1

Introduction

1.1 Notation and Basic Definitions

Throughout this book, we shall represent scalar quantities by italic letters and symbols (a, α), vectors in boldface ($\mathbf{a}$, $\boldsymbol{\alpha}$), and matrices in boldface capitals ($\mathbf{A}$). Unless otherwise mentioned, the axes of a frame are specified by unit vectors in the same case as that of respective axis labels, for example, the axis ox would be represented by the unit vector, $\mathbf{i}$, while OX is given by $\mathbf{I}$. The axes are labeled in order to constitute a right-handed triad (e.g., $\mathbf{i} \times \mathbf{j} = \mathbf{k}$). Components of a vector have the same subscripts as the axis labels, for example, a vector $\mathbf{a}$ resolved in the frame $oxyz$ is written as

$$\mathbf{a} = a_x\mathbf{i} + a_y\mathbf{j} + a_z\mathbf{k}, \tag{1.1}$$

or alternatively as

$$\mathbf{a} = \begin{Bmatrix} a_x \\ a_y \\ a_z \end{Bmatrix}. \tag{1.2}$$

An overdot represents a vector (or matrix) derived by taking the time derivative of the components in a frame of reference, for example,

$$\dot{\mathbf{a}} \doteq \begin{Bmatrix} \frac{\mathrm{d}a_x}{\mathrm{d}t} \\ \frac{\mathrm{d}a_y}{\mathrm{d}t} \\ \frac{\mathrm{d}a_z}{\mathrm{d}t} \end{Bmatrix} = \begin{Bmatrix} \dot{a}_x \\ \dot{a}_y \\ \dot{a}_z \end{Bmatrix}. \tag{1.3}$$

The vector product of two vectors $\mathbf{a}$, $\mathbf{b}$ is often expressed as the matrix product $\mathbf{a} \times \mathbf{b} = \mathbf{S}(\mathbf{a})\mathbf{b}$, where $\mathbf{S}(\mathbf{a})$ is the following skew-symmetric matrix of the components of $\mathbf{a}$:

$$\mathbf{S}(\mathbf{a}) = \begin{pmatrix} 0 & -a_z & a_y \\ a_z & 0 & -a_x \\ -a_y & a_x & 0 \end{pmatrix}. \tag{1.4}$$

Advanced Control of Aircraft, Spacecraft and Rockets, Ashish Tewari.

Table 1.1 Control system variables

Symbol	Variable	Dimension
$\mathbf{u}(t)$	control input vector	$m \times 1$
$\hat{\mathbf{u}}(t)$	optimal control input vector	$m \times 1$
$\mathbf{w}(t)$	measurement noise vector	$l \times 1$
$\mathbf{x}(t)$	state vector	$n \times 1$
$\hat{\mathbf{x}}(t)$	optimal state vector	$n \times 1$
$\mathbf{y}(t)$	output vector	$l \times 1$
$\mathbf{z}(t)$	state vector for augmentation	$q \times 1$
$\boldsymbol{\nu}(t)$	process noise vector	$p \times 1$

The generic term *planet* is used for any celestial body about which the flight is referenced (Earth, Moon, Sun, Jupiter, etc.). The Euclidean (or L_2) norm of a vector, $\mathbf{a} = (a_x, a_y, a_z)^T$, is written as

$$\| \mathbf{a} \| = \sqrt{a_x^2 + a_y^2 + a_z^2}. \tag{1.5}$$

Standard aerospace symbols define relevant flight parameters and variables as and when used. The nomenclature for control system variables is given in Table 1.1. Any departure from this labeling scheme, if necessary, will be noted.

Control is the name given to the general task of achieving a desired result by appropriate adjustments (or manipulations). The object to be controlled (a flight vehicle) is referred to as the *plant*, while the process that exercises the control is called the *controller*. Both the plant and the controller are *systems*, defined as self-contained sets of physical processes under study. A system has variables applied to it externally, called the *input* vector, and produces certain variables internally, called the *output* vector, which can be measured. In modeling a system, one must account for the relationship between the input and output vectors. This relationship generally takes the form of a set of differential and algebraic equations, if the system is governed by known physical laws. A system having known physical laws is said to be *deterministic*, whereas a system with unknown (or partially known) physical laws is called *non-deterministic* or *stochastic*. Every system has certain unwanted external input variables – called *disturbance inputs* – that cannot be modeled physically and are thus treated as stochastic disturbances. The disturbances are generally of two types: (i) *process noise* that can arise either due to unwanted external inputs, or internally due to uncertainty in modeling the system, and (ii) *measurement noise* that results from uncertainty in measuring the output vector. The presence of these external and internal imperfections renders all practical systems stochastic.

The condition, or *state*, of a system at a given time is specified by a set of scalar variables, called *state variables*, or, in vector form, the *state vector*. The vector space spanned by the state vector is called a *state space*. The state of a system is defined as a collection of the smallest number of variables necessary to completely specify the system's evolution in time, in absence of external inputs. The number of state variables required to represent a system is called *order* of the system, because it is equal to the net order of differential equations governing the system.

While the size of the state space (i.e., the order of the system) is unique, any given system can be described by infinitely many alternative state-space representations. For example, a flight vehicle's state can be described by the position, $\mathbf{r}(t)$, velocity, $\mathbf{v}(t)$, angular velocity, $\boldsymbol{\omega}(t)$, and orientation, $\boldsymbol{\xi}(t)$, relative to a frame of reference. Thus, the *state vector* of a flight vehicle's motion is $\mathbf{x}(t) = \{\mathbf{r}(t), \mathbf{v}(t), \boldsymbol{\omega}(t), \boldsymbol{\xi}(t)\}^T$. However, $\mathbf{x}(t)$ can be transformed into any number of different state vectors depending upon the choice of the reference frame.

A system consisting of the plant and the controller is called a *control system*. The controller manipulates the plant through a *control input* vector, which is actually an input vector to the plant but an output of the controller. In physical terms, this output can take the form of either a force or a torque (or both) applied to a flight vehicle. Often, only electrical (or mechanical) signals are generated by the controller through wires (cables, hydraulic lines), which must be converted into physical inputs for the plant by a separate subsystem called an *actuator*. Also, controllers generally require measurement of the output variables of the plant. Whenever a measurement of a variable is involved, it is necessary to model the dynamics of the measurement process as a separate subsystem called a *sensor*. Generally, there are as many sensors and actuators as there are measured scalar variables and scalar control inputs, respectively. The sensors and actuators can be modeled as part of either the plant or the controller. For our purposes, we shall model them as part of the plant.

The design of a control system requires an accurate mathematical model for the plant. A plant is generally modeled by nonlinear differential equations that can be expressed as a set of first-order ordinary differential equations called the *state equations* such as

$$\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}t} = \mathbf{f}[\mathbf{x}(t), \mathbf{u}(t), \boldsymbol{\nu}(t), t], \tag{1.6}$$

where t denotes time, $\mathbf{x}(t)$ is the state vector (of size $n \times 1$), $\mathbf{u}(t)$ is the control input vector ($m \times 1$), and $\boldsymbol{\nu}(t)$ is the process noise vector ($p \times 1$). The dimension n of the state vector is the order of the system. The nonlinear vector functional, $\mathbf{f}(.)$, is assumed to possess partial derivatives with respect to $\mathbf{x}(t)$, $\mathbf{u}(t)$, and $\boldsymbol{\nu}(t)$ in the neighborhood of $\mathbf{x}_\mathrm{d}(t)$ that constitute a special solution of the state equation (called the *nominal trajectory*). The nominal trajectory usually satisfies equation (1.6) for the unforced case, that is, for $\mathbf{u}(t) = \mathbf{0}$, $\boldsymbol{\nu}(t) = \mathbf{0}$:

$$\frac{\mathrm{d}\mathbf{x}_\mathrm{d}}{\mathrm{d}t} = \mathbf{f}[\mathbf{x}_\mathrm{d}(t), \mathbf{0}, \mathbf{0}, t], \quad t_\mathrm{i} \le t \le t_\mathrm{f}, \tag{1.7}$$

where ($t_\mathrm{i} \le t \le t_\mathrm{f}$) is called the *control interval* with initial time, t_i, and final time, t_f.

The output variables of a plant result from either direct or indirect measurements related to the state variables and control inputs through sensors. Certain errors due to sensor imperfections, called *measurement noise*, are invariably introduced in the measurement process. Therefore, the output vector, $\mathbf{y}(t)$, is related to the state vector, $\mathbf{x}(t)$, the control input vector, $\mathbf{u}(t)$, and the measurement noise vector, $\mathbf{w}(t)$, by an *output equation* given by

$$\mathbf{y}(t) = \mathbf{h}[\mathbf{x}(t), \mathbf{u}(t), \mathbf{w}(t), t], \tag{1.8}$$

where $\mathbf{h}(.)$ is a vector functional and $\mathbf{w}(t)$ is generally of the same size as $\mathbf{y}(t)$.

The most common task of a control system is to bring the plant to a desired state in the presence of disturbances, which can be achieved by either an *open-loop* or a *closed-loop* control system. In an open-loop control system, the controller has no knowledge of the actual state of the plant at a given time, and the control is exercised based upon a model of the plant dynamics, as well as an estimate of its state at a previous time instant, called the *initial condition*. Obviously, such a blind application of control can be successful in driving the plant to a desired state if and only if the plant model is exact, and the external disturbances are absent, which is seldom possible in practice. Therefore, a closed-loop control system is the more practical alternative, in which the actual state of the plant is provided to the controller through a *feedback loop*, so that the control input, $\mathbf{u}(t)$, can be appropriately adjusted. In practice, the feedback consists of measurements of an output vector, $\mathbf{y}(t)$, through which an estimate of the plant's state can be obtained by the controller. If the feedback loop is removed, the control system becomes an open-loop type.

1.2 Control Systems

Our principal task in this book is to design and analyze automatic controllers that perform their duties without human intervention. Generally, a control system can be designed for a plant that is *controllable*.

Controllability is a property of the plant whereby it is possible to take the plant from an initial state, $\mathbf{x}_i(t_i)$, to any desired final state, $\mathbf{x}_f(t_f)$, in a finite time, $t_f - t_i$, solely by the application of the control inputs, $\mathbf{u}(t)$, $t_i \leq t \leq t_f$. Controllability of the plant is a sufficient (but not necessary) condition for the ability to design a successful control system, as discussed in Chapter 2.

For achieving a given control task, a controller must obey well-defined mathematical relationships between the plant's state variables and control inputs, called *control laws*. Based upon the nature of the control laws, we can classify control systems into two broad categories: *terminal control* and *tracking control*. A terminal control system aims to change the plant's state from an initial state, $\mathbf{x}(t_i)$, to a *terminal* (or final) state, $\mathbf{x}(t_f)$, in a specified time, t_f, by applying a control input, $\mathbf{u}(t)$, in the fixed control interval, $(t_i \leq t \leq t_f)$. Examples of terminal control include guidance of spacecraft and rockets. The objective of a tracking control system is to maintain the plant's state, $\mathbf{x}(t)$, quite close to a nominal, reference state, $\mathbf{x}_d(t)$, that is available as a solution to the unforced plant state equation (1.7) by the application of the control input, $\mathbf{u}(t)$. Most flight control problems – such as aircraft guidance, orbital control of spacecraft, and attitude control of all aerospace vehicles – fall in this category.

While the design of a terminal controller is typically based on a nonlinear plant (equation (1.6)) and involves iterative solution of a *two-point boundary value problem*, the design of a tracking controller can be carried out by linearizing the plant about a nominal trajectory, $\mathbf{x}_d(t)$, which satisfies equation (1.6).

A tracking control system can be further classified into *state feedback* and *output feedback* systems. While a state feedback system involves measurement and feedback of all the state variables of the plant (which is rarely practical), an output feedback system is based upon measurement and feedback of some output variables that form the plant's output vector, $\mathbf{y}(t)$. The tracking controller continually compares the plant's state, $\mathbf{x}(t)$, with the nominal (desired) state, $\mathbf{x}_d(t)$, and generates a control signal that depends upon the error vector,

$$\mathbf{e}(t) = \mathbf{x}_d(t) - \mathbf{x}(t). \tag{1.9}$$

Clearly, the controller must be able to estimate the plant's state from the measured outputs and any control input vector, $\mathbf{u}(t)$, it applies to the plant. The estimation of plant's state vector from the available outputs and applied inputs is called *observation* (or *state estimation*), and the part of the controller that performs this essential task is called an *observer*. An observer can only be designed for *observable* plants (Appendix A). Since the observation is never exact, one only has the state estimate, $\mathbf{x}_o(t)$, in lieu of the actual state, $\mathbf{x}(t)$. Apart from the observer, the controller has a separate subsystem called the *regulator* for driving the error vector, $\mathbf{e}(t)$, to zero over a reasonable time interval. The regulator is thus the heart of the tracking control system and generates a control input based upon the detected error. Hence, the control input, $\mathbf{u}(t)$, depends upon $\mathbf{e}(t)$. Moreover, $\mathbf{u}(t)$ may also depend explicitly upon the nominal, reference state, $\mathbf{x}_d(t)$, which must be fed forward in order to contribute to the total control input. Therefore, there must be a third subsystem of the tracking controller, called a *feedforward controller*, which generates part of the control input depending upon the desired state.

A schematic block diagram of the tracking control system with an observer is shown in Figure 1.1. Evidently, the controller represents mathematical relationships between the plant's estimated state, $\mathbf{x}_o(t)$, the reference state, $\mathbf{x}_d(t)$, the control input, $\mathbf{u}(t)$, and time, t. Such relationships constitute a *control law*. For example, a *linear* control law can be expressed as follows:

$$\mathbf{u}(t) = \mathbf{K}_d(t)\mathbf{x}_d(t) + \mathbf{K}(t)\left[\mathbf{x}_d(t) - \mathbf{x}_o(t)\right], \tag{1.10}$$

where $\mathbf{K}_d$ is called the *feedforward gain matrix* and $\mathbf{K}$ the *feedback gain matrix*. Both $(\mathbf{K}, \mathbf{K_d})$ could be time-varying. Note that in Figure 1.1 we have adopted the convention of including sensors and actuators into the model for the plant.

Example 1.1 Consider the problem of guiding a missile to intercept a moving aerial target as shown in Figure 1.2. The centers of mass of the missile, o, and the target, T, are instantaneously located at $\mathbf{R}(t)$ and

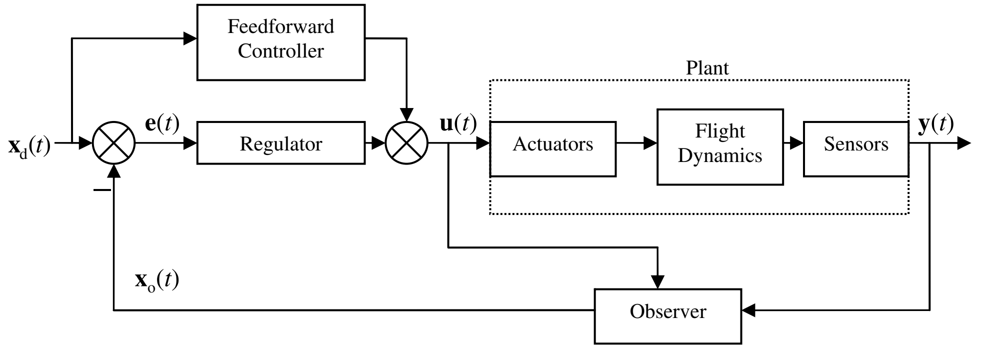

Figure 1.1 Tracking control system with an observer

$\mathbf{R}_T(t)$, respectively, with respective velocities $\mathbf{V}(t)$ and $\mathbf{V}_T(t)$ relative to a stationary frame of reference, $(SXYZ)$. The instantaneous position of the target relative to the missile is given by (the vector triangle Soo' in Figure 1.2)

$$\mathbf{r}(t) = \mathbf{R}_T(t) - \mathbf{R}(t), \tag{1.11}$$

while the target's relative velocity is obtained by differentiation as follows:

$$\mathbf{v}(t) = \frac{d\mathbf{r}}{dt} = \frac{d\mathbf{R}_T}{dt} - \frac{d\mathbf{R}}{dt} = \mathbf{V}_T(t) - \mathbf{V}(t). \tag{1.12}$$

Without considering the equations of motion of the missile and the target (to be derived in Chapter 4), we propose the following control law for missile guidance:

$$\mathbf{V}(t) = \mathbf{K}(t)\,[\mathbf{R}_T(t) - \mathbf{R}(t)] = \mathbf{K}(t)\mathbf{r}(t), \tag{1.13}$$

where $\mathbf{K}(t)$ is a time-varying gain matrix. A linear feedback control law of the form given by equation (1.13) is called a *proportional navigation guidance law* (PNG), whose time derivative gives the required

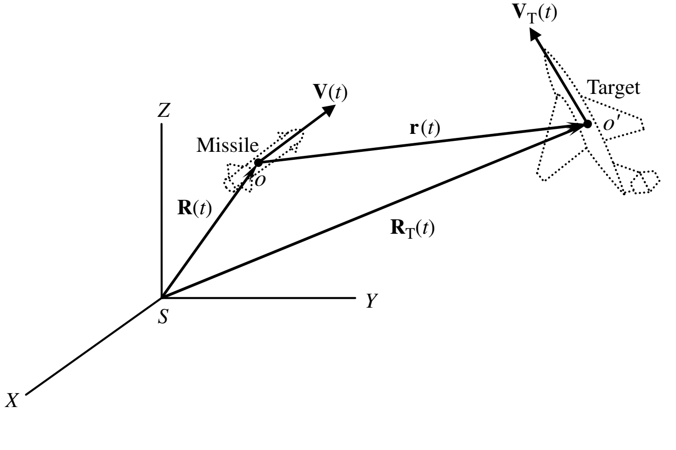

Figure 1.2 Missile guidance for interception of an aerial target

acceleration control input, $\mathbf{u}(t)$, to be applied to the missile:

$$\begin{aligned}\mathbf{u}(t) &= \frac{\mathrm{d}\mathbf{V}}{\mathrm{d}t} \\ &= \mathbf{K}(t)\left[\mathbf{V}_\mathrm{T}(t) - \mathbf{V}(t)\right] + \frac{\mathrm{d}\mathbf{K}}{\mathrm{d}t}\mathbf{r} \\ &= \mathbf{K}\mathbf{v}(t) + \dot{\mathbf{K}}\mathbf{r}(t).\end{aligned} \tag{1.14}$$

For a successful interception of the target, the relative separation, $\mathbf{r}$, must vanish at some time, $T = t_\mathrm{f} - t$, without any regard to the relative velocity, $\mathbf{v}$, prevailing at the time.

A likely choice of state vector for the interception problem is

$$\mathbf{x}(t) = [\mathbf{r}(t), \mathbf{v}(t)]^T, \tag{1.15}$$

which yields the following linear feedback control law of the tracking system:

$$\mathbf{u}(t) = [\dot{\mathbf{K}}(t), \mathbf{K}(t)]\ \mathbf{x}(t). \tag{1.16}$$

The main advantage of the control law given by equation (1.14) is the linear relationship it provides between the required input, $\mathbf{u}$, and the measured outputs, $(\mathbf{r}, \mathbf{v})$, even though the actual plant may have a nonlinear character. Thus the PNG control law is quite simple to implement, and nearly all practical air-to-air missiles are guided by PNG control laws. As the missile is usually rocket powered, its thrust during the engagement is nearly constant. In such a case, PNG largely involves a rotation of the missile's velocity vector through linear feedback relationships between the required normal acceleration components (that are generated by moving aerodynamic fins and/or by thrust vectoring) and the measured relative coordinates and velocity components of the target (obtained by a radar or an infrared sensor mounted on the missile). The proportional navigation gain matrix, $\mathbf{K}(t)$, must be chosen such that the interception $[\mathbf{r}(t_\mathrm{f}) \to \mathbf{0}]$ for the largest likely initial relative distance, $\|\ \mathbf{r}(0)\ \|$, takes place within the allowable maximum acceleration, $\|\ \mathbf{u}\ \| \le u_m$ as well as the maximum time of operation, t_f, of the engines powering the missile. We shall have occasion to discuss the PNG law a little later.

A tracking system with a time-varying reference state, $\mathbf{x}_\mathrm{d}(t)$, can be termed successful only if it can maintain the plant's state, $\mathbf{x}(t)$, within a specified percentage error, $\|\ \mathbf{x}_\mathrm{d}(t) - \mathbf{x}(t)\ \| \le \delta$, of the desired reference state at all times. The achieved error tolerance (or corridor about the reference state), δ, thus gives a measure of the control system's performance. The control system's performance is additionally judged by the time taken by the plant's state to reach the desired error tolerance about the reference state, as well as the magnitude of the control inputs required in the process. The behavior of the closed-loop system is divided into the response at large times, $t \to \infty$, called the *steady-state response*, and that at small values of time when large deviations (called *overshoots*) from the desired state could occur. A successful control system is one in which the maximum overshoot is small, and the time taken to reach within a small percentage of the desired state is also reasonably small.

Example 1.2 Consider a third-order tracking system with the state vector $\mathbf{x} = (x_1, x_2, x_3)^T$. A plot of the nominal trajectory, $\mathbf{x}_\mathrm{d}(t)$ is shown in Figure 1.3. The tracking error corridor is defined by the Euclidean norm of the off-nominal state deviation as follows:

$$\|\ \mathbf{x}_\mathrm{d}(t) - \mathbf{x}(t)\ \| = \sqrt{(x_1 - x_{1\mathrm{d}})^2 + (x_2 - x_{2\mathrm{d}})^2 + (x_3 - x_{3\mathrm{d}})^2} \le \delta, \tag{1.17}$$

where δ is the allowable error tolerance. The actual trajectory is depicted in Figure 1.3 and the maximum overshoot from the nominal is also indicated.

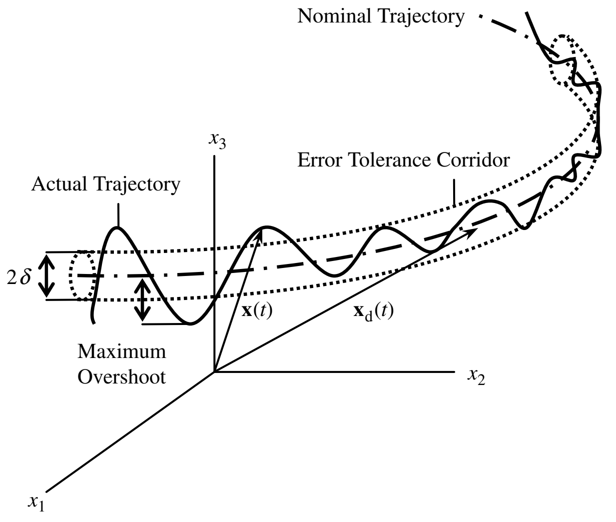

Figure 1.3 Nominal trajectory and tracking error corridor with tolerance, δ

1.2.1 Linear Tracking Systems

With the definition for a successful control system given above, one can usually approximate a nonlinear tracking system by linear differential equations resulting from the assumption of small deviations from the reference state. A first-order Taylor series expansion of the control system's governing (nonlinear) differential equations about the reference state thus yields a linear tracking system (Appendix A), and the given reference solution, $\mathbf{x}_d(t)$, is regarded as the nominal state of the resulting linear system. A great simplification occurs by making such an assumption, because we can apply the principle of *linear superposition* to a linearized system, in order to yield the total output due to a linear combination of several input vectors. Linear superposition also enables us to utilize operational calculus (such as Laplace and Fourier transforms) and linear algebraic methods for design and analysis of control systems. Appendix A briefly presents the linear systems theory, which can be found in detail in any textbook on linear systems, such as Kailath (1980).

Let the control system without disturbance variables be described by the state equation

$$\dot{\boldsymbol{\xi}} = \mathbf{f}[\boldsymbol{\xi}(t), \boldsymbol{\eta}(t), t], \tag{1.18}$$

where $\boldsymbol{\xi}$ is the state vector, and $\boldsymbol{\eta}$, the input vector. The nonlinear vector functional, $\mathbf{f}(.)$, is assumed to possess partial derivatives with respect to state and input variables in the neighborhood of the reference, nominal trajectory, $\boldsymbol{\xi}_0(t)$, which is a solution to equation (1.18) and thus satisfies

$$\dot{\boldsymbol{\xi}}_0(t) = \mathbf{f}[\boldsymbol{\xi}_0(t), \boldsymbol{\eta}_0(t), t], \quad t_i \le t \le t_f, \tag{1.19}$$

where $\boldsymbol{\eta}_0(t)$ is the known input (called the *nominal input*) applied to the system in the interval ($t_i \le t \le t_f$).

In order to maintain the system's state close to a given reference trajectory, the tracking system must possess a special property, namely *stability* about the nominal reference trajectory. While stability can be defined in various ways, for our purposes we will consider *stability in the sense of Lyapunov* (Appendix B), which essentially implies that a small control perturbation from the nominal input results in only a small deviation from the nominal trajectory.

In a tracking system, the system is driven close to the nominal trajectory by the application of the *control input*, $\mathbf{u}(t)$, defined as the difference between the actual and the nominal input vectors:

$$\mathbf{u}(t) = \boldsymbol{\eta}(t) - \boldsymbol{\eta}_0(t), \quad t_\mathrm{i} \le t \le t_\mathrm{f}, \tag{1.20}$$

such that the state deviation, $\mathbf{x}(t)$, from the nominal trajectory, given by

$$\mathbf{x}(t) = \boldsymbol{\xi}(t) - \boldsymbol{\xi}_0(t), \quad t_\mathrm{i} \le t \le t_\mathrm{f}, \tag{1.21}$$

remains small. The assumption of a small control input causing a small state perturbation (which results from the stability about the reference trajectory) is crucial to a successful control system design, and leads to the following Taylor series expansion around the nominal trajectory:

$$\begin{aligned}\dot{\mathbf{x}}(t) = &\tfrac{\partial \mathbf{f}}{\partial \boldsymbol{\xi}}[\boldsymbol{\xi}_0(t), \boldsymbol{\eta}_0(t), t]\mathbf{x}(t) \\ &+ \tfrac{\partial \mathbf{f}}{\partial \boldsymbol{\eta}}[\boldsymbol{\xi}_0(t), \boldsymbol{\eta}_0(t), t]\mathbf{u}(t) + \mathcal{O}(2), \quad t_\mathrm{i} \le t \le t_\mathrm{f},\end{aligned} \tag{1.22}$$

where $\mathcal{O}(2)$ denotes the second- and higher-order terms involving control and state deviations that are neglected due to the small perturbation (stability) assumption.

The Jacobian matrices (Appendix A) of $\mathbf{f}$ with respect to $\boldsymbol{\xi}$ and $\boldsymbol{\eta}$ in the neighborhood of the nominal trajectory are denoted by

$$\mathbf{A}(t) = \frac{\partial \mathbf{f}}{\partial \boldsymbol{\xi}}[\boldsymbol{\xi}_0(t), \boldsymbol{\eta}_0(t), t], \quad \mathbf{B}(t) = \frac{\partial \mathbf{f}}{\partial \boldsymbol{\eta}}[\boldsymbol{\xi}_0(t), \boldsymbol{\eta}_0(t), t]. \tag{1.23}$$

Retaining only the linear terms in equation (1.22), we have the following state-space description of the system as a set of first-order, linear, ordinary differential equations:

$$\dot{\mathbf{x}}(t) = \mathbf{A}(t)\mathbf{x}(t) + \mathbf{B}(t)\mathbf{u}(t), \quad t_\mathrm{i} \le t \le t_\mathrm{f}, \tag{1.24}$$

with the initial condition $\mathbf{x}_\mathrm{i} = \mathbf{x}(t_\mathrm{i})$. Often, the system's governing equations are linearized *before* expressing them as a set of first-order, nonlinear, state equations (equation (1.19)) leading to the same result as equation (1.24). For the time being, we are ignoring the disturbance inputs to the system, which can be easily included through an additional term on the right-hand side.

When the applied control input is zero ($\mathbf{u}(t) = \mathbf{0}$), equation (1.24) becomes the following *homogeneous* equation:

$$\dot{\mathbf{x}}(t) = \mathbf{A}(t)\mathbf{x}(t). \tag{1.25}$$

The solution to the homogeneous state equation, with the initial condition $\mathbf{x}_\mathrm{i} = \mathbf{x}(t_\mathrm{i})$ is expressed as

$$\mathbf{x}(t) = \boldsymbol{\Phi}(t, t_\mathrm{i})\mathbf{x}(t_\mathrm{i}), \quad \text{for all } t. \tag{1.26}$$

where $\boldsymbol{\Phi}(t, t_\mathrm{i})$ is called the *state transition matrix*. The state transition matrix thus has the important property of transforming the state at time t_i to another time t. Other important properties of the state transition matrix are given in Appendix A. Clearly, stability of the system about the nominal trajectory can be stated in terms of the initial response of the homogeneous system perturbed from the nominal state at some time t_i (Appendix B).

The general solution of nonhomogeneous state equation (1.24), subject to initial condition, $\mathbf{x}_\mathrm{i} = \mathbf{x}(t_\mathrm{i})$, can be written in terms of $\boldsymbol{\Phi}(t, t_\mathrm{i})$ as follows:

$$\mathbf{x}(t) = \boldsymbol{\Phi}(t, t_\mathrm{i})\mathbf{x}_\mathrm{i} + \int_{t_\mathrm{i}}^{t} \boldsymbol{\Phi}(t, \tau)\mathbf{B}(\tau)\mathbf{u}(\tau)\mathrm{d}\tau, \quad \text{for all } t. \tag{1.27}$$

The derivation of the state transition matrix is a formidable task for a linear system with time-varying coefficient matrices (called a *time-varying linear system*). Only in some special cases can the exact closed-form expressions for $\mathbf{\Phi}(t, t_i)$ be derived. Whenever $\mathbf{\Phi}(t, t_i)$ cannot be obtained in closed form, it is necessary to apply approximate numerical techniques for the solution of the state equation.

The output (or *response*) of the linearized tracking system can be expressed as

$$\mathbf{y}(t) = \mathbf{C}(t)\mathbf{x}(t) + \mathbf{D}(t)\mathbf{u}(t), \tag{1.28}$$

where $\mathbf{C}(t)$ is called the *output coefficient matrix*, and $\mathbf{D}(t)$ is called the *direct transmission matrix*. If $\mathbf{D}(t) = \mathbf{0}$, there is no direct connection from the input to the output and the system is said to be *strictly proper*.

1.2.2 Linear Time-Invariant Tracking Systems

In many flight control applications, the coefficient matrices of the plant linearized about a reference trajectory (equations (1.24) and (1.28)) are nearly constant with time. This may happen because the time scale of the deviations from the reference trajectory is too small compared to the time scale of reference dynamics. Examples of these include orbital maneuvers and attitude dynamics of spacecraft about a circular orbit, and small deviations of an aircraft's flight path and attitude from a straight and level, equilibrium flight condition. In such cases, the tracking system is approximated as a *linear time-invariant system*, with **A**, **B** treated as constant matrices. The state transition matrix of a linear time-invariant (LTI) system is written as

$$\mathbf{\Phi}(t, t_i) = e^{\mathbf{A}(t-t_i)}, \tag{1.29}$$

where $e^{\mathbf{A}(t-t_i)}$ is called the *matrix exponential* of the square matrix, $\mathbf{A}(t - t_i)$, and can be calculated by either Laplace transforms or linear algebraic numerical methods (Appendix A). By substituting equation (1.29) into equation (1.27) with $t_i = 0$, we can write the following general expression for the state of an LTI system in the presence of an arbitrary, Laplace transformable input, which starts acting at time $t = 0$ when the system's state is $\mathbf{x}(0) = \mathbf{x}_0$:

$$\mathbf{x}(t) = e^{\mathbf{A}t}\mathbf{x}_0 + \int_0^t e^{\mathbf{A}(t-\tau)}\mathbf{B}(\tau)\mathbf{u}(\tau)\mathrm{d}\tau. \tag{1.30}$$

Since a linear feedback control system can often be designed for an LTI plant using either the traditional transfer function approach (see Chapter 2 of Tewari 2002), or the multi-variable state-space approach (Chapters 5 and 6 of Tewari 2002), the resulting LTI tracking system can be considered a basic form of all flight control systems. For an LTI plant with state equation

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}, \tag{1.31}$$

and output (or measurement) equation

$$\mathbf{y} = \mathbf{C}\mathbf{x} + \mathbf{D}\mathbf{u}, \tag{1.32}$$

a typical linear feedback law for tracking a reference trajectory, $\mathbf{x}(t) = \mathbf{0}$, with a constant regulator gain, **K**, is

$$\mathbf{u} = -\mathbf{K}\mathbf{x}_o, \tag{1.33}$$

where $\mathbf{x}_o(t)$ is the estimated state deviation computed by a linear observer (or Kalman filter) with the observer state, $\mathbf{z}(t)$, observer gain, **L**, and the following observer (or filter) dynamics:

$$\mathbf{x}_o = \mathbf{L}\mathbf{y} + \mathbf{z}, \tag{1.34}$$

$$\dot{\mathbf{z}} = \mathbf{F}\mathbf{z} + \mathbf{G}\mathbf{y} + \mathbf{H}\mathbf{u}. \tag{1.35}$$

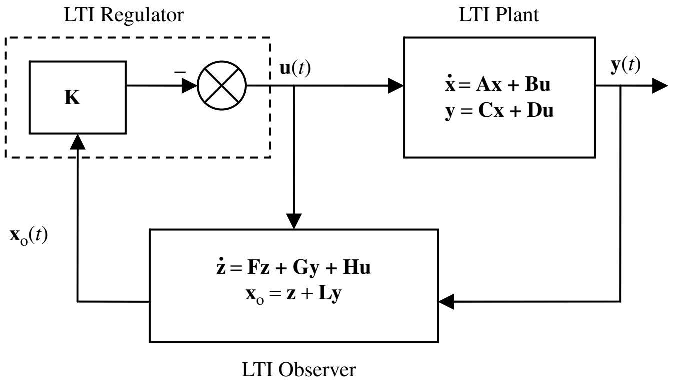

Figure 1.4 Linear time-invariant tracking system

The regulator gain matrix, **K**, the observer gain matrix, **L**, and the constant coefficient matrices of the observer, **F**, **G**, **H**, must satisfy the asymptotic stability criteria for the overall closed-loop system, which translates into separately guaranteeing asymptotic stability of the regulator and observer by the *separation principle* (Kailath 1980). Furthermore, in order to be practical, the design of both the regulator and the observer must be *robust* with respect to modeling uncertainties (process noise) and random external disturbances (measurement noise), both of which are often small in magnitude (thereby not causing a departure from the linear assumption), but occur in a wide spectrum of frequencies. A block diagram of an LTI tracking system with observer is shown in Figure 1.4. The design of such systems for achieving stability and robustness with respect to uncertainties and random disturbances is the topic of several control systems textbooks such as Kwakernaak and Sivan (1972), Maciejowski (1989), Skogestad and Postlethwaite (1996), and Tewari (2002). Examples of the LTI design methods for achieving robustness are *linear quadratic Gaussian* (LQG) and H_∞ techniques that are now standard in modern control design and involve the assumption of a zero-mean *Gaussian white noise*[1] process and measurement noise. A linear observer ensuring the maximum robustness in the state estimation process with respect to a zero-mean Gaussian white noise and measurement noise is called a *Kalman filter*, or *linear quadratic estimator*, and is a part of the LQG controller. Since LQG control, H_∞ control, and the Kalman filter naturally arise out of the optimal control theory, we shall have occasion to consider them in Chapter 2.

1.3 Guidance and Control of Flight Vehicles

Vehicles capable of sustained motion through air or space are termed *flight vehicles*, and are broadly classified as *aircraft*, *spacecraft*, and *rockets*. Aircraft flight is restricted to the atmosphere, and spacecraft flight to exo-atmospheric space, whereas rockets are equally capable of operating both inside and outside the sensible atmosphere. Of these, aircraft have the lowest operating speeds due to the restriction imposed by aerodynamic *drag* (an atmospheric force opposing the motion and proportional to the square of the speed). Furthermore, the requirement of deriving *lift* (a force normal to the flight direction necessary for balancing the weight) and *thrust* (the force along flight direction to counter drag) from the denser regions of the atmosphere restricts aircraft flight to the smallest altitudes of all flight vehicles. In contrast, spacecraft have the highest speeds and altitudes due to the requirement of generating centripetal acceleration entirely

[1] *White noise* is a statistical process with a flat power spectrum, that is, the signal has the same power at any given frequency. A Gaussian process has a normal (bell-shaped) probability density distribution.

by gravity, while the operating speeds and altitudes of rockets (mostly employed for launching the spacecraft) fall somewhere between those of the aircraft and spacecraft.

All flight vehicles require manipulation (i.e., adjustment or *control*) of position, velocity, and attitude (or orientation) for successful and efficient flight. A transport aircraft navigating between points A and B on the Earth's surface must follow a flight path that ensures the smallest possible fuel consumption in the presence of winds. A fighter aircraft has to maneuver in a way such that the normal acceleration is maximized while maintaining the total energy and without exceeding the structural load limits. A spacecraft launch rocket must achieve the necessary orbital velocity while maintaining a particular plane of flight. A missile rocket has to track a maneuvering target such that an intercept is achieved before running out of propellant. An atmospheric entry vehicle must land at a particular point with a specific terminal energy without exceeding the aero-thermal load limits. In all of these cases, precise control of the vehicle's attitude is required at all times since the aerodynamic forces governing an atmospheric trajectory are very sensitive to the body's orientation relative to the flight direction. Furthermore, in some cases, attitude control alone is crucial for the mission's success. For example, a tumbling (or oscillating) satellite is useless as an observation or communications platform, even though it may be in the desired orbit. Similarly, a fighter (or bomber) aircraft requires a stable attitude for accurate weapons delivery.

Flight vehicle control can be achieved by either a pilot, an automatic controller (or *autopilot*), or both acting in tandem. The process of controlling a flight vehicle is called *flying*. A manned vehicle is flown either manually by the pilot, or automatically by the autopilot that is programmed by the pilot to carry out a required task. Unmanned vehicles can be flown either remotely or by onboard autopilots that receive occasional instructions from the ground. It is thus usual to have some form of human intervention in flying, and rarely do we have a fully automated (or *autonomous*) flight vehicle that selects the task to be performed for a given mission and also performs it. Therefore, the job of designing an automatic control system (or autopilot) is quite simple and consists of: (i) generating the required flight tasks for a given mission that are then stored in the autopilot memory as computer programs or specific data points; and (ii) putting in place a mechanism that closely performs the required (or reference) flight tasks at a given time, despite external disturbances and internal imperfections. In (i), the reference tasks are generally a set of positions, velocities, and attitudes to be followed as functions of time, and can be updated (or modified) by giving appropriate signals by a human being (pilot or ground controller). The result is an automatic control system that continually compares the actual position, velocity, and attitude of the vehicle with the corresponding reference values (i) and makes the necessary corrections (ii) in order that the vehicle moves in the desired manner.

Any flight vehicle must have two separate classes of control systems: first, control of position and linear velocity relative to a planet fixed frame, called trajectory control (or more specifically *guidance* that results in the vehicle's *navigation*[2] from one position to another; and second, control of vehicle's orientation (*attitude control*) with respect to a frame of reference. The desired position and velocity – usually derived from the solution of a trajectory optimization problem – could be stored onboard at discrete times, serving as nominal (reference) values against which the actual position and velocity can be compared. A guidance system continually compares the vehicles's actual position and velocity with the nominal ones, and produces linear acceleration commands in order to correct the errors. Most flight vehicles require reorientation (rotation) is realized in practice. Vehicle rotation is performed by applying an angular acceleration external/internal torques. Thus, in such a case, attitude control system becomes subservient (actuator) to the guidance system. In layman terms, the guidance system can be said to "drive the vehicle on an invisible highway in the sky" by using the attitude control system to twist and turn the vehicle. In many flight conditions, a natural stability is inherent in the attitude dynamics so that

[2] The term *navigation* is sometimes applied specifically to the determination of a vehicle's current position. However, in this book, we shall employ the term in its broader context, namely the process of changing the vehicle's position between two given points.

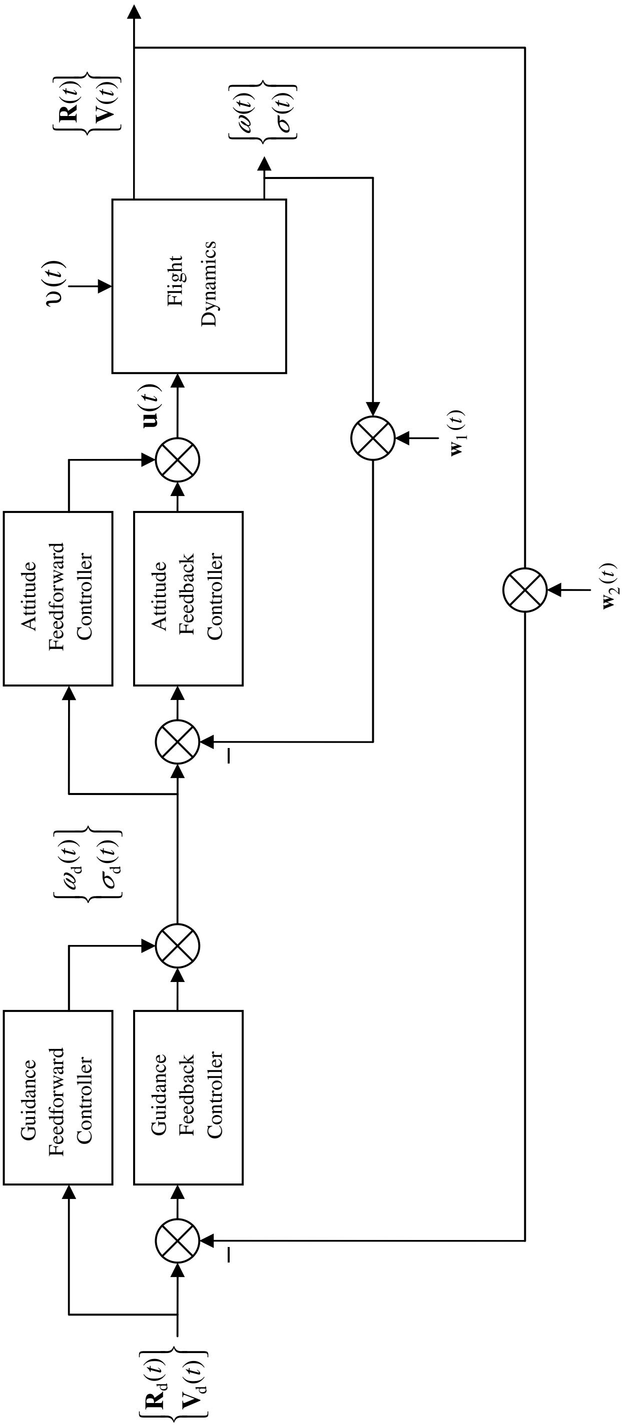

Figure 1.5 Block diagram of a typical flight control system

equilibrium is attained without control effort – that is, the vehicle may, by and large, keep on moving along a desired path on its own. However, even in such cases, it is necessary to augment the stability of the equilibrium point for improved performance. Therefore, we rarely have a flight situation where neither guidance nor attitude control is required.

For most flight vehicles, navigation involves a time scale about an order of magnitude larger than that of attitude dynamics. Hence, navigation along a slowly changing flight path can often be achieved manually (by either a pilot or a ground operator), especially when the time scale of the associated dynamics is larger than a few seconds. When the time scales of control system dynamics are too small to be managed manually, automatic flight control mechanisms become indispensable. Such is the case for all high-performance aircraft, missiles, launch-vehicles, entry vehicles, and some spacecraft. Automatic flight control is also necessary in order to alleviate pilot workload in demanding tasks that require accurate energy and momentum management, and has thus become a common feature of almost all flight vehicles.

A block diagram of a typical flight control system is shown in Figure 1.5. Note the separate feedback loops for guidance and control. The desired position vector, $\mathbf{r}_d(t)$, and the desired velocity vector, $\mathbf{v}_d(t)$ (usually in an inertial reference frame), are fed as a vector input – variously called *setpoint*, *reference signal*, or *desired output* – to the automatic flight control system. In addition, there are disturbance inputs to the control system in the form of process noise, $\boldsymbol{\nu}(t)$, measurement noise of attitude loop, $\mathbf{w}_1(t)$, and measurement noise of the guidance loop, $\mathbf{w}_2(t)$.

1.4 Special Tracking Laws

While the mathematical derivation of control laws for tracking systems generally requires optimal control theory (Chapter 2) as well as a detailed mathematical model of the plant dynamics, we can discuss certain special tracking laws without resorting to either optimal control, or accurate plant modeling. Such control laws are derived intuitively for guidance and attitude control of flight vehicles, and have been successfully applied in practical flight control situations in the past – such as in early tactical and strategic missiles and spacecraft (Battin 1999). They were especially indispensable at a time when the optimal control theory was not well established. Being extremely simple to implement, such intuitive control methods are still in vogue today.

1.4.1 Proportional Navigation Guidance

We briefly discussed proportional navigation guidance in Example 1.2 for guiding a missile to intercept an aerial target. However, PNG is a more general strategy and can be applied to any situation where the nominal trajectory is well known. Thus we consider $\mathbf{r}(t)$ in equation (1.13) to be the instantaneous separation (or position error) from a given nominal trajectory, which must be brought to zero in time $T - t$. The total time of flight, $t_f = t + T$, is generally specified at the outset. The feedback control law for PNG, equation (1.14) – rewritten below for convenience – implies that the corrective acceleration control, $\mathbf{u}(t)$, continues to be applied to the missile until *both* the position error and the velocity error, $\mathbf{v}(t) = \mathbf{V}_T(t) - \mathbf{V}(t)$, become zero:

$$\mathbf{u}(t) = \mathbf{K}\mathbf{v}(t) + \dot{\mathbf{K}}\mathbf{r}(t). \tag{1.36}$$

Thus PNG provides a strategy for intercepting the nominal trajectory with the necessary nominal velocity, $\mathbf{V}_T(t)$, and the velocity error, $\mathbf{v}(t)$, is referred to as the *velocity to be gained*.

Equation (1.13) indicates that the PNG law essentially guides the missile's velocity vector toward a *projected interception point*, P, which is instantaneously obtained at time t by projecting the target's current velocity, $\mathbf{V}_T(t)$, in a straight line as shown in Figure 1.6. Clearly, P must be a function of time as

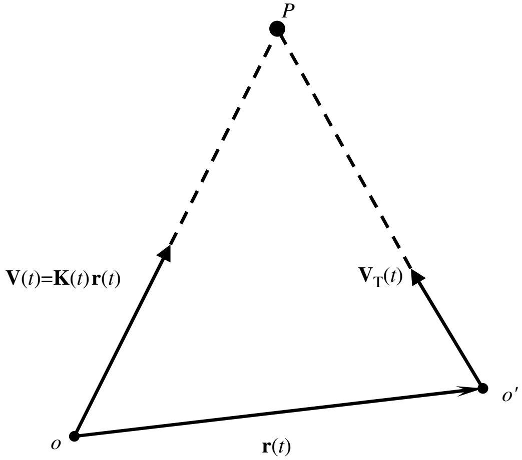

Figure 1.6 Proportional navigation guidance and the projected interception point

neither $\mathbf{r}(t)$ nor $\mathbf{V}_\mathrm{T}(t)$ is constant. Given the velocity triangle shown in Figure 1.6, we have

$$(T - t)\mathbf{K}\mathbf{r} = \mathbf{r} + (T - t)\mathbf{V}_\mathrm{T}, \tag{1.37}$$

where $T = t_\mathrm{f} - t$ is the time of projected interception. At first glance, it would appear that the elements of the PNG gain matrix can be computed from equation (1.15) at each time instant for the current values of the relative separation, $\mathbf{r}$, the target velocity, $\mathbf{V}_\mathrm{T}$, and the desired time before intercept (also called *time to go*), $T - t$. However, the determination of $\mathbf{K}(t)$ from equation (1.15) is not possible as it gives only three scalar equations for the nine unknown elements. Additional conditions should therefore be specified, such as the optimization of certain variables (Chapter 2) for a unique determination of the PNG gains.

Fortunately, it is not necessary to know the time-varying PNG gains explicitly for the implementation of the proportional navigation law. Substituting equation (1.15) into equation (1.36) gives

$$\begin{aligned}\mathbf{u}(t) &= \dot{\mathbf{V}} = \tfrac{\mathrm{d}}{\mathrm{d}t}\left\{\mathbf{K}\mathbf{r}\right\} \\ &= \dot{\mathbf{V}}_\mathrm{T} + \tfrac{\mathbf{r}}{(T-t)^2} + \tfrac{\dot{\mathbf{V}}_\mathrm{T}-\dot{\mathbf{V}}}{T-t},\end{aligned} \tag{1.38}$$

or

$$\mathbf{u}(t) = \dot{\mathbf{V}} = \dot{\mathbf{V}}_\mathrm{T} + \frac{\mathbf{r}}{(T - t)(T - t + 1)}. \tag{1.39}$$

Thus, given the target's velocity vector and the instantaneous separation vector, $\mathbf{r}(t)$, both of which can be either directly measured, or estimated from other measured variables, one can apply a corrective acceleration input to the interceptor according to equation (1.39). It is to be reiterated that there is no need to solve the equations of motion of the interceptor in real time for application of the PNG tracking law.

Example 1.3 Consider a simple example of tracking a projectile flying relatively slowly so that planetary curvature is negligible. Furthermore, the altitude is small enough for acceleration due to gravity, g, to be constant. Such an approximation is termed the *flat planet assumption*, and implies that the acceleration

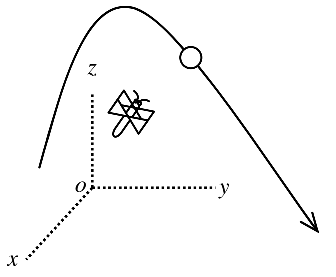

Figure 1.7 Geometry of interception of a ball by a dragonfly

vector due to gravity, $\mathbf{a}_g$, is always vertical. The time of flight is small enough that a frame of reference, $(oxyz)$, fixed to the planetary surface can be considered inertial by neglecting planetary rotation and revolution.

A dragonfly initially at the origin of the frame $(oxyz)$, is trying to alight on a cricket ball that has been hit at an initial velocity, $10\mathbf{i} + 20\mathbf{k}$ m/s, by a batsman at initial coordinates, $(-10, -5, 0.5)$ m (Figure 1.7). Devise a proportional navigation law for the dragonfly such that it makes an in-flight rendezvous with the ball. Neglect atmospheric drag on the ball.

The acceleration of the ball (the target) can be expressed as

$$\ddot{\mathbf{R}}_{\mathrm{T}} = \mathbf{a}_g = -g\mathbf{k}, \tag{1.40}$$

with initial position and velocity vectors, $\mathbf{R}_{\mathrm{T}}(0)$ and $\mathbf{V}_{\mathrm{T}}(0)$, respectively. Since the acceleration is constant, the instantaneous position of the ball is obtained by integration to be

$$\mathbf{R}_{\mathrm{T}}(t) = \mathbf{R}_{\mathrm{T}}(0) + \mathbf{V}_{\mathrm{T}}(0)t - \frac{g}{2}t^2\mathbf{k}. \tag{1.41}$$

The total time of flight, t_{f}, is easily calculated as follows:

$$z_{\mathrm{T}}(t_{\mathrm{f}}) = 0 = z_{\mathrm{T}}(0) + \dot{z}_{\mathrm{T}}(0)t_{\mathrm{f}} - \frac{g}{2}t_{\mathrm{f}}^2, \tag{1.42}$$

or

$$t_{\mathrm{f}} = \frac{\dot{z}_{\mathrm{T}}(0)}{g} + \sqrt{\frac{\dot{z}_{\mathrm{T}}^2(0)}{g^2} + \frac{2z_{\mathrm{T}}(0)}{g}} = 4.1023 \text{ s}.$$

Thus, it is necessary to have interception at a time $T < t_{\mathrm{f}}$. Let us select $T = 30/g$, which yields the following initial velocity for the dragonfly, by virtue of equation (1.15) and the dragonfly's initial position, $\mathbf{R}(0) = \mathbf{0}$:

$$\mathbf{V}(0) = \mathbf{V}_{\mathrm{T}}(0) + \frac{1}{T}\mathbf{R}_{\mathrm{T}}(0) = \begin{pmatrix} 6.7300 \\ -1.6350 \\ 20.1635 \end{pmatrix} \text{ m/s}.$$

The dragonfly must be traveling with this much velocity initially in order to successfully intercept the ball by the PNG approach.

For the simulation of the flight of the dragonfly by a fourth-order Runge–Kutta method, we wrote a MATLAB® program called *run_dragonfly.m* that has the following statements:

```
global g; g=9.81;% m/s^2
global T; T=30/g;% projected intercept time (s)
global RT0; RT0=[-10 -5 0.5]'; % target's initial position (m)
global VT0; VT0=[10 0 20]'; % target's initial velocity (m/s)
V0=VT0+RT0/T % interceptor's initial velocity (m/s)

%Runge-Kutta integration of interceptor's equations of motion:
[t,x]=ode45(@dragonfly,[0 3],[0 0 0 V0']);

% Calculation of instantaneous position error:
RT=RT0*ones(size(t'))+VT0*t'+.5*[0 0 -g]'*(t.^2)';
error=sqrt((x(:,1)-RT(1,:)').^2+(x(:,2)-RT(2,:)').^2+(x(:,3)-RT(3,:)').^2);
```

The equations of motion of the dragonfly are specified in the following file named *dragonfly.m* (which is called by *run_dragonfly.m*):

```
function xdot=dragonfly(t,x)
global g;
global T;
global RT0;
global VT0;
rT=RT0+VT0*t+0.5*[0 0 -g]'*t^2;
r=rT-x(1:3,1);
xdot(1:3,1)=x(4:6,1);
eps=1e-8;
% Avoiding singularity at t=T:
if abs(T-t)>eps
xdot(4:6,1)=[0 0 -g]'+r/((T-t)*(T-t+1));
else
    xdot(4:6,1)=[0 0 -g]';
end
```

In order to avoid the singularity in equation (1.39) at precisely $t = T$, we specify a zero corrective acceleration at that time, so that the dragonfly is freely falling in the small interval, $T - \epsilon < t < T + \epsilon$. The results of the simulation are plotted in Figures 1.8–1.10. Note that due to the non-planar nature of its flight path, the dragonfly is never able to actually sit on the ball, with the smallest position error, $|| \mathbf{r} ||$ (called *miss distance*) being 0.0783 m at $t = 2.295$ s. The miss distance is a characteristic of the PNG approach, which almost never results in a zero error intercept. A relatively small miss distance (such as in the present example) is generally considered to be a successful interception. When applied to a missile, the PNG law is often combined with another approach called *terminal guidance* that takes over near the point of interception in order to further reduce the position error. The maximum speed reached by the dragonfly is its initial speed of 21.3198 m/s, which appears to be within its physical capability of about 80 km/hr.

1.4.2 Cross-Product Steering

It was seen above in the dragonfly example that non-coplanar target and interceptor trajectories caused a near miss instead of a successful rendezvous. A suitable navigational strategy for a rendezvous would then appear to be the one that simultaneously achieves a zero miss distance and a zero relative speed. Intuition suggests that a simple approach for doing so is to turn the velocity vector, $\mathbf{V}(t)$, such that the

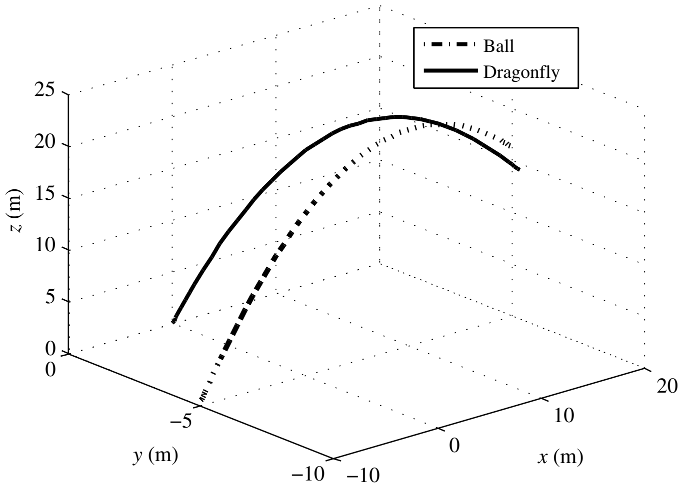

Figure 1.8 Three-dimensional plot of the flight paths taken by the ball and the dragonfly for PNG guidance

instantaneous velocity error (or velocity to be gained), $\mathbf{v}(t) = \mathbf{V}_T(t) - \mathbf{V}(t)$, becomes aligned with the acceleration error,

$$\mathbf{a} = \frac{d\mathbf{v}}{dt} = \frac{d\mathbf{V}_T}{dt} - \frac{d\mathbf{V}}{dt} = \dot{\mathbf{V}}_T - \dot{\mathbf{V}}. \tag{1.43}$$

This implies that the cross-product of velocity and acceleration errors must vanish:

$$\mathbf{v} \times \mathbf{a} = \mathbf{0}. \tag{1.44}$$

Such a navigational law is termed *cross-product steering*.

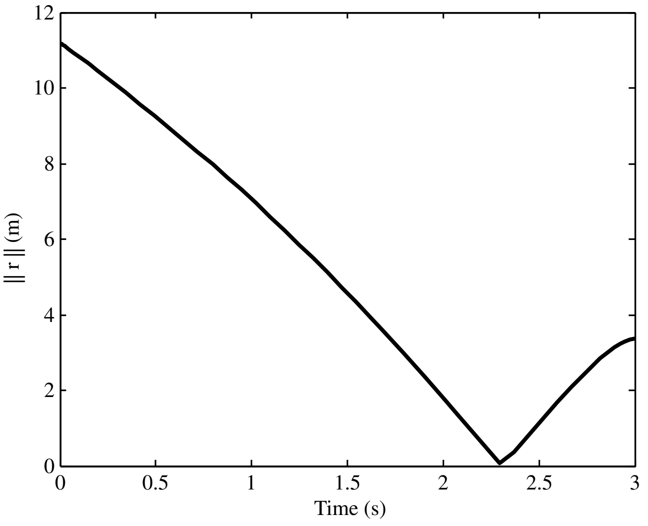

Figure 1.9 Separation error vs. time between the ball and the dragonfly with PNG guidance

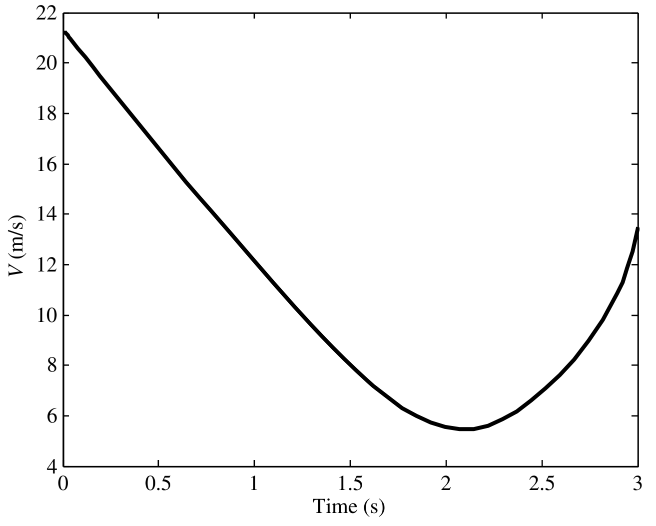

Figure 1.10 Time history of the required speed of the dragonfly for interception by proportional navigation

Since position error is an integral of the velocity error, it is quite possible to make both vanish simultaneously by choosing a constant, c, such that

$$\mathbf{a} = c\mathbf{v}. \tag{1.45}$$

Thus, the required initial velocity of the interceptor is obtained as follows by specifying $\mathbf{r}(T) = \mathbf{0}$ and $\mathbf{v}(T) = \mathbf{0}$:

$$\mathbf{v}(T) - \mathbf{v}(0) = c\,[\mathbf{r}(T) - \mathbf{r}(0)] \tag{1.46}$$

or

$$\mathbf{V}(0) = \mathbf{V}_{\mathrm{T}}(0) - c\mathbf{R}_{\mathrm{T}}(0). \tag{1.47}$$

The value of c must be chosen such that the rendezvous takes place with the desired accuracy within the specified time of flight.

Example 1.4 Let us repeat the ball and dragonfly example (Example 1.3) with cross-product steering instead of PNG navigation. To this end, the statements of the file *dragonfly.m* are modified as follows:

```
function xdot=dragonfly_cps(t,x)
global g;
global RT0;
global VT0;
global c;
vT=VT0+[0 0 -g]'*t;
v=vT-x(4:6,1);
rT=RT0+VT0*t+0.5*[0 0 -g]'*t^2;
r=rT-x(1:3,1);
xdot(1:3,1)=x(4:6,1);
% Cross-product steering (to be applied until
```

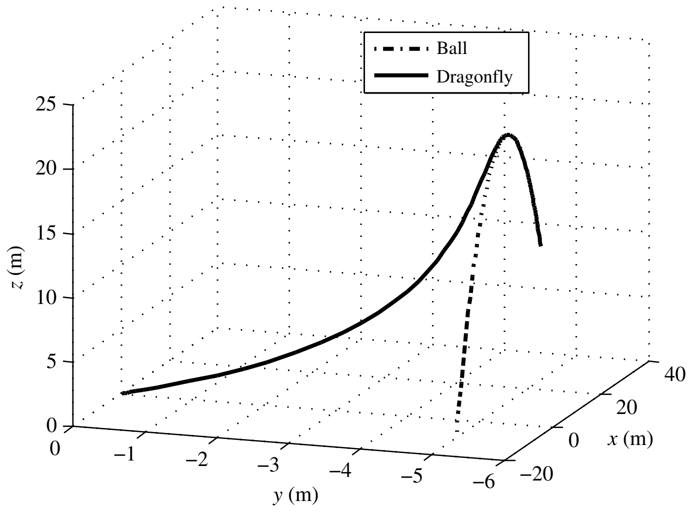

Figure 1.11 Three-dimensional plot of the flight paths taken by the ball and the dragonfly with cross-product steering

```
% a desired positional accuracy is achieved):
if abs(norm(r))>0.0009
xdot(4:6,1)=[0 0 -g]'-c*v;
else
xdot(4:6,1)=[0 0 -g]';
end
```

After some trial and error, $c = -3\ \text{s}^{-1}$ is selected, for which the dragonfly's initial velocity is

$$\mathbf{V}(0) = \mathbf{V}_T(0) - c\mathbf{R}_T(0) = \begin{pmatrix} -20 \\ -15 \\ 21.5 \end{pmatrix} \text{ m/s}.$$

The results of the Runge–Kutta simulation are plotted in Figures 1.11–1.13. The desired rendezvous is achieved in 2.029 s with a zero miss distance (8.35×10^{-6} m) and minimum velocity error 0.0027 m/s. The dragonfly now achieves the same plane of flight as the ball (Figure 1.11) and can alight on the ball at the minimum position error point, after which the velocity error and the positional error remain virtually zero (Figure 1.12). However, the required initial speed of the dragonfly is now increased to 32.97 m/s (Figure 1.13), which appears to be impractical. Therefore, while cross-product steering offers both position and velocity matching (compared with only position matching of PNG), it requires a much larger control input than the PNG navigation law for the same initial errors. A compromise can be achieved by using PNG in the initial phase of the flight, followed by cross-product steering in the terminal phase.

1.4.3 Proportional-Integral-Derivative Control

A large variety of tracking problems for mechanical systems (including flight dynamic systems) involve the simultaneous control of a generalized position vector, $\mathbf{q}(t)$, as well as its time derivative

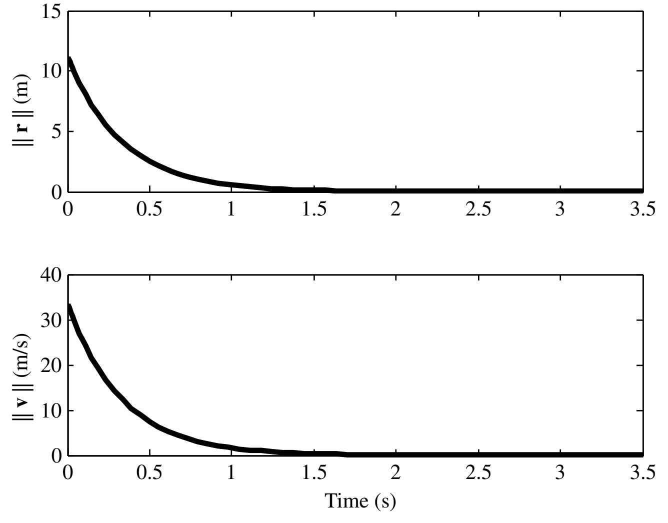

Figure 1.12 Separation and relative speed error vs. time between the ball and the dragonfly with cross-product steering

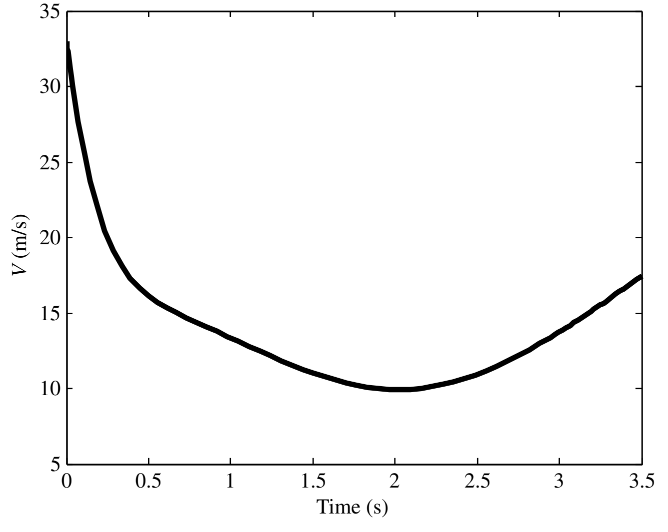

Figure 1.13 Time history of the required speed of the dragonfly for rendezvous by cross-product steering

(generalized velocity), $\dot{\mathbf{q}}(t)$, both of which constitute the nonlinear system's state vector,

$$\xi(t) = \begin{Bmatrix} \mathbf{q}(t) \\ \dot{\mathbf{q}}(t) \end{Bmatrix}. \tag{1.48}$$

For ensuring that the error from a desired (or nominal) generalized trajectory, $\mathbf{q}_d(t)$, defined by

$$\mathbf{e}(t) = \mathbf{q}_d(t) - \mathbf{q}(t), \tag{1.49}$$

is always kept small, a likely feedback control law is the one that applies a corrective control input, $\mathbf{u}(t)$, proportional to the error, the time integral of the error, and its rate as follows:

$$\mathbf{u}(t) = \mathbf{K}_p\mathbf{e}(t) + \mathbf{K}_i \int_0^t \mathbf{e}(\tau)\mathrm{d}\tau + \mathbf{K}_d\dot{\mathbf{e}}(t), \quad t \geq 0, \tag{1.50}$$

where $(\mathbf{K}_p, \mathbf{K}_i, \mathbf{K}_d)$ are (usually constant) gain matrices. In this way, not only can a correction be applied based upon the instantaneous deviation from the trajectory, but also the historically accumulated error as well as its tendency for the future can be corrected through the integral and derivative terms, respectively. Such a control law is termed *proportional-integral-derivative* (PID) control and requires a measurement and feedback of not only the error vector, but also its time integral and time derivative, as shown in the schematic block diagram of Figure 1.14. The process of determination of the gain matrices (often from specific performance objectives) is called *PID tuning*. While the PID control was originally intended for single-variable systems, its application can be extended to multi-variable plants as follows.

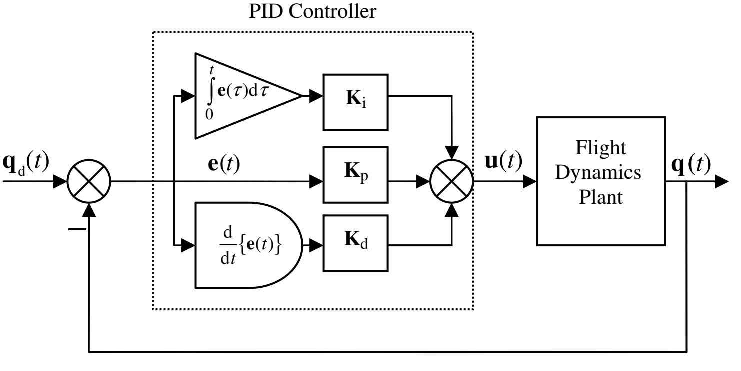

Figure 1.14 Schematic block diagram of PID control system

Since a successful application of PID control would result in a small error vector, the plant's governing equations can be generally linearized about the nominal trajectory resulting in the following governing error equation:

$$\ddot{\mathbf{e}} = \mathbf{A}_1\mathbf{e} + \mathbf{A}_2\dot{\mathbf{e}} + \mathbf{B}\mathbf{u}, \tag{1.51}$$

with the small initial conditions,

$$\mathbf{e}(0) = \mathbf{e}_0, \quad \dot{\mathbf{e}}(0) = \dot{\mathbf{e}}_0. \tag{1.52}$$

Here, $(\mathbf{A}_1, \mathbf{A}_2, \mathbf{B})$ are Jacobian matrices of the linearized plant (Appendix A) and could be varying with time along the nominal trajectory. Substitution of equation (1.50) into equation (1.51) with the error state vector

$$\mathbf{x}(t) = \begin{Bmatrix} \int_0^t \mathbf{e}(\tau)\mathrm{d}\tau \\ \mathbf{e}(t) \\ \dot{\mathbf{e}}(t) \end{Bmatrix} \tag{1.53}$$

results in the following dynamic state equation of the closed-loop system:

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x}, \tag{1.54}$$

where the closed-loop dynamics matrix is given by

$$\mathbf{A} = \begin{bmatrix} \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} \\ \mathbf{B}\mathbf{K}_\mathrm{i} & \mathbf{A}_1 + \mathbf{B}\mathbf{K}_\mathrm{p} & \mathbf{A}_2 + \mathbf{B}\mathbf{K}_\mathrm{d} \end{bmatrix}. \tag{1.55}$$

Clearly, for the error to remain small at all times $t \geq 0$ beginning from a small initial state of

$$\mathbf{x}(0) = \begin{Bmatrix} \mathbf{0} \\ \mathbf{e}_0 \\ \dot{\mathbf{e}}_0 \end{Bmatrix}, \tag{1.56}$$

the closed-loop error dynamics (equation (1.54)) must be stable in the sense of Lyapunov (Appendix B). The PID gain matrices must therefore be selected in a way that ensures stability of the closed-loop system. However, such a derivation of the gain matrices can be termed practical only for a time-invariant system (i.e., when $(\mathbf{A}_1, \mathbf{A}_2, \mathbf{B})$ are constant matrices).

Example 1.5 Consider an axisymmetric, rigid spacecraft of principal moments of inertia, $J_{xx} = J_{yy}$ and J_{zz}, equipped with a rotor that can apply a gyroscopic, internal torque input, $\mathbf{u} = (u_x, u_y)^T$, normal to the axis of symmetry. The spacecraft has the following equations of motion:

$$\mathbf{u} = \mathbf{J}\dot{\boldsymbol{\omega}} + \boldsymbol{\omega} \times \mathbf{J}\dot{\boldsymbol{\omega}}, \tag{1.57}$$

where

$$\mathbf{J} = \begin{pmatrix} J_{xx} & 0 & 0 \\ 0 & J_{xx} & 0 \\ 0 & 0 & J_{zz} \end{pmatrix} \tag{1.58}$$

and

$$\boldsymbol{\omega} = \begin{Bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{Bmatrix}. \tag{1.59}$$

When the torque input is zero, the spacecraft has an equilibrium state with pure spin about the axis of symmetry, $\boldsymbol{\omega}^T = (0, 0, n)$. If an initial disturbance with small, off-axis angular velocity components, $\omega_x(0), \omega_y(0)$, is applied at $t = 0$, the spacecraft has the following LTI dynamics linearized about the equilibrium state:

$$\begin{Bmatrix} \dot{\omega}_x \\ \dot{\omega}_y \end{Bmatrix} = \begin{pmatrix} 0 & -k \\ k & 0 \end{pmatrix} \begin{Bmatrix} \omega_x \\ \omega_y \end{Bmatrix} + \frac{1}{J_{xx}} \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{Bmatrix} u_x \\ u_y \end{Bmatrix}, \tag{1.60}$$

$$\omega_z = n = \text{const.}, \tag{1.61}$$

where

$$k = n\frac{J_{zz} - J_{xx}}{J_{xx}}$$

is real and constant.

It is necessary to devise an active controller that can reduce the off-axis spin error vector, $\boldsymbol{\omega}_{xy} = (\omega_x, \omega_y)^T$, quickly to zero with the help of the feedback control law

$$\mathbf{u} = J_{xx}\left(k_1\boldsymbol{\omega}_{xy} + k_2\dot{\boldsymbol{\omega}}_{xy}\right), \quad t \geq 0, \tag{1.62}$$

where (k_1, k_2) are constant gains. Since the integral term of the PID control law, equation (1.50), is absent here, the resulting feedback law is termed *proportional-derivative* (PD) control. Substituting equation (1.62) into equation (1.60), we have

$$\begin{aligned} J_{xx}\dot{\omega}_x + \omega_y\omega_z(J_{zz} - J_{xx}) &= J_{xx}\left(k_1\omega_x + k_2\dot{\omega}_x\right), \\ J_{xx}\dot{\omega}_y + \omega_x\omega_z(J_{xx} - J_{zz}) &= J_{xx}(k_1\omega_y + k_2\dot{\omega}_y), \\ J_{zz}\dot{\omega}_z &= 0, \end{aligned} \tag{1.63}$$

which implies

$$\frac{\mathrm{d}\omega_{xy}^2}{\mathrm{d}t} = 2k_1\omega_{xy}^2 + k_2\frac{\mathrm{d}\omega_{xy}^2}{\mathrm{d}t}. \tag{1.64}$$

By selecting the state variables of the closed-loop error dynamics as $x_1 = \omega_x$ and $x_2 = \omega_y$, we arrive at the state space form of equation (1.54) with the state dynamics matrix

$$\mathbf{A} = \frac{1}{1 - k_2}\begin{pmatrix} k_1 & -k \\ k & k_1 \end{pmatrix}. \tag{1.65}$$

While the complete solution vector for the off-axis angular velocity components is given by

$$\begin{Bmatrix} \omega_x(t) \\ \omega_y(t) \end{Bmatrix} = e^{\mathbf{A}t}\begin{Bmatrix} \omega_x(0) \\ \omega_y(0) \end{Bmatrix}, \quad t \geq 0, \tag{1.66}$$

the solution for the closed-loop error magnitude, $\omega_{xy} = \sqrt{\omega_x^2 + \omega_y^2}$, is simply obtained as

$$\omega_{xy}(t) = \omega_{xy}(0)e^{\frac{k_1}{1-k_2}t}, \quad t \geq 0. \tag{1.67}$$

For asymptotic stability (Appendix B) we require that

$$\frac{k_1}{1 - k_2} < 0,$$

which can be satisfied by choosing $k_1 < 0$ and $0 < k_2 < 1$. The magnitudes of (k_1, k_2) are limited by actuator torque and sensor sensitivity constraints.

A representative case of $J_{xx} = J_{yy} = 1000$ kg.m^2, $J_{zz} = 3000$ kg.m^2, $n = 0.01$ rad/s, $k_1 = -1$, $k_2 = 0.5$, $\omega_x(0) = -0.001$ rad/s, and $\omega_y(0) = 0.002$ rad/s, computed with the following MATLAB statements:

```
>> k1=-1;k2=.5;n=0.01;k=2*n;
>> A=[k1 -k;k k1]/(1-k2); % CL state dynamics matrix
>> wx0=-0.001;wy0=0.002;
```

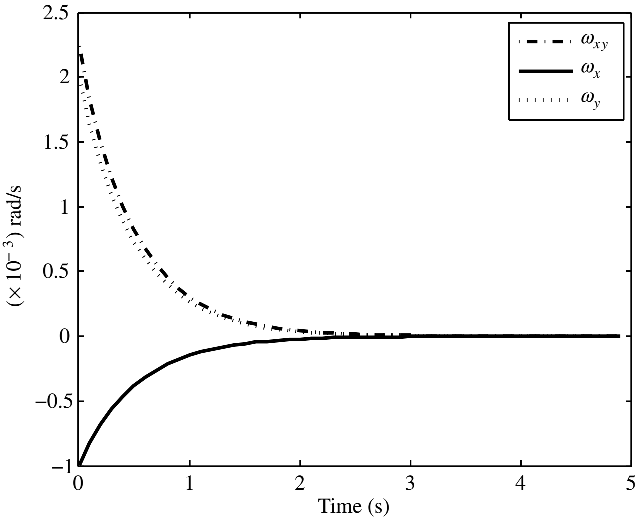

Figure 1.15 Off-axis, closed-loop angular velocity error for an axisymmetric, spin stabilized spacecraft controlled by a rotor that applies a feedback internal torque by the PD scheme

```
>> for i=1:50;
>>   t=(i-1)*0.1;T(:,i)=t;
>>   X(:,i)=sqrt(wx0^2+wy0^2)*exp(k1*t/(1-k2)); %omega_xy magnitude
>>   wxy(:,i)=expm(A*t)*[wx0;wy0]; % Components of omega_xy vector
>> end
```

is plotted in Figure 1.15. Note that the error components decay to almost zero in about 3 s.

1.5 Digital Tracking System

Now we consider how a tracking system is practically implemented. A modern control system drives the actuators through electrical signals as control inputs, while the sensors produce electrical signals as plant outputs. A tracking controller must therefore manipulate the input electrical signals and generate corresponding output electrical signals. While the control system blocks shown in Figure 1.1 represent the input, output, and state variables as continuous functions of time (called *continuous time* or *analog* variables) this is not how an actual control system might work. For example, all modern control systems have at their heart a digital computer, which can process only *discrete time* (or *digital*) signals that are essentially electrical impulses corresponding to the current values (magnitudes and signs) of the time-varying signals. Effectively, this implies that a digital control system can only *sample* time-varying signals at discrete time intervals, called *time steps*. What this means is that, rather than passing prescribed continuous functions of time such as $x(t)$, $y(t)$, one can only send and receive electrical impulses through/from the controller block that correspond to the current values of the variables at a given time instant such as x_k, y_k, where k denotes the kth time instant. Since all modern controllers are digital in character, we require special blocks for converting continuous time signals to digital ones (and vice versa) for communicating with a controller. These essential blocks are called *analog-to-digital* (A/D) and *digital-to-analog* (D/A) *converters* and are shown in Figure 1.16. The A/D and D/A blocks are synchronized by the same clock that sends out an electrical impulse every Δt seconds, which is a fixed duration called the *sampling*

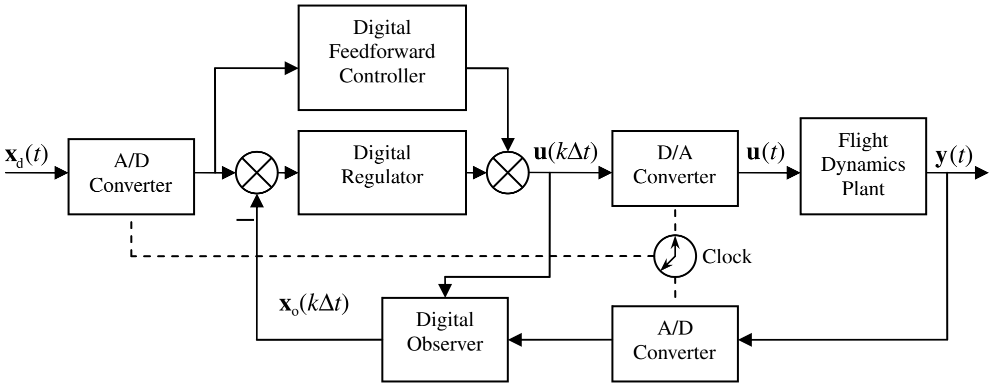

Figure 1.16 Block diagram of a digital flight control system

interval. At the arrival of the clock signal, the A/D block releases its output signal that was *held* at either a constant value (zero-order hold), or an interpolated value of the continuous time input signals received during the sampling interval. In contrast, the D/A block continuously produces its output signal as a weighted average of discrete input signals received (and stored) over the past few sampling intervals.

The digital sampling and holding process involved in A/D conversion mathematically transforms the governing linear differential equations in time into linear algebraic, *difference equations* that describe the evolution of the system's state over a sampling interval. In the frequency domain (Appendix A), the transform method for a discrete time, linear time-invariant system (equivalent to continuous time Laplace transform) is the *z-transform*. For more details about digital control systems, see Chapter 8 of Tewari (2002), or Phillips and Nagle (1995).

A digital control system is unable to respond to signals with frequency greater than the *Nyquist frequency*, defined as half of the sampling frequency and given by $\pi/\Delta t$. Hence, a digital control system has an inherent robustness with respect to high-frequency noise that is always present in any physical system. This primary advantage of a digital control system over an equivalent analog control system is responsible for the replacement of electronic hardware in almost all control applications in favor of digital electronics – ranging from the common digital video recorders to aircraft and spacecraft autopilots.

Since a digital control system is designed and analyzed using techniques equivalent to a continuous time control system, one can easily convert one into the other for a sufficiently small sampling interval, Δt. While many important statistical processes and noise filters are traditionally described as discrete rather than continuous time signals, their continuous time analogs can be easily derived. Therefore, it is unnecessary to discuss both continuous time and discrete time flight control design, and we shall largely restrict ourselves to the former in the remainder of this book.

1.6 Summary

A control system consists of a plant and a controller, along with actuators and sensors that are usually clubbed with the plant. Closed-loop (automatic) control is the only practical alternative in reaching and maintaining a desired state in the presence of disturbances. There are two distinct types of automatic controllers: (a) terminal controllers that take a nonlinear system to a final state in a given time, and (b) tracking controllers which maintain a system close to a nominal, reference trajectory. A tracking controller generally consists of an observer, a regulator, and a feedforward controller. While most plants are governed by nonlinear state equations, they can be linearized about a nominal, reference trajectory, which is a particular solution of the plant's state equations. Linearized state equations are invaluable in designing practical automatic controllers as they allow us to make use of linear systems theory, which easily lends itself to solution by transition matrix as well as providing systematic stability criteria.

All flight vehicles require control of position, velocity, and attitude for a successful mission. Separate control subsystems are required for guiding the vehicle's center of mass along a specific trajectory, and control of vehicle's attitude by rotating it about the center of mass. Due to their greatly different time scales, the guidance system and the attitude control system are usually designed separately and then combined into an overall flight control system. While optimal control techniques are generally required for designing the guidance and attitude control systems, certain intuitive techniques have been successfully applied for practical guidance and control systems in the past, and continue to be in use today due to their simplicity. Examples of intuitive methods include proportional navigation, cross-product steering, and proportional-integral-derivative control.

Many practical control systems are implemented as discrete time (or digital) rather than continuous time (or analog) systems. Due to a bandwidth naturally limited by the sampling rate, a digital control system has an inbuilt robustness with respect to high-frequency noise signals that is not present in analog systems, which has led to the replacement of analog systems by digital systems in all modern control hardware.

Exercises

(1) A system has the following state equations:

$$
\begin{aligned}
\dot{x}_1 &= -2x_2 + u_1, \\
\dot{x}_2 &= x_3 + u_2, \\
\dot{x}_3 &= -3x_1.
\end{aligned}
$$

   (a) If the initial condition at $t = 0$ is $x_1(0) = 10$, $x_2(0) = 5$, $x_3(0) = 2$ and $u(t) = (t, 1)^T$, solve the state equations for the first 10 s.
   (b) Is the system controllable? (See Appendix A.)
   (c) Is the system observable with x_1 and x_2 as outputs? (See Appendix A.)

(2) For a system with the following state-space coefficient matrices:

$$
\mathbf{A} = \begin{pmatrix} -2 & -3 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}, \quad \mathbf{B} = \begin{pmatrix} 1 & 0 \\ 0 & 0 \\ 0 & -1 \end{pmatrix}
$$

   (a) Determine the response to a zero initial condition and a unit impulse function applied at $t = 0$ as the first input.
   (b) What are the eigenvalues of the matrix $\mathbf{A}$? (*Ans.* $1, -1, -2$.)
   (c) Analyze the controllability of the system when only the first input is applied.
   (d) Analyze the controllability of the system when only the second input is applied.

(3) Can the following be the state transition matrix of a homogeneous linear system with state $\mathbf{x}(t) = (-1, 0, -2)^T$ at time $t = 1$?

$$
\mathbf{\Phi}(t, 0) = \begin{pmatrix} 1 - \sin t & 0 & -te^{-2t} \\ 0 & e^{-t}\cos t & 0 \\ -2t & 0 & 1 \end{pmatrix}
$$

Why? (*Ans. No*.)

(4) A homogeneous linear system has the following state transition matrix:

$$\boldsymbol{\Phi}(t, 0) = \begin{pmatrix} \cos t & -\sin t \\ \sin t & \cos t \end{pmatrix}.$$

If the state at time $t = 1$ is $\mathbf{x}(t) = (-1.3818, -0.3012)^T$, what was the state at $t = 0$?

(5) Compute and plot the internal control torque components for the rotor-stabilized spacecraft of Example 1.5. What is the maximum torque magnitude required?

(6) Repeat the simulation of Example 1.3 by adding the following drag deceleration on the ball:

$$\dot{\mathbf{V}}_{\mathrm{T}} = -0.0016(V_{\mathrm{T}}\mathbf{V}_{\mathrm{T}})\ (\mathrm{m/s^2})$$

Is there a change in the miss distance and the maximum speed of the dragonfly?

(7) Repeat the simulation of Example 1.4 with the drag term given in Exercise 4. What (if any) changes are observed?

(8) Consider a controller for the plant given in Exercise 2 based upon proportional-integral-derivative feedback from the first state variable to the second control input:

$$u_2(t) = k_{\mathrm{p}} x_1(t) + k_{\mathrm{i}} \int_0^t x_1(\tau)\mathrm{d}\tau + k_{\mathrm{d}} \dot{x}_1(t).$$

(a) Draw a block diagram of the resulting closed-loop system.
(b) Is it possible to select the PID gains, $k_{\mathrm{p}}, k_{\mathrm{i}}, k_{\mathrm{d}}$, such that closed-loop system has all eigenvalues in the left-half s-plane?

(9) Replace the PID controller of Exercise 8 by a full-state feedback to the second control input given by

$$u_2(t) = -k_1 x_1(t) - k_2 x_2(t) - k_3 x_1(t).$$

(a) Draw a block diagram of the resulting closed-loop system.
(b) Select the regulator gains, k_1, k_2, k_3, such that closed-loop system has all eigenvalues at $s = -1$. (*Ans.* $k_1 = -1/3,\ k_2 = -2,\ k_3 = -1$.)
(c) Determine the resulting control system's response to a zero initial condition and a unit impulse function applied at $t = 0$ as the first input, $u_1(t) = \delta(t)$.

(10) Rather than using the full-state feedback of Exercise 9, it is decided to base the controller on the feedback of the first state variable, $x_1(t)$. In order to do so, a full-order observer (Appendix A) must be separately designed for estimating the state vector from the measurement of $x_1(t)$ as well as the applied input, $u_2(t)$, which is then supplied to the regulator designed in Exercise 9. The state equation of the observer is given by

$$\dot{\mathbf{x}}_{\mathrm{o}} = (\mathbf{A} - \mathbf{LC})\mathbf{x}_{\mathrm{o}} + \mathbf{B}_2 u_2 + \mathbf{L} x_1,$$

where

$$\mathbf{C} = (1,\ 0,\ 0), \quad \mathbf{B_2} = \begin{pmatrix} 0 \\ 0 \\ -1 \end{pmatrix}, \quad \mathbf{L} = \begin{pmatrix} \ell_1 \\ \ell_2 \\ \ell_3 \end{pmatrix}.$$

(a) Select the observer gains, ℓ_1, ℓ_2, ℓ_3, such that the observer dynamics matrix, $(\mathbf{A} - \mathbf{LC})$, has all eigenvalues at $s = -5$. (*Ans.* $\ell_1 = 13, \ \ell_2 = -76/3, \ \ell_3 = -140/3$.)

(b) Close the loop with $u_2(t) = -(k_1, k_2, k_3)\mathbf{x}_o(t)$ and draw the block diagram of the overall closed-loop system with the plant, observer, and regulator.

(c) Determine the resulting control system's response to zero initial condition and a unit impulse function applied at $t = 0$ as the first input. What difference (if any) is seen compared with the response in Exercise 9?

(11) Repeat Exercise 10 after replacing the full-order observer with a reduced-order observer (Appendix A).

References

Battin, R.H. (1999) *An Introduction to the Mathematics and Methods of Astrodynamics*. American Institute of Aeronautics and Astronautics, Reston, VA.

Kailath, T. (1980) *Linear Systems*. Prentice Hall, Englewood Cliffs, NJ.

Kwakernaak, H. and Sivan, R. (1972) *Linear Optimal Control Systems*. John Wiley & Sons, Inc., New York.

Maciejowski, J.M. (1989) *Multivariable Feedback Design*. Addison-Wesley, Reading, MA.

Phillips, C.L. and Nagle, H.T. (1995) *Digital Control Systems Analysis and Design*. Prentice Hall, Englewood Cliffs, NJ.

Skogestad, S. and Postlethwaite, I. (1996) *Multivariable Feedback Design: Analysis and Control*. John Wiley & Sons, Ltd, Chichester.

Tewari, A. (2002) *Modern Control Design with MATLAB and Simulink*. John Wiley & Sons, Ltd, Chichester.