, there might not be much of a problem. Where the nonlinearities are severe, such as in process control, special consideration must be given to start-up. Usually, in such cases, the control signal sent to the final control elements is manually adjusted until the system variables are within the linear range of the controller. Then the system is switched into automatic mode. Digital computers are often used to replace the manual adjustment process because they can be readily coded to produce complicated functions for the start-up signals. Care must also be taken when switching from manual to automatic. For example, the integrators in electronic controllers must be provided with the proper initial conditions.

7.5 Reset Windup

In practice, all actuators and final control elements have a limited operating range. For example, a motor–amplifier combination can produce a torque proportional to the input voltage over only a limited range. No amplifier can supply an infinite current; there is a maximum current and thus a maximum torque that the system can produce. The final control elements are said to be *overdriven* when they are commanded by the controller to do something they cannot do. Since the limitations of the final control elements are ultimately due to the limited rate at which they can supply energy, it is important that all system performance specifications and controller designs be consistent with the energy delivery capabilities of the elements to be used.

Controllers using integral action can exhibit the phenomenon called *reset windup* or *integrator buildup* when overdriven, if they are not properly designed. For a step change in set point, the proportional term responds instantly and saturates immediately if the set-point change is large enough. On the other hand, the integral term does not respond as fast. It integrates the error signal and saturates some time later if the error remains large for a long enough time. As the error decreases, the proportional term no longer causes saturation. However, the integral term continues to increase as long as the error has not changed sign, and thus the manipulated variable remains saturated. Even though the output is very near its desired value, the manipulated variable remains saturated until after the error has reversed sign. The result can be an undesirable overshoot in the response of the controlled variable.

Limits on the controller prevent the voltages from exceeding the value required to saturate the actuator and thus protect the actuator, but they do not prevent the integral buildup that causes the overshoot. One way to prevent integrator buildup is to select the gains so that saturation will never occur. This requires knowledge of the maximum input magnitude that the system will encounter. General algorithms for doing this are not available; some methods for low-order systems are presented in Ref. 1, Chapter 7; Ref. 2, Chapter 7, and Ref. 4, Chapter 11. Integrator buildup is easier to prevent when using digital control; this is discussed in Section 10.

8 COMPENSATION AND ALTERNATIVE CONTROL STRUCTURES

A common design technique is to insert a *compensator* into the system when the PID control algorithm can be made to satisfy most but not all of the design specifications. A compensator is a device that alters the response of the controller so that the overall system will have satisfactory performance. The three categories of compensation techniques generally recog-

Copyrighted Material

Copyright © 2006 John Wiley & Sons Retrieved from: www.knovel.com

nized are *series compensation, parallel* (or *feedback*) *compensation,* and *feedforward compensation.* The three structures are loosely illustrated in Fig. 34, where we assume the final control elements have a unity transfer function. The transfer function of the controller is $G_1(s)$. The feedback elements are represented by $H(s)$, and the compensator by $G_c(s)$. We assume that the plant is unalterable, as is usually the case in control system design. The choice of compensation structure depends on what type of specifications must be satisfied. The physical devices used as compensators are similar to the pneumatic, hydraulic, and electrical devices treated previously. Compensators can be implemented in software for digital control applications.

8.1 Series Compensation

The most commonly used series compensators are the *lead,* the *lag,* and the *lead–lag* compensators. Electrical implementations of these are shown in Fig. 35. Other physical implementations are available. Generally, the lead compensator improves the speed of response; the lag compensator decreases the steady-state error; and the lead–lag affects both. Graphical aids, such as the root-locus and frequency response plots, are usually needed to design these compensators (Ref. 1, Chapter 8; Ref. 2, Chapter 9; and Ref. 4, Chapter 11).

8.2 Feedback Compensation and Cascade Control

The use of a tachometer to obtain velocity feedback, as in Fig. 24, is a case of feedback compensation. The feedback compensation principle of Fig. 3 is another. Another form is *cascade control,* in which another controller is inserted within the loop of the original control system (Fig. 36). The new controller can be used to achieve better control of variables within the forward path of the system. Its set point is manipulated by the first controller.

Cascade control is frequently used when the plant cannot be satisfactorily approximated with a model of second order or lower. This is because the difficulty of analysis and control increases rapidly with system order. The characteristic roots of a second-order system can easily be expressed in analytical form. This is not so for third order or higher, and few general design rules are available. When faced with the problem of controlling a high-order system, the designer should first see if the performance requirements can be relaxed so that the system can be approximated with a low-order model. If this is not possible, the designer should attempt to divide the plant into subsystems, each of which is second order or lower. A controller is then designed for each subsystem. An application using cascade control is given in Section 11.

8.3 Feedforward Compensation

The control algorithms considered thus far have counteracted disturbances by using measurements of the output. One difficulty with this approach is that the effects of the disturbance must show up in the output of the plant before the controller can begin to take action. On the other hand, if we can measure the disturbance, the response of the controller can be improved by using the measurement to augment the control signal sent from the controller to the final control elements. This is the essence of feedforward compensation of the disturbance, as shown in Fig. 34*c*.

Feedforward compensation modified the output of the main controller. Instead of doing this by measuring the disturbance, another form of feedforward compensation utilizes the

Copyrighted Material

Copyright © 2006 John Wiley & Sons Retrieved from: www.knovel.com

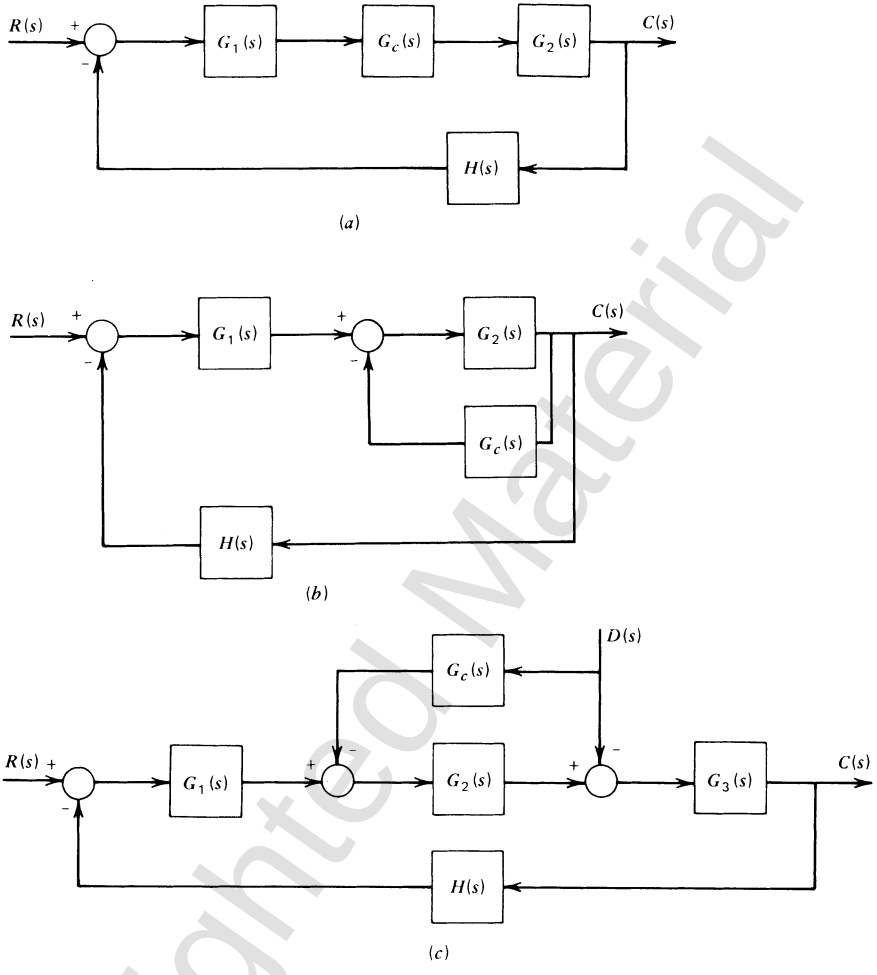


Figure 34 General structures of the three compensation types: (*a*) series; (*b*) parallel (or feedback); (*c*) feedforward. The compensator transfer function is $G_c(s)$.[1]

command input. Figure 37 is an example of this approach. The closed-loop transfer function is

$$\frac{\Omega(s)}{\Omega_r(s)} = \frac{K_f + K}{Is + c + K} \tag{33}$$

For a unit step input, the steady-state output is $\omega_{ss} = (K_f + K)/(c + K)$. Thus, if we choose the feedforward gain K_f to be $K_f = c$, then $\omega_{ss} = 1$ as desired, and the error is zero. Note that this form of feedforward compensation does not affect the disturbance response. Its effectiveness depends on how accurately we know the value of c. A digital application of feedforward compensation is presented in Section 11.

Copyrighted Material

Copyright © 2006 John Wiley & Sons Retrieved from: www.knovel.com

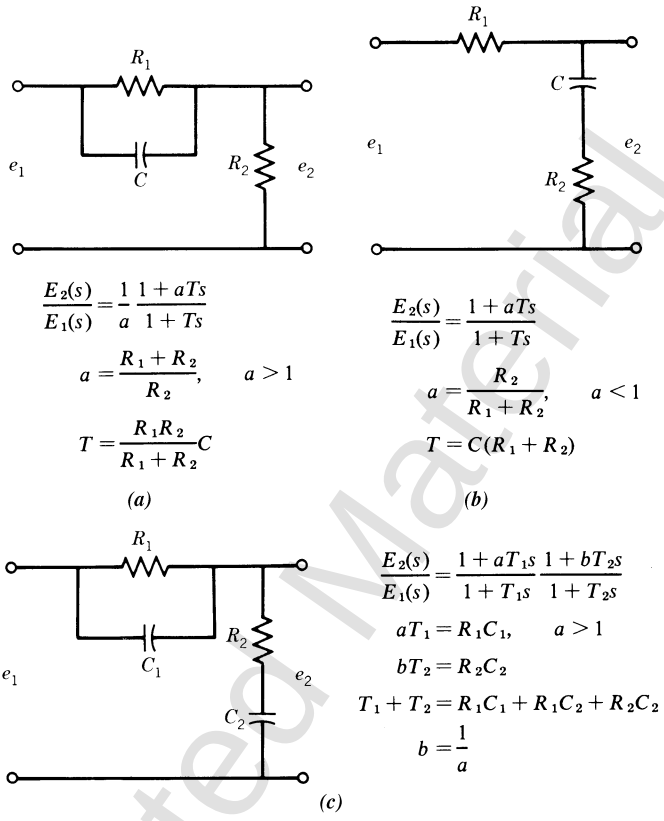


Figure 35 Passive electrical compensators: (*a*) lead; (*b*) lag; (*c*) lead–lag.

8.4 State-Variable Feedback

There are techniques for improving system performance that do not fall entirely into one of the three compensation categories considered previously. In some forms these techniques can be viewed as a type of feedback compensation, while in other forms they constitute a modification of the control law. *State-variable feedback* (SVFB) is a technique that uses information about all the system's state variables to modify either the control signal or the

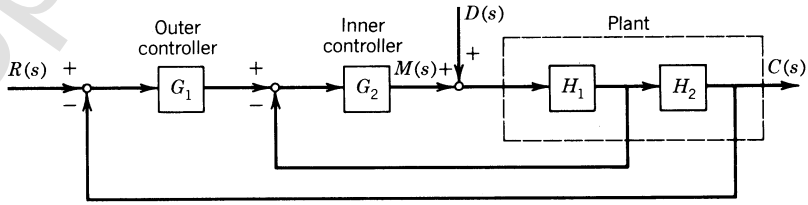


Figure 36 Cascade control structure.

Copyright © 2006 John Wiley & Sons Retrieved from: www.knovel.com

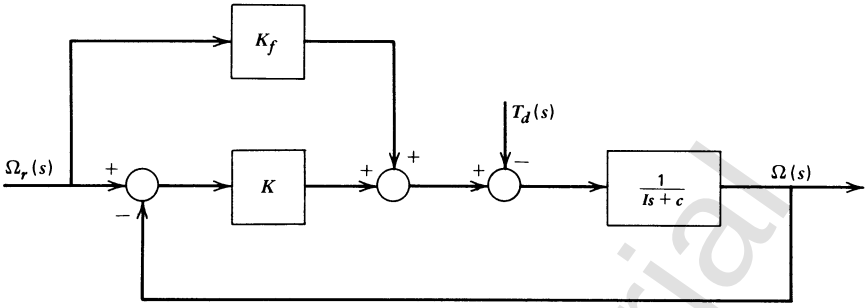


Figure 37 Feedforward compensation of the command input to augment proportional control.[2]

actuating signal. These two forms are illustrated in Fig. 38. Both forms require that the state vector $\boldsymbol{x}$ be measurable or at least derivable from other information. Devices or algorithms used to obtain state-variable information other than directly from measurements are variously termed *state reconstructors, estimators, observers,* or *filters* in the literature.

8.5 Pseudoderivative Feedback

Pseudoderivative feedback (PDF) is an extension of the velocity feedback compensation concept of Fig. 24.[1,2] It uses integral action in the forward path plus an internal feedback loop whose operator $H(s)$ depends on the plant (Fig. 39). For $G(s) = 1/(Is + c)$, $H(s) = K_1$. For $G(s) = 1/Is^2$, $H(s) = K_1 + K_2 s$. The primary advantage of PDF is that it does not need derivative action in the forward path to achieve the desired stability and damping characteristics.

9 GRAPHICAL DESIGN METHODS

Higher order models commonly arise in control systems design. For example, integral action is often used with a second-order plant, and this produces a third-order system to be designed. Although algebraic solutions are available for third- and fourth-order polynomials, these solutions are cumbersome for design purposes. Fortunately, there exist graphical techniques to aid the designer. Frequency response plots of both the open- and closed-loop transfer

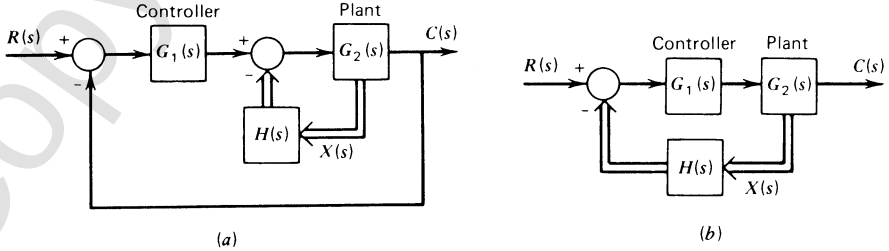


Figure 38 Two forms of state-variable feedback: (*a*) internal compensation of the control signal; (*b*) modification of the actuating signal.[1]

Copyrighted Material

Copyright © 2006 John Wiley & Sons Retrieved from: www.knovel.com

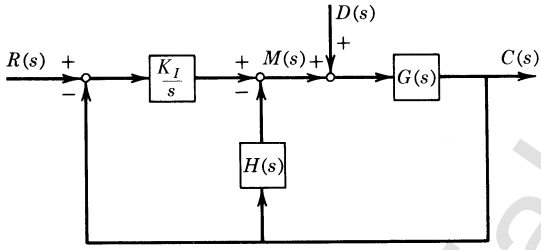


Figure 39 Structure of pseudoderivative feedback.

functions are useful. The *Bode plot* and the *Nyquist plot* present the frequency response information in different forms. Each form has its own advantages. The root-locus plot shows the location of the characteristic roots for a range of values of some parameters, such as a controller gain. A tabulation of these plots for typical transfer functions is given in the previous chapter (Fig. 27.8). The design of two-position and other nonlinear control systems is facilitated by the *describing function,* which is a linearized approximation based on the frequency response of the controller (see Section 27.8.4). Graphical design methods are discussed in more detail in Refs. 1–4.

9.1 The Nyquist Stability Theorem

The Nyquist stability theorem is a powerful tool for linear system analysis. If the open-loop system has no poles with positive real parts, we can concentrate our attention on the region around the point $-1 + i0$ on the polar plot of the open-loop transfer function. Figure 40 shows the polar plot of the open-loop transfer function of an arbitrary system that is assumed to be open-loop stable. The Nyquist stability theorem is stated as follows:

> A system is closed-loop stable if and only if the point $-1 + i0$ lies to the left of the open-loop Nyquist plot relative to an observer traveling along the plot in the direction of increasing frequency ω.

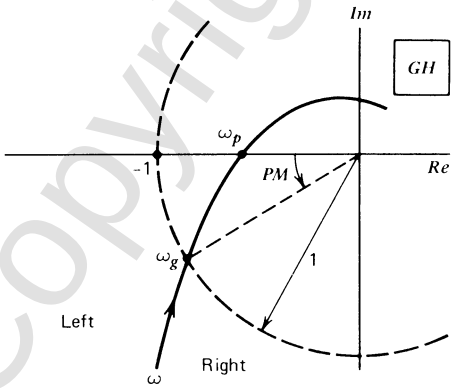


Figure 40 Nyquist plot for a stable system.[1]

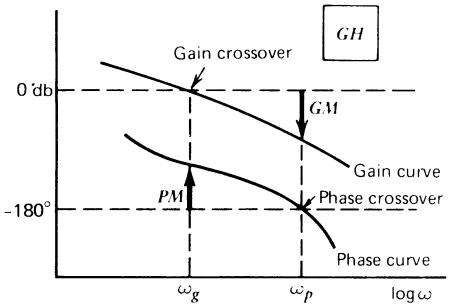


Figure 41 Bode plot showing definitions of phase and gain margin.[1]

Copyrighted Material

Copyright © 2006 John Wiley & Sons Retrieved from: www.knovel.com

Therefore, the system described by Fig. 39 is closed-loop stable.

The Nyquist theorem provides a convenient measure of the relative stability of a system. A measure of the proximity of the plot to the $-1 + i0$ point is given by the angle between the negative real axis and a line from the origin to the point where the plot crosses the unit circle (see Fig. 39). The frequency corresponding to this intersection is denoted ω_g. This angle is the *phase margin* (PM) and is positive when measured down from the negative real axis. The phase margin is the phase at the frequency ω_g where the magnitude ratio or "gain" of $G(i\omega)H(i\omega)$ is unity, or 0 decibels (dB). The frequency ω_p, the *phase crossover frequency,* is the frequency at which the phase angle is $-180°$. The *gain margin* (GM) is the difference in decibels between the unity gain condition (0 dB) and the value of $|G(\omega_p)H(\omega_p)|$ decibels at the phase crossover frequency ω_p. Thus,

$$\text{Gain margin} = -|G(\omega_p)H(\omega_p)| \quad \text{(dB)} \tag{34}$$

A system is stable only if the phase and gain margins are both positive.

The phase and gain margins can be illustrated on the Bode plots shown in Fig. 41. The phase and gain margins can be stated as safety margins in the design specifications. A typical set of such specifications is as follows:

$$\text{Gain margin} \geq 8 \text{ dB} \qquad \text{and} \qquad \text{Phase margin} \geq 30° \tag{35}$$

In common design situations, only one of these equalities can be met, and the other margin is allowed to be greater than its minimum value. It is not desirable to make the margins too large, because this results in a low gain, which might produce sluggish response and a large steady-state error. Another commonly used set of specifications is

$$\text{Gain margin} \geq 6 \text{ dB} \qquad \text{and} \qquad \text{Phase margin} \geq 40° \tag{36}$$

The 6-dB limit corresponds to the quarter amplitude decay response obtained with the gain settings given by the Ziegler–Nichols ultimate-cycle method (Table 2).

9.2 Systems with Dead-Time Elements

The Nyquist theorem is particularly useful for systems with dead-time elements, especially when the plant is of an order high enough to make the root-locus method cumbersome. A delay D in either the manipulated variable or the measurement will result in an open-loop transfer function of the form

$$G(s)H(s) = e^{-Ds}P(s) \tag{37}$$

Its magnitude and phase angle are

$$|G(i\omega)H(i\omega)| = |P(i\omega)||e^{-i\omega D}| = |P(i\omega)| \tag{38}$$

$$\angle G(i\omega)H(i\omega)) = \angle P(i\omega) + \angle e^{-i\omega D} = \angle P(i\omega) - \omega D \tag{39}$$

Thus, the dead time decreases the phase angle proportionally to the frequency ω, but it does not change the gain curve. This makes the analysis of its effects easier to accomplish with the open-loop frequency response plot.

9.3 Open-Loop Design for PID Control

Some general comments can be made about the effects of proportional, integral, and derivative control actions on the phase and gain margins. P action does not affect the phase curve at all and thus can be used to raise or lower the open-loop gain curve until the specifications

Copyrighted Material

Copyright © 2006 John Wiley & Sons Retrieved from: www.knovel.com

for the gain and phase margins are satisfied. If I action or D action is included, the proportional gain is selected last. Therefore, when using this approach to the design, it is best to write the PID algorithm with the proportional gain factored out, as

$$F(s) = K_P \left(1 + \frac{1}{T_I s} + T_D s\right) E(s) \tag{40}$$

D action affects both the phase and gain curves. Therefore, the selection of the derivative gain is more difficult than the proportional gain. The increase in phase margin due to the positive phase angle introduced by D action is partly negated by the derivative gain, which reduces the gain margin. Increasing the derivative gain increases the speed of response, makes the system more stable, and allows a larger proportional gain to be used to improve the system's accuracy. However, if the phase curve is too steep near $-180°$, it is difficult to use D action to improve the performance. I action also affects both the gain and phase curves. It can be used to increase the open-loop gain at low frequencies. However, it lowers the phase crossover frequency ω_p and thus reduces some of the benefits provided by D action. If required, the D-action term is usually designed first, followed by I action and P action, respectively.

The classical design methods based on the Bode plots obviously have a large component of trial and error because usually both the phase and gain curves must be manipulated to achieve an acceptable design. Given the same set of specifications, two designers can use these methods and arrive at substantially different designs. Many rules of thumb and ad hoc procedures have been developed, but a general foolproof procedure does not exist. However, an experienced designer can often obtain a good design quickly with these techniques. The use of a computer plotting routine greatly speeds up the design process.

9.4 Design with the Root Locus

The effect of D action as a series compensator can be seen with the root locus. The term $1 + T_D s$ in Fig. 32 can be considered as a series compensator to the proportional controller. The D action adds an open-loop zero at $s = -1/T_D$. For example, a plant with the transfer function $1/s(s + 1)(s + 2)$, when subjected to proportional control, has the root locus shown in Fig. 42*a*. If the proportional gain is too high, the system will be unstable. The smallest achievable time constant corresponds to the root $s = -0.42$ and is $\tau = 1/0.42 = 2.4$. If D action is used to put an open-loop zero at $s = -1.5$, the resulting root locus is given by Fig. 42*b*. The D action prevents the system from becoming unstable and allows a smaller time constant to be achieved (τ can be made close to $1/0.75 = 1.3$ by using a high proportional gain).

The integral action in PI control can be considered to add an open-loop pole at $s = 0$ and a zero at $s = -1/T_I$. Proportional control of the plant $1/(s + 1)(s + 2)$ gives a root locus like that shown in Fig. 43, with $a = 1$ and $b = 2$. A steady-state error will exist for a step input. With the PI compensator applied to this plant, the root locus is given by Fig. 42*b*, with $T_I = 2/3$. The steady-state error is eliminated, but the response of the system has been slowed because the dominant paths of the root locus of the compensated system lie closer to the imaginary axis than those of the uncompensated system.

As another example, let the plant transfer function be

$$G_P(s) = \frac{1}{s^2 + a_2 s + a_1} \tag{41}$$

where $a_1 > 0$ and $a_2 > 0$. PI control applied to this plant gives the closed-loop command transfer function

Copyrighted Material

Copyright © 2006 John Wiley & Sons Retrieved from: www.knovel.com

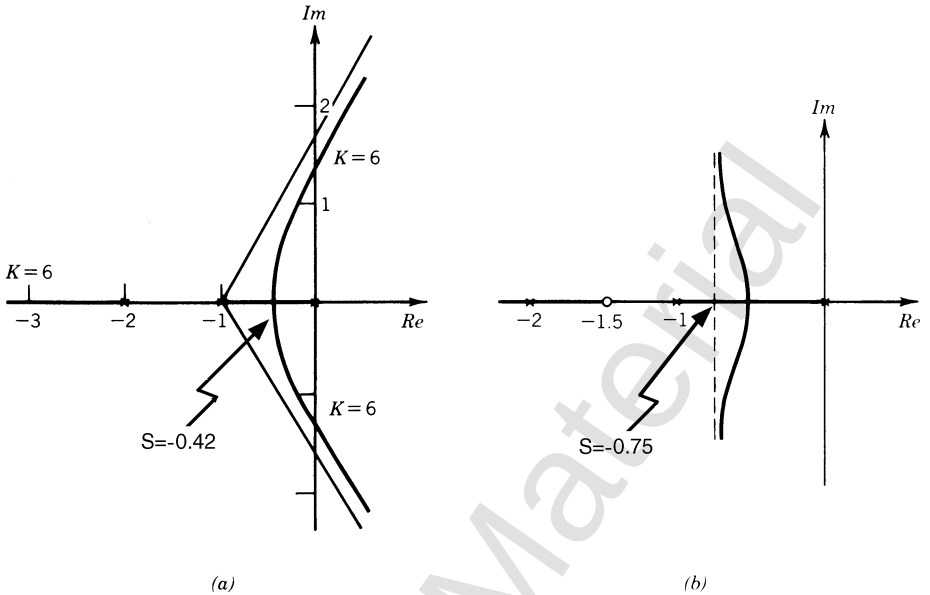


Figure 42 (*a*) Root-locus plot for $s(s + 1)(s + 2) + K = 0$, for $K \geq 0$. (*b*) The effect of PD control with $T_D = \frac{2}{3}$.

$$T_1(s) = \frac{K_P s + K_I}{s^3 + a_2 s^2 + (a_1 + K_P)s + K_I} \tag{42}$$

Note that the Ziegler–Nichols rules cannot be used to set the gains K_P and K_I. The second-order plant, Eq. (41), does not have the S-shaped signature of Fig. 33, so the process reaction method does not apply. The ultimate-cycle method requires K_I to be set to zero and the ultimate gain K_{Pu} determined. With $K_I = 0$ in Eq. (42) the resulting system is stable for all $K_P > 0$, and thus a positive ultimate gain does not exist.

Take the form of the PI control law given by Eq. (42) with $T_D = 0$, and assume that the characteristic roots of the plant (Fig. 44) are real values $-r_1$ and $-r_2$ such that $-r_2 < -r_1$. In this case the open-loop transfer function of the control system is

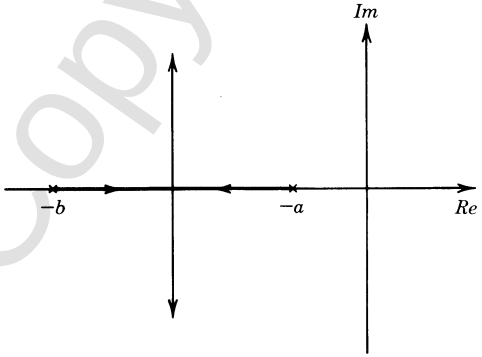


Figure 43 Root-locus plot for $(s + a)(s + b) + K = 0$.

Copyrighted Material

Copyright © 2006 John Wiley & Sons Retrieved from: www.knovel.com

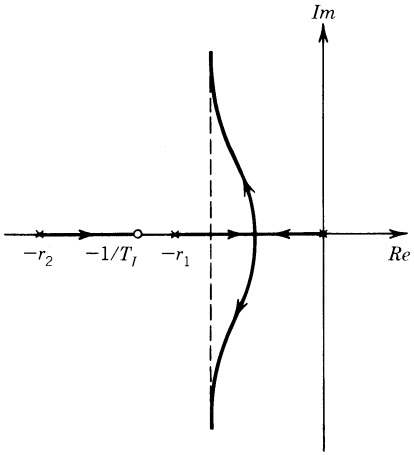


Figure 44 Root-locus plot for PI control of a second-order plant.

$$G(s)H(s) = \frac{K_P(s + 1/T_I)}{s(s + r_1)(s + r_2)} \tag{43}$$

One design approach is to select T_I and plot the locus with K_P as the parameter. If the zero at $s = -1/T_I$ is located to the right of $s = -r_1$, the dominant time constant cannot be made as small as is possible with the zero located between the poles at $s = -r_1$ and $s = -r_2$ (Fig. 44). A large integral gain (small T_I and/or large K_P) is desirable for reducing the overshoot due to a disturbance, but the zero should not be placed to the left of $s = -r_2$ because the dominant time constant will be larger than that obtainable with the placement shown in Fig. 44 for large values of K_P. Sketch the root-locus plots to see this. A similar situation exists if the poles of the plant are complex.

The effects of the lead compensator in terms of time-domain specifications (characteristic roots) can be shown with the root-locus plot. Consider the second-order plant with the real distinct roots $s = -\alpha$, $s = -\beta$. The root locus for this system with proportional control is shown in Fig. 45*a*. The smallest dominant time constant obtainable is τ_1, marked in the figure. A lead compensator introduces a pole at $s = -1/T$ and a zero at $s = -1/aT$, and the root locus becomes that shown in Fig. 45*b*. The pole and zero introduced by the compensator reshape the locus so that a smaller dominant time constant can be obtained. This is done by choosing the proportional gain high enough to place the roots close to the asymptotes.

With reference to the proportional control system whose root locus is shown in Fig. 45*a*, suppose that the desired damping ratio ζ_1 and desired time constant τ_1 are obtainable with a proportional gain of K_{P1}, but the resulting steady-state error $\alpha\beta/(\alpha\beta + K_{P1})$ due to a step input is too large. We need to increase the gain while preserving the desired damping ratio and time constant. With the lag compensator, the root locus is as shown in Fig. 45*c*. By considering specific numerical values, one can show that for the compensated system, roots with a damping ratio ζ_1 correspond to a high value of the proportional gain. Call this value K_{P2}. Thus $K_{P2} > K_{P1}$, and the steady-state error will be reduced. If the value of T is chosen large enough, the pole at $s = -1/T$ is approximately canceled by the zero at $s = -1/aT$, and the open-loop transfer function is given approximately by

$$G(s)H(s) = \frac{aK_P}{(s + \alpha)(s + \beta)} \tag{44}$$

Copyrighted Material

Copyright © 2006 John Wiley & Sons Retrieved from: www.knovel.com

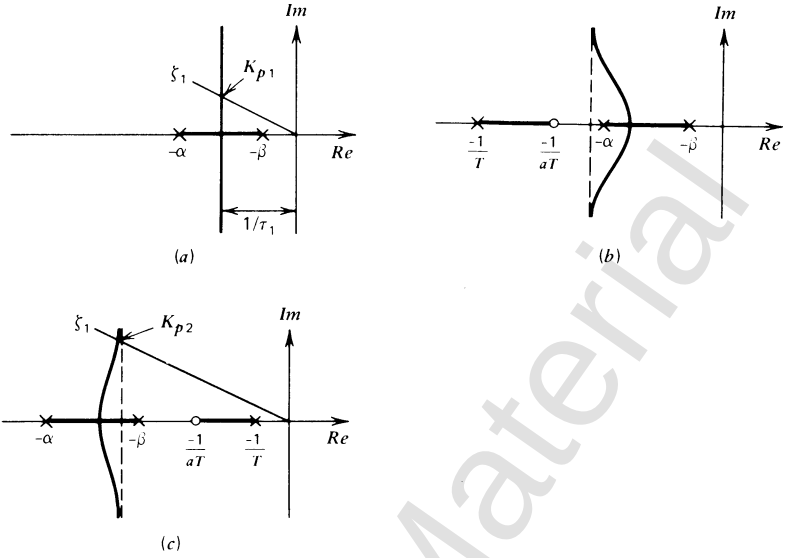


Figure 45 Effects of series lead and lag compensators: (*a*) uncompensated system's root locus; (*b*) root locus with lead compensation; (*c*) root locus with lag compensation.[1]

Thus, the system's response is governed approximately by the complex roots corresponding to the gain value K_{P2}. By comparing Fig. 45*a* with 45*c*, we see that the compensation leaves the time constant relatively unchanged. From Eq. (44) it can be seen that since $a < 1$, K_P can be selected as the larger value K_{P2}. The ratio of K_{P1} to K_{P2} is approximately given by the parameter a.

Design by pole–zero cancellation can be difficult to accomplish because a response pattern of the system is essentially ignored. The pattern corresponds to the behavior generated by the canceled pole and zero, and this response can be shown to be beyond the influence of the controller. In this example, the canceled pole gives a stable response because it lies in the left-hand plane. However, another input not modeled here, such as a disturbance, might excite the response and cause unexpected behavior. The designer should therefore proceed with caution. None of the physical parameters of the system are known exactly, so exact pole–zero cancellation is not possible. A root-locus study of the effects of parameter uncertainty and a simulation study of the response are often advised before the design is accepted as final.

10 PRINCIPLES OF DIGITAL CONTROL

Digital control has several advantages over analog devices. A greater variety of control algorithms is possible, including nonlinear algorithms and ones with time-varying coefficients. Also, greater accuracy is possible with digital systems. However, their additional hardware complexity can result in lower reliability, and their application is limited to signals whose time variation is slow enough to be handled by the samplers and the logic circuitry. This is now less of a problem because of the large increase in the speed of digital systems.

Copyrighted Material

Copyright © 2006 John Wiley & Sons Retrieved from: www.knovel.com

10.1 Digital Controller Structure

Sampling, discrete-time models, the z-transform, and pulse transfer functions were outlined in the previous chapter. The basic structure of a single-loop controller is shown in Fig. 46. The computer with its internal clock drives the *digital-to-analog* (D/A) and *analog-to-digital* (A/D) converters. It compares the command signals with the feedback signals and generates the control signals to be sent to the final control elements. These control signals are computed from the control algorithm stored in the memory. Slightly different structures exist, but Fig. 46 shows the important aspects. For example, the comparison between the command and feedback signals can be done with analog elements, and the A/D conversion made on the resulting error signal. The software must also provide for *interrupts,* which are conditions that call for the computer's attention to do something other than computing the control algorithm.

The time required for the control system to complete one loop of the algorithm is the time T, the *sampling time* of the control system. It depends on the time required for the computer to calculate the control algorithm and on the time required for the interfaces to convert data. Modern systems are capable of very high rates, with sample times under 1 μs.

In most digital control applications, the plant is an analog system, but the controller is a discrete-time system. Thus, to design a digital control system, we must either model the controller as an analog system or model the plant as a discrete-time system. Each approach has its own merits, and we will examine both.

If we model the controller as an analog system, we use methods based on *differential* equations to compute the gains. However, a digital control system requires *difference* equations to describe its behavior. Thus, from a strictly mathematical point of view, the gain values we will compute will not give the predicted response exactly. However, if the sampling time is small compared to the smallest time constant in the system, then the digital system will act like an analog system, and our designs will work properly. Because most physical systems of interest have time constants greater than 1 ms and controllers can now achieve sampling times less than 1 μs, controllers designed with analog methods will often be adequate.

10.2 Digital Forms of PID Control

There are a number of ways that PID control can be implemented in software in a digital control system, because the integral and derivative terms must be approximated with formulas chosen from a variety of available algorithms. The simplest integral approximation is to replace the integral with a sum of rectangular areas. With this rectangular approximation, the error integral is calculated as

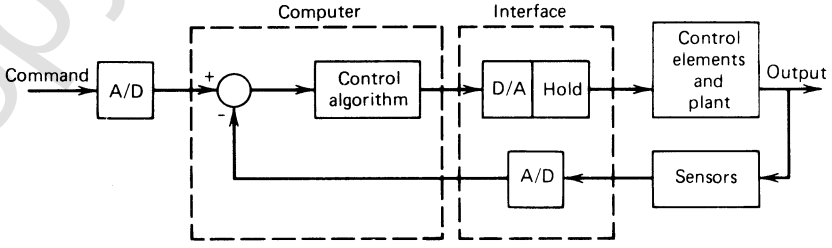


Figure 46 Structure of a digital control system.[1]

Copyrighted material

Copyright © 2006 John Wiley & Sons Retrieved from: www.knovel.com

$$\int_0^{(k+1)T} e(t)\,dt \approx Te(0) + Te(t_1) + Te(t_2) + \cdots + Te(t_k) = T\sum_{i=0}^{k} e(t_i) \tag{45}$$

where $t_k = kT$ and the width of each rectangle is the sampling time $T = t_{i+1} - t_i$. The times t_i are the times at which the computer updates its calculation of the control algorithm after receiving an updated command signal and an updated measurement from the sensor through the A/D interfaces. If the time T is small, then the value of the sum in (45) is close to the value of the integral. After the control algorithm calculation is made, the calculated value of the control signal $f(t_k)$ is sent to the actuator via the output interface. This interface includes a D/A converter and a *hold* circuit that "holds" or keeps the analog voltage corresponding to the control signal applied to the actuator until the next updated value is passed along from the computer. The simplest digital form of PI control uses (45) for the integral term. It is

$$f(t_k) = K_P e(t_k) + K_I T\sum_{i=0}^{k} e(t_i) \tag{46}$$

This can be written in a more efficient form by noting that

$$f(t_{k-1}) = K_P e(t_{k-1}) + K_I T\sum_{i=0}^{k-1} e(t_i)$$

and subtracting this from (46) to obtain

$$f(t_k) = f(t_{k-1}) + K_P[e(t_k) - e(t_{k-1})] + K_I T e(t_k) \tag{47}$$

This form—called the *incremental* or *velocity* algorithm—is well suited for incremental output devices such as stepper motors. Its use also avoids the problem of integrator buildup, the condition in which the actuator saturates but the control algorithm continues to integrate the error.

The simplest approximation to the derivative is the first-order difference approximation

$$\frac{de}{dt} \approx \frac{e(t_k) - e(t_{k-1})}{T} \tag{48}$$

The corresponding PID approximation using the rectangular integral approximation is

$$f(t_k) = K_P e(t_k) + K_I T\sum_{i=0}^{k} e(t_i) + \frac{K_D}{T}[e(t_k) - e(t_{k-1})] \tag{49}$$

The accuracy of the integral approximation can be improved by substituting a more sophisticated algorithm, such as the following trapezoidal rule:

$$\int_0^{(k+1)T} e(t)\,dt \approx T\sum_{i=0}^{k}\frac{1}{2}[e(t_{i+1} + e(t_i)] \tag{50}$$

The accuracy of the derivative approximation can be improved by using values of the sampled error signal at more instants. Using the four-point central-difference method (Refs. 1 and 2), the derivative term is approximated by

$$\frac{de}{dt} \approx \frac{1}{6T}[e(t_k) + 3e(t_{k-1}) - 3e(t_{k-2}) - e(t_{k-3})]$$

Copyrighted Material

Copyright © 2006 John Wiley & Sons Retrieved from: www.knovel.com

The derivative action is sensitive to the resulting rapid change in the error samples that follows a step input. This effect can be eliminated by reformulating the control algorithm as follows (Refs. 1 and 2):

$$f(t_k) = f(t_{k-1}) + K_P[c(t_{k-1}) - c(t_k)] + K_I T[r(t_k) - c(t_k)] + \frac{K_D}{T}[-c(t_k) + 2c(t_{k-1}) - c(t_{k-2})] \tag{51}$$

where $r(t_k)$ is the command input and $c(t_k)$ is the variable being controlled. Because the command input $r(t_k)$ appears in this algorithm only in the integral term, we cannot apply this algorithm to PD control; that is, the integral gain K_I must be nonzero.

11 UNIQUELY DIGITAL ALGORITHMS

Development of analog control algorithms was constrained by the need to design physical devices that could implement the algorithm. However, digital control algorithms simply need to be programmable and are thus less constrained than analog algorithms.

11.1 Digital Feedforward Compensation

Classical control system design methods depend on linear models of the plant. With linearization we can obtain an approximately linear model, which is valid only over a limited operating range. Digital control now allows us to deal with nonlinear models more directly using the concepts of feedforward compensation discussed in Section 8.

Computed Torque Method

Figure 47 illustrates a variation of feedforward compensation of the disturbance called the *computed torque method*. It is used to control the motion of robots. A simple model of a robot arm is the following nonlinear equation:

$$I\ddot{\theta} = T - mgL \sin \theta \tag{52}$$

where θ is the arm angle, I is its inertia, mg is its weight, and L is the distance from its mass center to the arm joint where the motor acts. The motor supplies the torque T. To position the arm at some desired angle θ_r, we can use PID control on the angle error $\theta_r - \theta$. This works well if the arm angle θ is never far from the desired angle θ_r so that we can linearize the plant model about θ_r. However, the controller will work for large-angle excursions if we compute the nonlinear gravity torque term $mgL \sin \theta$ and add it to the PID output. That is, part of the motor torque will be computed specifically to cancel the gravity torque, in effect producing a linear system for the PID algorithm to handle. The nonlinear torque calculations required to control multi-degree-of-freedom robots are very complicated and can be done only with a digital controller.

Feedforward Command Compensation

Computers can store lookup tables, which can be used to control systems that are difficult to model entirely with differential equations and analytical functions. Figure 48 shows a speed control system for an internal combustion engine. The fuel flow rate required to achieve

Copyrighted Material

Copyright © 2006 John Wiley & Sons Retrieved from: www.knovel.com

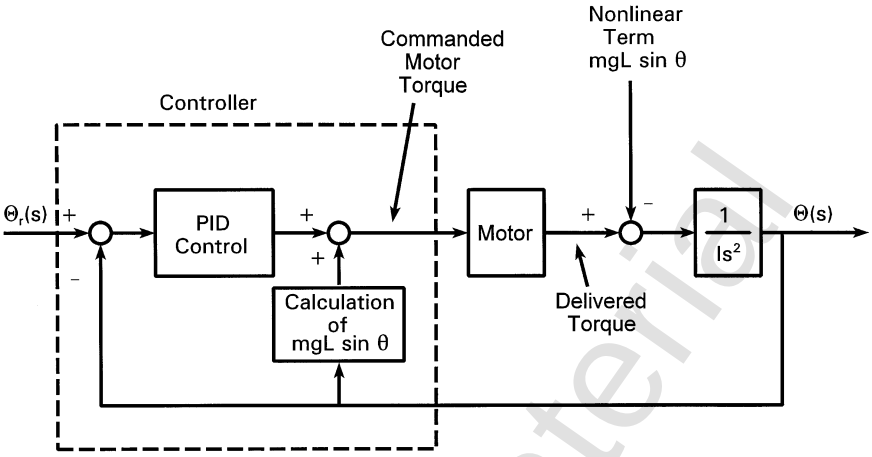


Figure 47 The computed torque method applied to robot arm control.

a desired speed depends in a complicated way on many variables not shown in the figure, such as temperature, humidity, and so on. This dependence can be summarized in tables stored in the control computer and can be used to estimate the required fuel flow rate. A PID algorithm can be used to adjust the estimate based on the speed error. This application is an example of feedforward compensation of the command input, and it requires a digital computer.

11.2 Control Design in the *z*-Plane

There are two common approaches to designing a digital controller:

1. The performance is specified in terms of the desired continuous-time response, and the controller design is done entirely in the *s*-plane, as with an analog controller. The

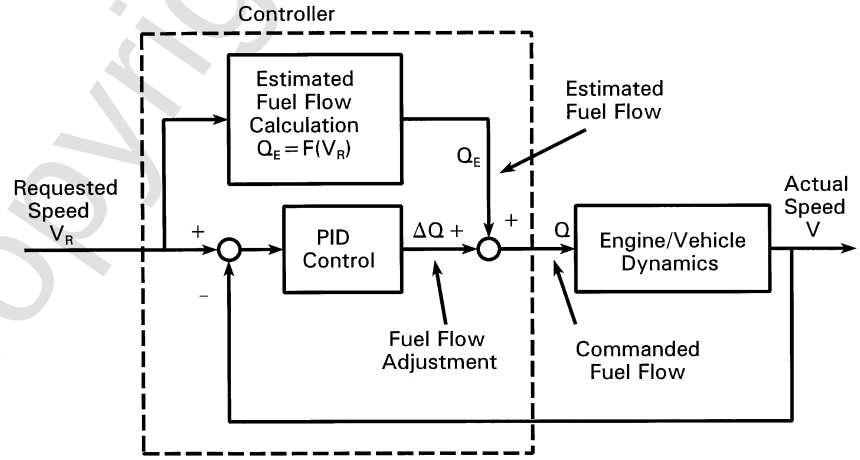


Figure 48 Feedforward compensation applied to engine control.

Copyrighted Material

Copyright © 2006 John Wiley & Sons Retrieved from: www.knovel.com

resulting control law is then converted to discrete-time form, using approximations for the integral and derivative terms. This method can be successfully applied if the sampling time is small. The technique is widely used for two reasons. When existing analog controllers are converted to digital control, the form of the control law and the values of its associated gains are known to have been satisfactory. Therefore, the digital version can use the same control law and gain values. Second, because analog design methods are well established, many engineers prefer to take this route and then convert the design into a discrete-time equivalent.

2. The performance specifications are given in terms of the desired continuous-time response and/or desired root locations in the s-plane. From these the corresponding root locations in the z-plane are found and a discrete control law is designed. This method avoids the derivative and integral approximation errors that are inherent in the first method and is the preferred method when the sampling time T is large. However, the algebraic manipulations are more cumbersome.

The second approach uses the z-transform and pulse transfer functions, which were outlined in the previous chapter. If we have an analog model of the plant, with its transfer function $G(s)$, we can obtain its pulse transfer function $G(z)$ by finding the z-transform of the impulse response $g(t) = \mathcal{L}^{-1}[G(s)]$; that is, $G(z) = \mathcal{Z}[g(t)]$. A table of transforms facilitates this process; see Refs. 1 and 2. Figure 49*a* shows the basic elements of a digital control system. Figure 49*b* is an equivalent diagram with the analog transfer functions inserted. Figure 49*c* represents the same system in terms of pulse transfer functions. From the diagram we can find the closed-loop pulse transfer function. It is

$$\frac{C(z)}{R(z)} = \frac{G(z)P(z)}{1 + G(z)P(z)} \tag{53}$$

The variable z is related to the Laplace variable s by

$$z = e^{sT} \tag{54}$$

If we know the desired root locations and the sampling time T, we can compute the z roots from this equation.

Digital PI Control Design

For example, the first-order plant $1/(2s + 1)$ with a zero-order hold has the following pulse transfer function (Refs. 1 and 2):

$$P(z) = \frac{1 - e^{-0.5T}}{z - e^{-0.5T}} \tag{55}$$

Suppose we use a control algorithm described by the following pulse transfer function:

$$G(z) = \frac{F(z)}{E(z)} = \frac{K_1 z + K_2}{z - 1} = \frac{K_1 + K_2 z^{-1}}{1 - z^{-1}} \tag{56}$$

The corresponding difference equation that the control computer must implement is

$$f(t_k) = f(t_{k-1}) + K_1 e(t_k) + K_2 e(t_{k-1}) \tag{57}$$

where $e(t_k) = r(t_k) - c(t_k)$. By comparing (57) with (47), it can be seen that this is the digital equivalent of PI control, where $K_P = -K_2$ and $K_I = (K_1 + K_2)/T$. Using the form of $G(z)$ given by (56), the closed-loop transfer function is

Copyrighted Material

Copyright © 2006 John Wiley & Sons Retrieved from: www.knovel.com

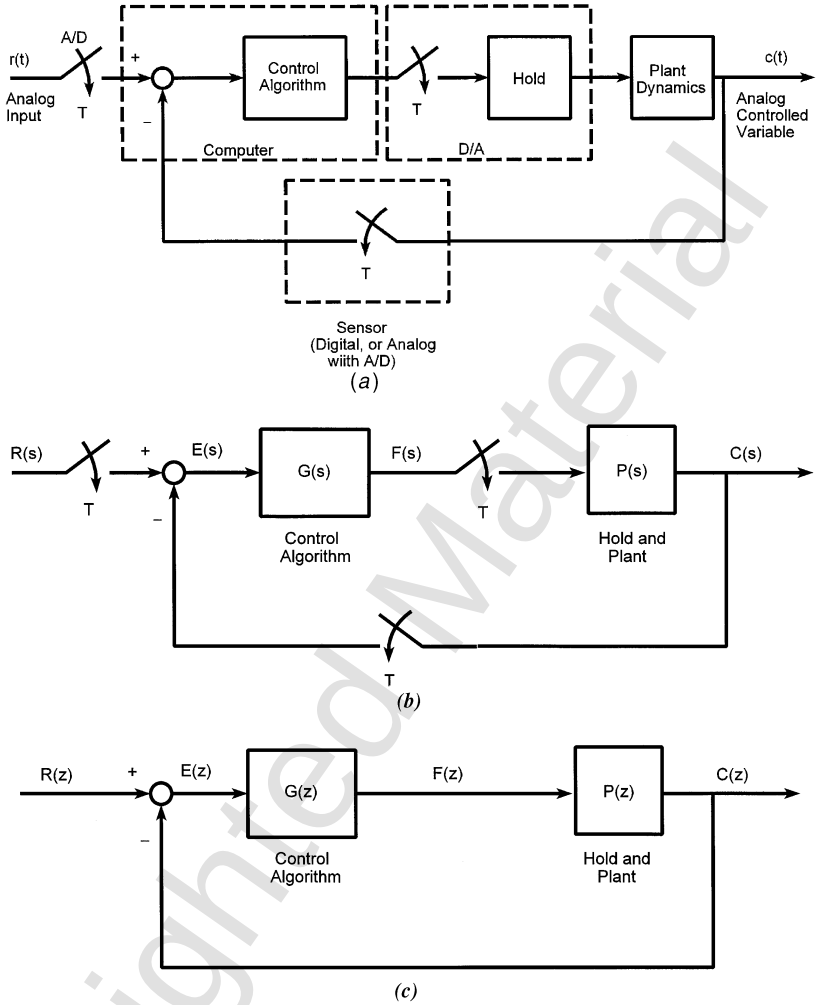


Figure 49 Block diagrams of a typical digital controller. (*a*) Diagram showing the components. (*b*) Diagram of the *s*-plane relations. (*c*) Diagram of the *z*-plane relations.

$$\frac{C(z)}{R(z)} = \frac{(1 - b)(K_1 z + K_2)}{z^2 + (K_1 - 1 - b - bK_1)z + b + K_2 - bK_2} \tag{58}$$

where $b = e^{-05T}$.

If the design specifications call for $\tau = 1$ and $\zeta = 1$, then the desired s roots are $s = -1, -1$, and the analog PI gains required to achieve these roots are $K_P = 3$ and $K_I = 2$. Using a sampling time of $T = 0.1$, the z roots must be $z = e^{-0.1}, e^{-0.1}$. To achieve these roots, the denominator of the transfer function (58) must be $z^2 - 2e^{-0.1}z + e^{-0.2}$. Thus the control gains must be $K_1 = 2.903$ and $K_2 = -2.717$. These values of K_1 and K_2 correspond to $K_P = 2.72$ and $K_I = 1.86$, which are close to the PI gains computed for an analog controller. If we had used a sampling time smaller than 0.1, say $T = 0.01$, the values of K_P and K_I computed from K_1 and K_2 would be $K_P = 2.97$ and $K_I = 1.98$, which are even closer

Copyrighted Material

Copyright © 2006 John Wiley & Sons Retrieved from: www.knovel.com

to the analog gain values. This illustrates the earlier claim that analog design methods can be used when the sampling time is small enough.

Digital Series Compensation

Series compensation can be implemented digitally by applying suitable discrete-time approximations for the derivative and integral to the model represented by the compensator's transfer function $G_c(s)$. For example, the form of a lead or a lag compensator's transfer function is

$$G_c(s) = \frac{M(s)}{F(s)} = K\frac{s + c}{s + d} \tag{59}$$

where $m(t)$ is the actuator command and $f(t)$ is the control signal produced by the main (PID) controller. The differential equation corresponding to (59) is

$$\dot{m} + dm = K(\dot{f} + cf) \tag{60}$$

Using the simplest approximation for the derivative, Eq. (48), we obtain the following difference equation that the digital compensator must implement:

$$\frac{m(t_k) - m(t_{k-1})}{T} + dm(t_k) = K\left[\frac{f(t_k) - f(t_{k-1})}{T} + cf(t_k)\right]$$

In the z-plane, the equation becomes

$$\frac{1 - z^{-1}}{T}M(z) + dM(z) = K\left[\frac{1 - z^{-1}}{T}F(z) + cF(z)\right] \tag{61}$$

The compensator's pulse transfer function is thus seen to be

$$G_c(z) = \frac{M(z)}{F(z)} = \frac{K(1 - z^{-1}) + cT}{1 - z^{-1} + dT}$$

which has the form

$$G_c(z) = K_c\frac{z + a}{z + b} \tag{62}$$

where K_c, a, and b can be expressed in terms of K, c, d, and T if we wish to use analog design methods to design the compensator. When using commercial controllers, the user might be required to enter the values of the gain, the pole, and the zero of the compensator. The user must ascertain whether these values should be entered as s-plane values (i.e., K, c, and d) or as z-plane values (K_c, a, and b).

Note that the digital compensator has the same number of poles and zeros as the analog compensator. This is a result of the simple approximation used for the derivative. Note that Eq. (61) shows that when we use this approximation, we can simply replace s in the analog transfer function with $1 - z^{-1}$. Because the integration operation is the inverse of differentiation, we can replace $1/s$ with $1/(1 - z^{-1})$ when integration is used. [This is equivalent to using the rectangular approximation for the integral, and can be verified by finding the pulse transfer function of the incremental algorithm (47) with $K_P = 0$.]

Some commercial controllers treat the PID algorithm as a series compensator, and the user is expected to enter the controller's values, not as PID gains, but as pole and zero locations in the z-plane. The PID transfer function is

Copyrighted Material

Copyright © 2006 John Wiley & Sons Retrieved from: www.knovel.com

$$\frac{F(s)}{E(s)} = K_P + \frac{K_I}{s} + K_D s \tag{63}$$

Making the indicated replacements for the s terms, we obtain

$$\frac{F(z)}{E(z)} = K_P + \frac{K_I}{1 - z^{-1}} + K_D(1 - z^{-1})$$

which has the form

$$\frac{F(z)}{E(z)} = K_c \frac{z^2 - az + b}{z - 1} \tag{64}$$

where K_c, a, and b can be expressed in terms of K_P, K_I, K_D, and T. Note that the algorithm has two zeros and one pole, which is fixed at $z = 1$. Sometimes the algorithm is expressed in the more general form

$$\frac{F(z)}{E(z)} = K_c \frac{z^2 - az + b}{z - c} \tag{65}$$

to allow the user to select the pole as well.

Digital compensator design can be done with frequency response methods or with the root-locus plot applied to the z-plane rather than the s-plane. However, when better approximations are used for the derivative and integral, the digital series compensator will have more poles and zeros than its analog counterpart. This means that the root-locus plot will have more root paths, and the analysis will be more difficult. This topic is discussed in more detail in Refs. 1–3 and 8.

11.3 Direct Design of Digital Algorithms

Because almost any algorithm can be implemented digitally, we can specify the desired response and work backward to find the required control algorithm. This is the *direct-design* method. If we let $D(z)$ be the desired form of the closed-loop transfer function $C(z)/R(z)$ and solve (53 in Chapter 14) for the controller transfer function $G(z)$, we obtain

$$G(z) = \frac{D(z)}{P(z)[1 - D(z)]} \tag{66}$$

We can pick $D(z)$ directly or obtain it from the specified input transform $R(z)$ and the desired output transform $C(z)$, because $D(z) = C(z)/R(z)$.

Finite-Settling-Time Algorithm

This method can be used to design a controller to compensate for the effects of process dead time. A plant having such a response can often be approximately described by a first-order model with a dead-time element; that is,

$$G_P(s) = K \frac{e^{-Ds}}{\tau s + 1} \tag{67}$$

where D is the dead time. This model also approximately describes the S-shaped response curve used with the Ziegler–Nichols method (Fig. 33). When combined with a zero-order hold, this plant has the following pulse transfer function:

Copyrighted Material

Copyright © 2006 John Wiley & Sons Retrieved from: www.knovel.com

$$P(z) = Kz^{-n}\frac{1-a}{z-a} \tag{68}$$

where $a = \exp(-T/\tau)$ and $n = D/T$. If we choose $D(z) = z^{-(n+1)}$, then with a step command input, the output $c(k)$ will reach its desired value in $n + 1$ sample times, one more than is in the dead time D. This is the fastest response possible. From (66) the required controller transfer function is

$$G(z) = \frac{1}{K(1-a)}\frac{1-az^{-1}}{1-z^{-(n+1)}} \tag{69}$$

The corresponding difference equation that the control computer must implement is

$$f(t_k) = f(t_{k-n-1}) + \frac{1}{K(1-a)}[e(t_k) - ae(t_{k-1})] \tag{70}$$

This algorithm is called a *finite-settling-time* algorithm because the response reaches its desired value in a finite, prescribed time. The maximum value of the manipulated variable required by this algorithm occurs at $t = 0$ and is $1/K(1 - a)$. If this value saturates the actuator, this method will not work as predicted. Its success depends also on the accuracy of the plant model.

Dahlin's Algorithm

This sensitivity to plant modeling errors can be reduced by relaxing the minimum-response-time requirement. For example, choosing $D(z)$ to have the same form as $P(z)$, namely,

$$D(z) = K_d z^{-n}\frac{1-a_d}{z-a_d} \tag{71}$$

we obtain from (66) the following controller transfer function:

$$G(z) = \frac{K_d(1-a_d)}{K(1-a)}\frac{1-az^{-1}}{1-a_dz^{-1}-K_d(1-a_d)z^{-(n+1)}} \tag{72}$$

This is *Dahlin's algorithm.*[3] The corresponding difference equation that the control computer must implement is

$$f(t_k) = a_d f(t_{k-1}) + K_d(1-a_d)f(t_{k-n-1}) + \frac{K_d(1-a_d)}{K(1-a)}[e(t_k) - ae(t_{k-1})] \tag{73}$$

Normally we would first try setting $K_d = K$ and $a_d = a$, but since we might not have good estimates of K and a, we can use K_d and a_d as tuning parameters to adjust the controller's performance. The constant a_d is related to the time constant τ_d of the desired response: $a_d = \exp(-T/\tau_d)$. Choosing τ_d smaller gives faster response.

Algorithms such as these are often used for system startup, after which the control mode is switched to PID, which is more capable of handling disturbances.

12 HARDWARE AND SOFTWARE FOR DIGITAL CONTROL

This section provides an overview of the general categories of digital controllers that are commercially available. This is followed by a summary of the software currently available for digital control and for control system design.

Copyrighted Material

Copyright © 2006 John Wiley & Sons Retrieved from: www.knovel.com

12.1 Digital Control Hardware

Commercially available controllers have different capabilities, such as different speeds and operator interfaces, depending on their targeted application.

Programmable Logic Controllers (PLCs)

These are controllers that are programmed with relay ladder logic, which is based on Boolean algebra. Now designed around microprocessors, they are the successors to the large relay panels, mechanical counters, and drum programmers used up to the 1960s for sequencing control and control applications requiring only a finite set of output values (for example, opening and closing of valves). Some models now have the ability to perform advanced mathematical calculations required for PID control, thus allowing them to be used for modulated control as well as finite-state control. There are numerous manufacturers of PLCs.

Digital Signal Processors (DSPs)

A modern development is the *digital signal processor* (DSP), which has proved useful for feedback control as well as signal processing.[9] This special type of processor chip has separate buses for moving data and instructions and is constructed to perform rapidly the kind of mathematical operations required for digital filtering and signal processing. The separate buses allow the data and the instructions to move in parallel rather than sequentially. Because the PID control algorithm can be written in the form of a digital filter, DSPs can also be used as controllers.

The DSP architecture was developed to handle the types of calculations required for digital filters and discrete Fourier transforms, which form the basis of most signal-processing operations. DSPs usually lack the extensive memory management capabilities of general-purpose computers because they need not store large programs or large amounts of data. Some DSPs contain A/D and D/A converters, serial ports, timers, and other features. They are programmed with specialized software that runs on popular personal computers. Low-cost DSPs are now widely used in consumer electronics and automotive applications, with Texas Instruments being a major supplier.

Motion Controllers

Motion controllers are specialized control systems that provide feedback control for one or more motors. They also provide a convenient operator interface for generating the commanded trajectories. Motion controllers are particularly well suited for applications requiring coordinated motion of two or more axes and for applications where the commanded trajectory is complicated. A higher level host computer might transmit required distance, speed, and acceleration rates to the motion controller, which then constructs and implements the continuous position profile required for each motor. For example, the host computer would supply the required total displacement, the acceleration and deceleration times, and the desired slew speed (the speed during the zero acceleration phase). The motion controller would generate the commanded position versus time for each motor. The motion controller also has the task of providing feedback control for each motor to ensure that the system follows the required position profile.

Figure 50 shows the functional elements of a typical motion controller, such as those built by Galil Motion Control, Inc. Provision for both analog and digital input signals allows these controllers to perform other control tasks besides motion control. Compared to DSPs, such controllers generally have greater capabilities for motion control and have operator interfaces that are better suited for such applications. Motion controllers are available as plug-in cards for most computer bus types. Some are available as stand-alone units.

Copyrighted Material

Copyright © 2006 John Wiley & Sons Retrieved from: www.knovel.com

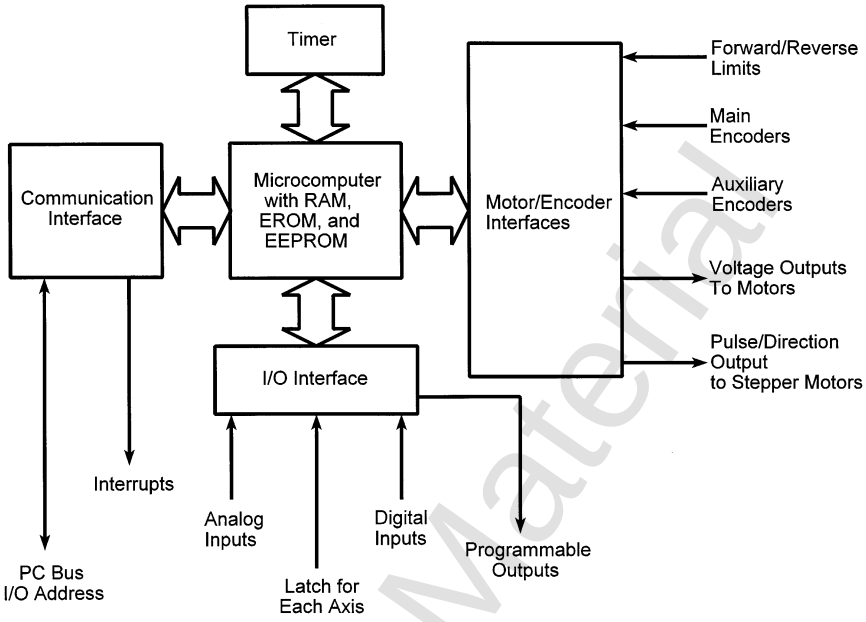


Figure 50 Functional diagram of a motion controller.

Motion controllers use a PID control algorithm to provide feedback control for each motor (some manufacturers call this algorithm a "filter"). The user enters the values of the PID gains (some manufacturers provide preset gain values, which can be changed; others provide tuning software that assists in selecting the proper gain values). Such controllers also have their own language for programming a variety of motion profiles and other applications. For example, they provide for linear and circular interpolation for two-dimensional coordinated motion, motion smoothing (to eliminate jerk), contouring, helical motion, and electronic gearing. The latter is a control mode that emulates mechanical gearing in software, in which one motor (the slave) is driven in proportion to the position of another motor (the master) or an encoder.

Process Controllers

Process controllers are designed to handle inputs from sensors, such as thermocouples, and outputs to actuators, such as valve positioners, that are commonly found in process control applications. Figure 51 illustrates the input–output capabilities of a typical process controller such as those manufactured by Honeywell, which is a major supplier of such devices. This device is a stand-alone unit designed to be mounted in an instrumentation panel. The voltage and current ranges of the analog inputs are those normally found with thermocouple-based temperature sensors. The current outputs are designed for devices like valve positioners, which usually require 4–20-mA signals.

The controller contains a microcomputer with built-in math functions normally required for process control, such as thermocouple linearization, weighted averaging, square roots, ratio/bias calculations, and the PID control algorithm. These controllers do not have the same software and memory capabilities as desktop computers, but they are less expensive. Their operator interface consists of a small keypad with typically fewer than 10 keys, a small

Copyright © 2006 John Wiley & Sons Retrieved from: www.knovel.com

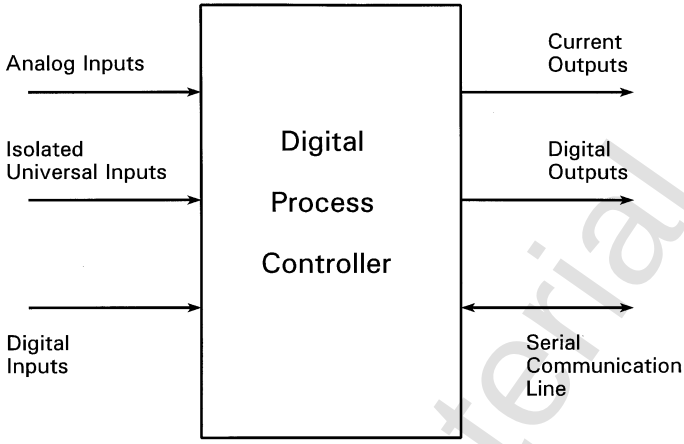


Figure 51 Functional diagram of a digital process controller.

graphical display for displaying bargraphs of the set points and the process variables, indicator lights, and an alphanumeric display for programming the controller.

The PID gains are entered by the user. Some units allow multiple sets of gains to be stored; the unit can be programmed to switch between gain settings when certain conditions occur. Some controllers have an adaptive tuning feature that is supposed to adjust the gains to prevent overshoot in startup mode, to adapt to changing process dynamics, and to adapt to disturbances. However, at this time, adaptive tuning cannot claim a 100% success rate, and further research and development in adaptive control is needed.

Some process controllers have more than one PID control loop for controlling several variables. Figure 52 illustrates a boiler feedwater control application for a controller with two PID loops arranged in a cascade control structure. Loop 1 is the main or outer loop controller for maintaining the desired water volume in the boiler. It uses sensing of the steam flow rate to implement feedforward compensation. Loop 2 is the inner loop controller that directly controls the feedwater control valve.

12.2 Software for Digital Control

The software available to the modern control engineer is quite varied and powerful and can be categorized according to the following tasks:

1. Control algorithm design, gain selection, and simulation
2. Tuning
3. Motion programming
4. Instrumentation configuration
5. Real-time control functions

Many analysis and simulation packages now contain algorithms of specific interest to control system designers. *MATLAB*® is one such package that is widely used. It contains built-in functions for generating root-locus and frequency response plots, system simulation,

Copyright © 2006 John Wiley & Sons Retrieved from: www.knovel.com

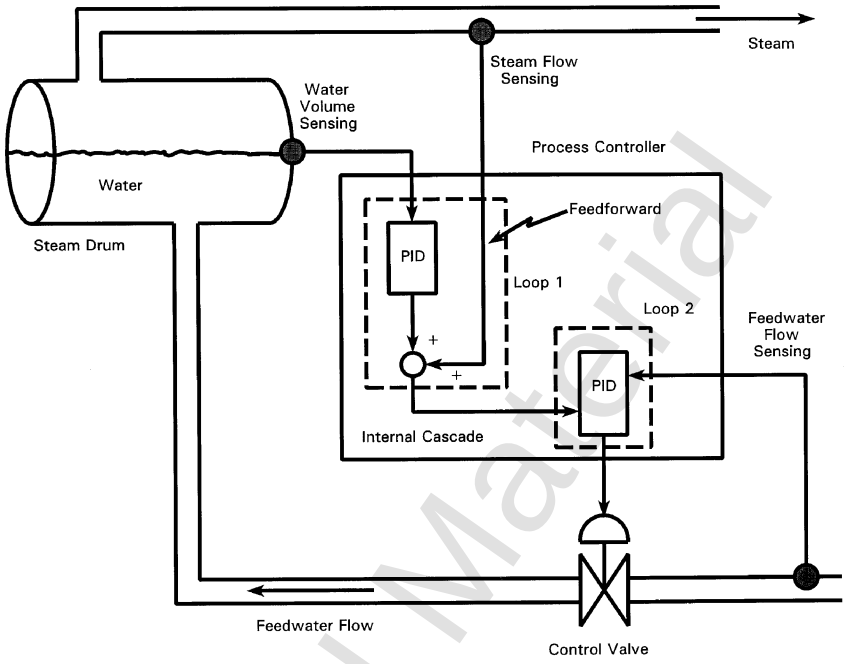


Figure 52 Application of a two-loop process controller for feedwater control.

digital filtering, calculation of control gains, and data analysis. It can accept model descriptions in the form of transfer functions or as state-variable equations.[1,4,10]

Some manufacturers provide software to assist the engineer in sizing and selecting components. An example is the *Motion Component Selector* (MCS) sold by Galil Motion Control, Inc. It assists the engineer in computing the load inertia, including the effects of the mechanical drive, and then selects the proper motor and amplifier based on the user's description of the desired motion profile.

Some hardware manufacturers supply software to assist the engineer in selecting control gains and modifying (*tuning*) them to achieve good response. This might require that the system to be controlled be available for experiments prior to installation. Some controllers, such as some Honeywell process controllers, have an autotuning feature that adjusts the gains in real time to improve performance.

Motion programming software supplied with motion controllers was mentioned previously. Some packages, such as Galil's, allow the user to simulate a multiaxis system having more than one motor and to display the resulting trajectory.

Instrumentation configuration software, such as *LabView*®, provides specialized programming languages for interacting with instruments and for creating graphical real-time displays of instrument outputs.

Until recently, development of real-time digital control software involved tedious programming, often in assembly language. Even when implemented in a higher level language, such as Fortran or C, programming real-time control algorithms can be very challenging, partly because of the need to provide adequately for interrupts. Software packages are now available that provide real-time control capability, usually a form of the PID algorithm, that

Copyrighted Material

Copyright © 2006 John Wiley & Sons Retrieved from: www.knovel.com

can be programmed through user-friendly graphical interfaces. Examples include the Galil motion controllers and the add-on modules for Labview and MATLAB.

12.3 Embedded Control Systems and Hardware-in-the Loop Testing

An *embedded control system* is a microprocessor and sensor suite designed to be an integral part of a product. The aerospace and automotive industries have used embedded controllers for some time, but the decreased cost of components now makes embedded controllers feasible for more consumer and biomedical applications.

For example, embedded controllers can greatly increase the performance of orthopedic devices. One model of an artificial leg now uses sensors to measure in real time the walking speed, the knee joint angle, and the loading due to the foot and ankle. These measurements are used by the controller to adjust the hydraulic resistance of a piston to produce a stable, natural, and efficient gait. The controller algorithms are adaptive in that they can be tuned to an individual's characteristics and their settings changed to accommodate different physical activities.

Engines incorporate embedded controllers to improve efficiency. Embedded controllers in new active suspensions use actuators to improve on the performance of traditional passive systems consisting only of springs and dampers. One design phase of such systems is *hardware-in-the-loop testing,* in which the controlled object (the engine or vehicle suspension) is replaced with a real-time simulation of its behavior. This enables the embedded system hardware and software to be tested faster and less expensively than with the physical prototype and perhaps even before the prototype is available.

Simulink®, which is built on top of MATLAB and requires MATLAB to run, is often used to create the simulation model for hardware-in-the-loop testing. Some of the *toolboxes* available for MATLAB, such as the control systems toolbox, the signal-processing toolbox, and the DSP and fixed-point blocksets, are also useful for such applications.

13 SOFTWARE SUPPORT FOR CONTROL SYSTEM DESIGN

Software packages are available for graphical control system design methods and control system simulation. These greatly reduce the tedious manual computation, plotting, and programming formerly required for control system design and simulation.

13.1 Software for Graphical Design Methods

Several software packages are available to support graphical control system design methods. The most popular of these is MATLAB, which has extensive capabilities for generation and interactive analysis of root-locus plots and frequency response plots. Some of these capabilities are discussed in Refs. 1 and 4.

13.2 Software for Control Systems Simulation

It is difficult to obtain closed-form expressions for system response when the model contains *dead time* or nonlinear elements that represent realistic control system behavior. Dead time (also called *transport delay*), rate limiters, and actuator saturation are effects that often occur in real control systems, and simulation is often the only way to analyze their response. Several software packages are available to support system simulation. One of the most popular is Simulink.

Copyrighted Material

Copyright © 2006 John Wiley & Sons Retrieved from: www.knovel.com

Systems having dead-time elements are easily simulated in Simulink. Figure 53 shows a Simulink model for PID control of the plant $53/(3.44s^2 + 2.61s + 1)$, with a dead time between the output of the controller and the plant. The block implementing the dead-time transfer function e^{-Ds} is called the *transport delay* block. When you run this model, you will see the response in the scope block.

In addition to being limited by saturation, some actuators have limits on how fast they can react. This limitation is independent of the time constant of the actuator and might be due to deliberate restrictions placed on the unit by its manufacturer. An example is a flow control valve whose rate of opening and closing is controlled by a *rate limiter*. Simulink has such a block, and it can be used in series with the saturation block to model the valve behavior. Consider the model of the height h of liquid in a tank whose input is a flow rate q_i. For specific parameter values, such a model has the form $H(s)/Q_i(s) = 2/(5s + 1)$. A Simulink model is shown in Figure 54 for a specific PI controller whose gains are $K_P = 4$ and $K_I = 5/4$. The saturation block models the fact that the valve opening must be between 0 and 100%. The model enables us to experiment with the lower and upper limits of the rate limiter block to see its effect on the system performance.

An introduction to Simulink is given in Refs. 4 and 10. Applications of Simulink to control system simulation are given in Ref. 4.

14 FUTURE TRENDS IN CONTROL SYSTEMS

Microprocessors have rejuvenated the development of controllers for mechanical systems. Currently, there are several applications areas in which new control systems are indispensable to the product's success:

1. Active vibration control
2. Noise cancellation
3. Adaptive optics
4. Robotics
5. Micromachines
6. Precision engineering

Most of the design techniques presented here comprise "classical" control methods. These methods are widely used because when they are combined with some testing and computer simulation, an experienced engineer can rapidly achieve an acceptable design. Modern control algorithms, such as state-variable feedback and the linear–quadratic optimal controller, have had some significant mechanical engineering applications—for example, in the control of aerospace vehicles. The current approach to multivariable systems like the one shown in Fig. 55 is to use classical methods to design a controller for each subsystem because

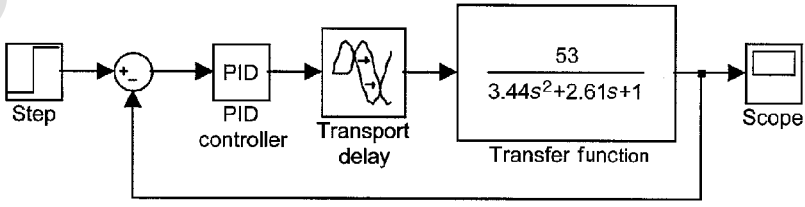


Figure 53 Simulink model of a system with transport delay.

Copyrighted Material

Copyright © 2006 John Wiley & Sons Retrieved from: www.knovel.com

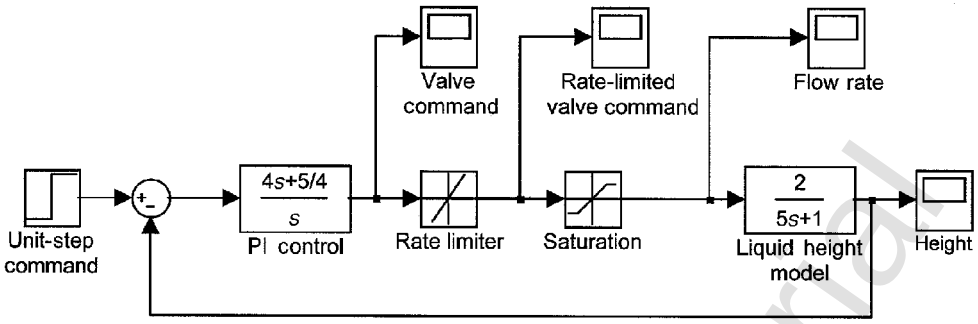


Figure 54 Simulink model of a system with actuator saturation and a rate limiter.

they can often be modeled with low-order linearized models. The coordination of the various low-level controllers is a nonlinear problem. High-order, nonlinear, multivariable systems that cannot be controlled with classical methods cannot yet be handled by modern control theory in a general way, and further research is needed.

In addition to the improvements, such as lower cost, brought on by digital hardware, microprocessors have allowed designers to incorporate algorithms of much greater complexity into control systems. The following is a summary of the areas currently receiving much attention in the control systems community.

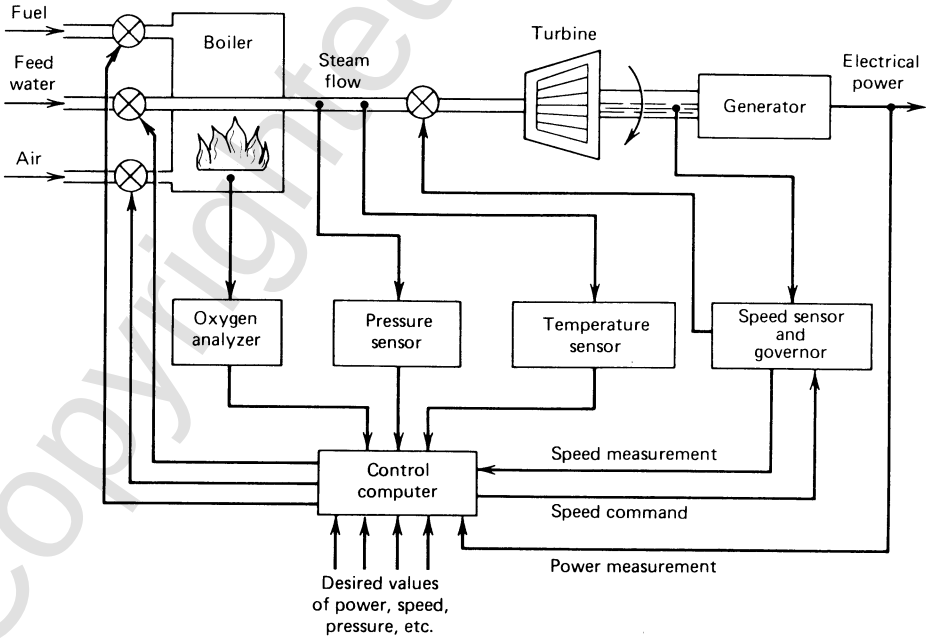


Figure 55 Computer control system for a boiler-generator. Each important variable requires its own controller. The interaction between variables calls for coordinated control of all loops.[1]

Copyrighted Material

Copyright © 2006 John Wiley & Sons Retrieved from: www.knovel.com

14.1 Fuzzy Logic Control

In classical set theory, an object's membership in a set is clearly defined and unambiguous. *Fuzzy logic control* is based on a generalization of classical set theory to allow objects to belong to several sets with various degrees of membership. Fuzzy logic can be used to describe processes that defy precise definition or precise measurement, and thus it can be used to model the inexact and subjective aspects of human reasoning. For example, room temperature can be described as cold, cool, just right, warm, or hot. Development of a fuzzy logic temperature controller would require the designer to specify the membership functions that describe "warm" as a function of temperature, and so on. The control logic would then be developed as a linguistic algorithm that models a human operator's decision process (for example, if the room temperature is "cold," then "greatly" increase the heater output; if the temperature is "cool," then increase the heater output "slightly").

Fuzzy logic controllers have been implemented in a number of applications. Proponents of fuzzy logic control point to its ability to convert a human operator's reasoning process into computer code. Its critics argue that because all the controller's fuzzy calculations must eventually reduce to a specific output that must be given to the actuator (e.g., a specific voltage value or a specific valve position), why not be unambiguous from the start, and define a "cool" temperature to be the range between 65° and 68°, for example? Perhaps the proper role of fuzzy logic is at the human operator interface. Research is active in this area, and the issue is not yet settled.[11,12]

14.2 Nonlinear Control

Most real systems are nonlinear, which means that they must be described by nonlinear differential equations. Control systems designed with the linear control theory described in this chapter depend on a linearized approximation to the original nonlinear model. This linearization can be explicitly performed, or implicitly made, as when we use the small-angle approximation: $\sin \theta \approx \theta$. This approach has been enormously successful because a well-designed controller will keep the system in the operating range where the linearization was done, thus preserving the accuracy of the linear model. However, it is difficult to control some systems accurately in this way because their operating range is too large. Robot arms are a good example.[13,14] Their equations of motion are very nonlinear, due primarily to the fact that their inertia varies greatly as their configuration changes.

Nonlinear systems encompass everything that is "not linear," and thus there is no general theory for nonlinear systems. There have been many nonlinear control methods proposed—too many to summarize here.[15] *Lyapunov's stability theory* and Popov's method play a central role in many such schemes. Adaptive control is a subcase of nonlinear control (see below).

The high speeds of modern digital computers now allow us to implement nonlinear control algorithms not possible with earlier hardware. An example is the computed-torque method for controlling robot arms, which was discussed in Section 11 (see Fig. 47).

14.3 Adaptive Control

The term *adaptive control,* which unfortunately has been loosely used, describes control systems that can change the form of the control algorithm or the values of the control gains in real time, as the controller improves its internal model of the process dynamics or in response to unmodeled disturbances.[16] Constant control gains do not provide adequate response for some systems that exhibit large changes in their dynamics over their entire op-

Copyrighted Material

Copyright © 2006 John Wiley & Sons Retrieved from: www.knovel.com

erating range, and some adaptive controllers use several models of the process, each of which is accurate within a certain operating range. The adaptive controller switches between gain settings that are appropriate for each operating range. Adaptive controllers are difficult to design and are prone to instability. Most existing adaptive controllers change only the gain values, not the form of the control algorithm. Many problems remain to be solved before adaptive control theory becomes widely implemented.

14.4 Optimal Control

A rocket might be required to reach orbit using minimum fuel or it might need to reach a given intercept point in minimum time. These are examples of potential applications of *optimal-control theory.* Optimal-control problems often consist of two subproblems. For the rocket example, these subproblems are (1) the determination of the minimum-fuel (or minimum-time) trajectory and the open-loop control outputs (e.g., rocket thrust as a function of time) required to achieve the trajectory and (2) the design of a feedback controller to keep the system near the optimal trajectory.

Many optimal-control problems are nonlinear, and thus no general theory is available. Two classes of problems that have achieved some practical successes are the *bang-bang control* problem, in which the control variable switches between two fixed values (e.g., on and off or open and closed),[6] and the *linear-quadratic-regulator* (LQG), discussed in Section 7, which has proven useful for high-order systems.[1,6]

Closely related to optimal-control theory are methods based on stochastic process theory, including *stochastic control theory,*[17] *estimators, Kalman filters,* and *observers.*[1,6,17]

REFERENCES

1. W. J. Palm III, *Modeling, Analysis, and Control of Dynamic Systems,* 2nd ed., Wiley, New York, 2000.
2. W. J. Palm III, *Control Systems Engineering,* Wiley, New York, 1986.
3. D. E. Seborg, T. F. Edgar, and D. A. Mellichamp, *Process Dynamics and Control,* Wiley, New York, 1989.
4. W. J. Palm III, *System Dynamics,* McGraw-Hill, New York, 2005.
5. D. McCloy and H. Martin, *The Control of Fluid Power,* 2nd ed., Halsted, London, 1980.
6. A. E. Bryson and Y. C. Ho, *Applied Optimal Control,* Blaisdell, Waltham, MA, 1969.
7. F. Lewis, *Optimal Control,* Wiley, New York, 1986.
8. K. J. Astrom and B. Wittenmark, *Computer Controlled Systems,* Prentice-Hall, Englewood Cliffs, NJ, 1984.
9. Y. Dote, *Servo Motor and Motion Control Using Digital Signal Processors,* Prentice-Hall, Englewood Cliffs, NJ, 1990.
10. W. J. Palm III, *Introduction to MATLAB 7 for Engineers,* McGraw-Hill, New York, 2005.
11. G. Klir and B. Yuan, *Fuzzy Sets and Fuzzy Logic,* Prentice-Hall, Englewood Cliffs, NJ, 1995.
12. B. Kosko, *Neural Networks and Fuzzy Systems,* Prentice-Hall, Englewood Cliffs, NJ, 1992.
13. J. Craig, *Introduction to Robotics,* 3rd ed., Addison-Wesley, Reading, MA, 2005.
14. M. W. Spong and M. Vidyasagar, *Robot Dynamics and Control,* Wiley, New York, 1989.
15. J. Slotine and W. Li, *Applied Nonlinear Control,* Prentice-Hall, Englewood Cliffs, NJ, 1991.
16. K. J. Astrom, *Adaptive Control,* Addison-Wesley, Reading, MA, 1989.
17. R. Stengel, *Stochastic Optimal Control,* Wiley, New York, 1986.

Copyrighted Material

Copyright © 2006 John Wiley & Sons Retrieved from: www.knovel.com