

10

Guidance Systems

This chapter describes methods for the design of *guidance systems* for marine craft (Siouris, 2004, Yanushevsky, 2008). Guidance can be defined as (Shneydor, 1998): “*The process for guiding the path of an object towards a given point, which in general may be moving.*” Draper (1971) states: “*Guidance depends upon fundamental principles and involves devices that are similar for vehicles moving on land, on water, under water, in air, beyond the atmosphere within the gravitational field of Earth and in space outside this field.*” Thus, guidance represents a basic methodology concerned with the transient motion behavior associated with the achievement of motion control objectives (see Breivik and Fossen, 2009).

In its simplest form, open-loop guidance systems for marine craft are used to generate a reference trajectory for time-varying *trajectory tracking* or, alternatively, a path for time-invariant *path following* (see Section 9.2). A motion control system will work in close interaction with the guidance system.

In the control literature, the different motion control scenarios are typically classified according to:

- *Setpoint regulation (point stabilization)* is a special case where the desired position and attitude are chosen to be constant.
- *Trajectory tracking*, where the objective is to force the system output $y(t) \in \mathbb{R}^m$ to track a desired output $y_d(t) \in \mathbb{R}^m$. The desired trajectory can be computed using reference models generated by low-pass filters, optimization methods or by simply simulating the marine craft motion using an adequate model of the craft. Feasible trajectories can be generated in the presence of both *spatial* and *temporal constraints*.
- *Path following* is following a predefined path independent of time. No restrictions are placed on the temporal propagation along the path. Spatial constraints can, however, be added to represent obstacles and other positional constraints if they are known in advance.

Tracking control systems can also be designed for target tracking and path tracking. For instance, a target-tracking system tracks the motion of a target that is either stationary (analogous to point stabilization) or that moves such that only its instantaneous motion is known; that is no information about the future target motion is available (Breivik and Fossen, 2009).

As shown in Figure 10.1, the guidance system can use joystick or keyboard inputs, external inputs (weather data, for instance measured wind, wave and current speeds and directions), Earth topological information (digital chart, radar and sonar data), obstacle and collision avoidance data, and finally the state vector, which is available as output from the navigation and sensor systems. The required data are further processed to generate a feasible trajectory for motion control. This can be done using ad hoc techniques or sophisticated methods such as interpolation techniques, dynamic optimization or filtering

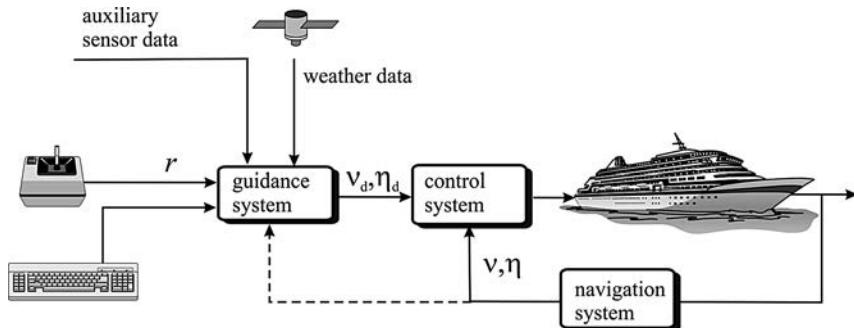


Figure 10.1 In closed-loop guidance (dotted line) the states are fed back to the guidance system while open-loop guidance only uses sensor and reference signal inputs.

techniques. A feasible trajectory means one that is consistent with the craft dynamics. For a linear system, this implies that the eigenvalues of the desired states must be chosen such that the reference model is slower than the craft dynamics.

For a ship or an underwater vehicle, the guidance and control system usually consists of the following subsystems:

- *Attitude control system*
- *Path-following control system*

In its simplest form, the attitude control system is a heading autopilot, while roll and pitch are regulated to zero or left uncontrolled. The main function of the attitude feedback control system is to maintain the craft in a desired attitude on the ordered path by controlling the craft in roll, pitch and yaw. The task of the path-following controller is to keep the craft on the prescribed path with some predefined dynamics, for instance a speed control system by generating orders to the attitude control system. For surface vessels it is common to use a heading controller in combination with a speed controller while aircraft and underwater vehicles also need a height/depth controller. The principles and definitions of *guidance*, *navigation* and *control* are further outlined in Section 9.2.

10.1 Target Tracking

Sometimes no information about the path is known in advance and there is no trajectory to track. Hence, if the goal is to track a moving object, for which no future motion information is available, target-tracking methods can be applied. Guidance laws for target tracking can be used in marine operations such as underway replenishment (UNREP) operations and formation control. UNREP operations involve cargo transfer between two or more cooperating craft in transit. The task of the so-called *guide ship* is to maintain a steady course and speed while the *approach ship* moves up alongside the guide or target ship to receive fuel, munitions and personnel (Skejic *et al.*, 2009).

For surface vessels, the 2-D position of the target is denoted by $\mathbf{p}_t^n = [N_t, E_t]^\top$. The control objective of a target-tracking scenario can be formulated as (Breivik and Fossen, 2009)

$$\lim_{t \rightarrow \infty} [\mathbf{p}_t^n(t) - \mathbf{p}_t^n(t)] = \mathbf{0} \quad (10.1)$$

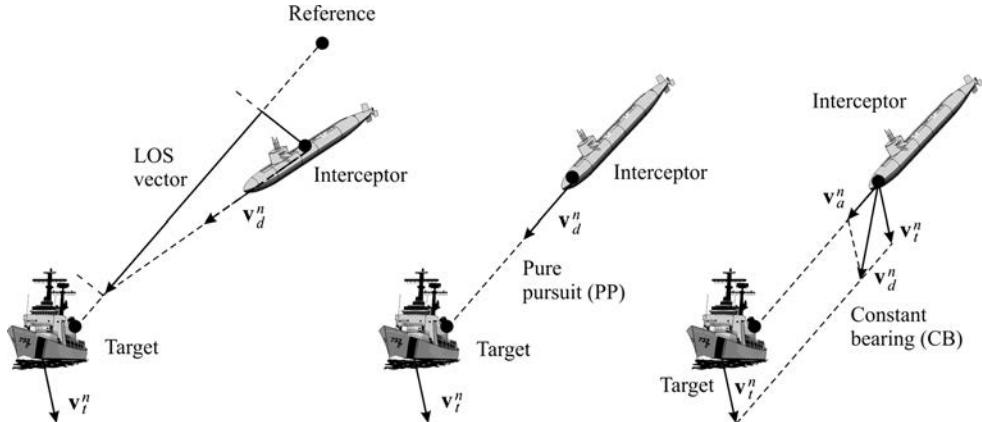


Figure 10.2 Desired interceptor approach speed $U_d = \|v_d^n\|$ for the classical guidance principles: line-of-sight LOS, pure pursuit (PP) and constant bearing (CB). The target speed is $U_t = \|v_t^n\|$.

where $\mathbf{p}^n \in \mathbb{R}^2$ is the craft position. The target velocity is $\mathbf{v}_t^n = \dot{\mathbf{p}}_t^n \in \mathbb{R}^2$. In the missile guidance community an object that is supposed to destroy another object is referred to as a missile, an interceptor or a pursuer. Conversely, the threatened object is typically called a target or an evader. In the following, the designations *interceptor* and *target* will be used.

An interceptor typically undergoes three phases during its operation:

1. Launch phase
2. Midcourse phase
3. Terminal phase

The greatest accuracy demand is associated with the terminal phase, where the interceptor guidance system must compensate for the accumulated errors from the previous phases to achieve a smallest possible final miss distance to the target. The remainder of this section is based on Breivik and Fossen (2009) and Breivik (2010). The discussion is limited to three terminal guidance strategies, *line-of-sight*, *pure pursuit* and *constant bearing*, which are illustrated in Figure 10.2.

Note that while the main objective of a guided missile is to hit and destroy a physical target in finite time, the analogy is to hit or converge to a virtual target asymptotically. This is also referred to as asymptotic interception given by (10.1).

10.1.1 Line-of-Sight Guidance

Line-of-sight (LOS) guidance is classified as a three-point guidance scheme since it involves a typically stationary reference point in addition to the interceptor and the target. The LOS denotation stems from the fact that the interceptor is supposed to achieve an intercept by constraining its motion along the LOS vector between the reference point and the target. LOS guidance has typically been employed for surface-to-air missiles, often mechanized by a ground station, which illuminates the target with a beam that the guided missile is supposed to ride, also known as beam-rider guidance. The LOS guidance principle is illustrated in Figure 10.2, where the interceptor velocity \mathbf{v}_a^n is pointed to LOS vector to obtain the desired velocity \mathbf{v}_d^n . LOS guidance will be applied to track straight-line paths in section 10.3 while curved paths are discussed in Section 10.4.

10.1.2 Pure Pursuit Guidance

Pure pursuit (PP) guidance belongs to the two-point guidance schemes, where only the interceptor and the target are considered in the engagement geometry. The interceptor aligns its velocity \mathbf{v}_a^n along the LOS vector between the interceptor and the target by choosing the desired velocity as

$$\mathbf{v}_d^n = -\kappa \frac{\tilde{\mathbf{p}}^n}{\|\tilde{\mathbf{p}}^n\|} \quad (10.2)$$

where $\kappa > 0$. This strategy is equivalent to a predator chasing a prey in the animal world, and very often results in a tail chase. PP guidance has typically been employed for air-to-surface missiles. The PP guidance principle is represented in Figure 10.2 by a vector pointing directly at the target.

Deviated pursuit guidance is a variant of PP guidance, where the velocity of the interceptor is supposed to lead the interceptor–target line of sight by a constant angle in the direction of the target movement. An equivalent term is fixed-lead navigation.

10.1.3 Constant Bearing Guidance

Constant bearing (CB) guidance is also a two-point guidance scheme, with the same engagement geometry as PP guidance. However, in a CB engagement, the interceptor is supposed to align the interceptor–target velocity \mathbf{v}_a^n along the LOS vector between the interceptor and the target. This goal is equivalent to reducing the LOS rotation rate to zero such that the interceptor perceives the target at a constant bearing, closing in on a direct collision course. CB guidance is often referred to as *parallel navigation* and has typically been employed for air-to-air missiles. Also, the CB rule has been used for centuries by mariners to avoid collisions at sea, steering away from a situation where another craft approaches at a constant bearing. Thus, guidance principles can just as well be applied to avoid collisions as to achieve them. The CB guidance principle is indicated in Figure 10.2.

The most common method of implementing CB guidance is to make the rotation rate of the interceptor velocity directly proportional to the rotation rate of the interceptor–target LOS, which is widely known as *proportional navigation* (PN). However, CB guidance can also be implemented through the direct velocity assignment as proposed by Breivik *et al.* (2006); see Breivik (2010) for details.

The CB desired velocity is given by

$$\mathbf{v}_d^n = \mathbf{v}_t^n + \mathbf{v}_a^n \quad (10.3)$$

$$\mathbf{v}_a^n = -\kappa \frac{\tilde{\mathbf{p}}^n}{\|\tilde{\mathbf{p}}^n\|} \quad (10.4)$$

where $\mathbf{v}_a^n = [\dot{N}_a, \dot{E}_a]^\top$ is the approach velocity vector specified such that the desired approach speed $U_a = \|\mathbf{v}_a^n\|$ is tangential to the LOS vector as shown in Figure 10.2 and

$$\tilde{\mathbf{p}}^n := \mathbf{p}^n - \mathbf{p}_t^n \quad (10.5)$$

is the LOS vector between the interceptor and the target, $\|\tilde{\mathbf{p}}^n\| \geq 0$ is the Euclidean length of this vector and

$$\kappa = U_{a,\max} \frac{\|\tilde{\mathbf{p}}^n\|}{\sqrt{(\tilde{\mathbf{p}}^n)^\top \tilde{\mathbf{p}}^n + \Delta_{\tilde{\mathbf{p}}}^2}} \quad (10.6)$$

where $U_{a,\max}$ specifies the maximum approach speed toward the target and $\Delta_{\tilde{p}} > 0$ affects the transient interceptor-target rendezvous behavior. The CB guidance law (10.3) computes the velocity commands needed to track the target.

Note that CB guidance becomes equal to PP guidance for a stationary target; that is the basic difference between the two guidance schemes is whether the target velocity is used as a kinematic feedforward or not.

Convergence and Stability Analyses

The convergence properties of (10.3)–(10.4) and (10.6) can be investigated by considering a Lyapunov function candidate (LFC):

$$V = \frac{1}{2}(\tilde{\mathbf{p}}^n)^\top \tilde{\mathbf{p}}^n > 0, \quad \forall \tilde{\mathbf{p}}^n \neq \mathbf{0} \quad (10.7)$$

Time differentiation of V along the trajectories of $\tilde{\mathbf{p}}^n$ gives

$$\begin{aligned} \dot{V} &= (\tilde{\mathbf{p}}^n)^\top \tilde{\mathbf{v}}^n \\ &= -\kappa \frac{(\tilde{\mathbf{p}}^n)^\top \tilde{\mathbf{p}}^n}{\|\tilde{\mathbf{p}}^n\|} \\ &= -U_{a,\max} \frac{(\tilde{\mathbf{p}}^n)^\top \tilde{\mathbf{p}}^n}{\sqrt{(\tilde{\mathbf{p}}^n)^\top \tilde{\mathbf{p}}^n + \Delta_{\tilde{p}}^2}} \\ &< 0, \quad \forall \tilde{\mathbf{p}}^n \neq \mathbf{0} \end{aligned} \quad (10.8)$$

The LFC (10.7) is positive definite and radially unbounded, while its derivative with respect to time (10.8) is negative definite when adhering to $U \geq U_{a,\max} > 0$. Hence, by standard Lyapunov arguments the origin $\tilde{\mathbf{p}}^n = \mathbf{0}$ is UGAS (see Appendix A.1). Finally, the Jacobian of the error dynamics $\tilde{\mathbf{p}}^n$ at the origin $\tilde{\mathbf{p}}^n = \mathbf{0}$ has strictly negative eigenvalues, which proves ULES.

Example 10.1 (UNREP Operation)

For ships equipped with a rudder and a main propeller, the rudder can be used to obtain a parallel course with the guide ship (lateral alignment). We will consider a guide ship moving on a straight line and the goal for the second ship is to approach this ship to carry out a UNREP operation. In Skejic et al. (2009) the lookahead-based steering law of Breivik and Fossen (2009) in Section 10.3 is used to ensure that the approach ship is able to assume a parallel course with the guide ship by adhering to the desired course angle

$$\chi_d = \chi_t + \chi_r \quad (10.9)$$

where χ_t is the course angle of the guide ship. The heading of the approach ship is adjusted using the steering law χ_r such that the lateral distance is adjusted as desired. The lateral distance and cross-track error (s, e) are obtained by the following transformation:

$$\begin{bmatrix} s \\ e \end{bmatrix} = \begin{bmatrix} s_d \\ e_d \end{bmatrix} + \mathbf{R}_p(\chi_t)^\top (\mathbf{p}^n - \mathbf{p}_t^n) \quad (10.10)$$

where $s_d = 0$ (interceptor on parallel course), $e_d = \text{constant}$ (distance between interceptor and target) and

$$\mathbf{R}_p(\chi_t) = \begin{bmatrix} \cos(\chi_t) & -\sin(\chi_t) \\ \sin(\chi_t) & \cos(\chi_t) \end{bmatrix} \in SO(2) \quad (10.11)$$

Hence, the LOS steering law can be chosen as

$$\chi_r = \arctan(-e/\Delta_e) \quad (10.12)$$

where the steering law tuning parameter $\Delta_e > 0$ represents the lookahead distance. This parameter is given in meters and usually takes values between 1.5 and 2.5 of a ship length L_{pp} . Finally, the desired heading angle for the approaching ship is input to the ship autopilot, suggesting that

$$\psi_d = \chi_d - \beta \quad (10.13)$$

where the sideslip (drift) angle is

$$\beta = \arcsin\left(\frac{v}{U}\right) \quad (10.14)$$

The speed command $U_d = \sqrt{u_d^2 + v_d^2} \approx u_d$ (assuming that $u_d \gg v_d$) is computed according to (10.3), (10.4) and (10.6) such that

$$U_d = U_t - \kappa \frac{s}{\sqrt{s^2 + \Delta_s^2}} \quad (10.15)$$

where $\kappa = U_{a,\max}$ and

$$U_t = \sqrt{u_t^2 + v_t^2} \quad (10.16)$$

The speed tuning parameter $\Delta_s > 0$ specifies the rendezvous behavior towards the projection of the guide ship on to the parallel course defined by e_d , ensuring that the approach ship smoothly ramps down its total speed to U_t as the along-course distance goes to zero.

10.2 Trajectory Tracking

Guidance systems designed for tracking a smooth time-varying trajectory $y_d(t) \in \mathbb{R}^m$ are useful in many applications. The desired speed and acceleration are obtained from time-differentiation of $y_d(t)$ one and two times, respectively. This means that the signal $y_d(t)$ defines the desired position/attitude, velocity and acceleration as a function of time t for a moving craft in 6 DOF. We will make use of the following definition in the forthcoming:

Definition 10.1 (Trajectory Tracking)

A control system that forces the system output $y(t) \in \mathbb{R}^m$ to track a desired output $y_d(t) \in \mathbb{R}^m$ solves a trajectory tracking problem.

This definition is consistent with Athans and Falb (1966) and later with Hauser and Hindmann (1995), Ortega *et al.* (1998), Encarnacao and Pascoal (2001) and Skjetne *et al.* (2002, 2004). In this section, methods for computation of the desired trajectory corresponding to a desired virtual target will be presented. The following methods are discussed:

- Low-pass filters for the generation of position, velocity and acceleration (PVA) trajectories
- Time-domain simulation using an adequate model of the craft
- Optimization methods

Within this framework, it is possible to generate feasible trajectories incorporating both *spatial constraints* (obstacle avoidance and maximum velocity/acceleration) and *temporal constraints* (minimum time, on time and maximum time problems)

Trajectory-Tracking Control

Trajectory-tracking control laws are classified according to the number of available actuators. This can be illustrated by considering a marine craft in *surge*, *sway* and *yaw*. Tracking of a *time-varying reference trajectory* $\eta_d(t) = [N_d(t), E_d(t), \psi_d(t)]^\top$ is achieved by minimizing the tracking error, $e(t) := \eta(t) - \eta_d(t)$. Moreover,

$$e(t) := \begin{bmatrix} N(t) - N_d(t) \\ E(t) - E_d(t) \\ \psi(t) - \psi_d(t) \end{bmatrix} \quad (10.17)$$

Based on this interpretation, the following considerations can be made (see Section 9.3):

- **Three or more controls:** This is referred to as a *fully actuated* dynamic positioning (DP) system and typical applications are crab wise motions (low-speed maneuvering) and stationkeeping, where the goal is to drive $e(t) \in \mathbb{R}^2 \times S \rightarrow \mathbf{0}$. This is the standard configuration for offshore DP vessels. Feedback control laws for fully actuated vessels are discussed in Chapter 12.
- **Two controls and trajectory-tracking control:** Trajectory tracking in 3 DOF, $e(t) \in \mathbb{R}^2 \times S$, with only two controls, $u(t) \in \mathbb{R}^2$, is an *underactuated* control problem, which cannot be solved using linear theory. This problem has limited practical use. However, since all marine craft operate in a uniform force field due to mean wind, waves and ocean currents, it is possible to steer the craft along a path with a constant sideslip angle (given by the mean environmental force field) using only two controls, that is turning and forward speed control. This is the classical approach used in path-following control (see Section 12.2.8).
- **Two controls and weather-optimal heading:** If the ship is aligned up against the mean resulting force due to wind, waves and ocean currents, a weathervaning controller can be designed such that only two controls, $u(t) \in \mathbb{R}^2$, are needed to stabilize the ship positions. In this approach the heading angle is allowed to vary automatically with the mean environmental forces (Pinkster, 1971, Pinkster and Nienhuis, 1986, Fossen and Strand, 2001) (see Section 13.3.10).
- **Two controls and path-following control:** It is standard procedure to define a 2-D workspace (along-track and cross-track errors) and minimize the cross-track error by means of an LOS path-following controller; see Sections 10.3–10.4 and 12.2.8–12.2.9. Hence, it is possible to follow a path by using only two controls (surge speed and yaw moment). For a conventional ship this is achieved by using a rudder and a propeller only. Since the input and output vectors are of dimension two, the 6 DOF system model must be internally stable.
- **One control:** It is impossible to design stationkeeping systems and trajectory-tracking control systems in 3 DOF for a marine craft using only one control.

For underwater vehicles operating in 6 DOF it is also important to control the heave and sometimes the pitch-roll motions in addition to the surge, sway and yaw motions. However, roll and pitch can be left uncontrolled for metacentrically stable vehicles. For operation in 6 DOF, a fully actuated vehicle must have six or more actuators producing independent forces and moments in all directions in order to track a 6 DOF time-varying reference trajectory.



Figure 10.3 Joystick control system used to generate reference signals. Illustration by Bjarne Stenberg/SINTEF.

10.2.1 Reference Models for Trajectory Generation

In a practical system, it is highly advantageous to keep the software as simple as possible. As a result of this, many industrial systems are designed using linear reference models for trajectory generation. This corresponds to *open-loop guidance* as described in Section 9.2 since no feedback from the states is required. The simplest form of a reference model is obtained by using a low-pass (LP) filter structure:

$$\frac{x_d}{r}(s) = h_{lp}(s) \quad (10.18)$$

where x_d is the desired state and r denotes the reference signal usually specified by an operator (see Figure 10.3). The choice of filter should reflect the dynamics of the craft such that a feasible trajectory is constructed. For instance, it is important to take into account physical speed and acceleration limitations of the craft as well as input saturation. This is a nontrivial task so a compromise between performance and accurate tracking must be made by tuning the bandwidth of the reference model. It is important that the bandwidth of the reference model is chosen lower than the bandwidth of the motion control system in order to obtain satisfactory tracking performance and stability.

A frequently used method to generate a smooth reference trajectory $\mathbf{x}_d \in \mathbb{R}^n$ for tracking control is to use a physically motivated model. For marine craft it is convenient to use reference models

motivated by the dynamics of *mass–damper–spring systems* to generate the desired state trajectories, for instance

$$h_{lp}(s) = \frac{\omega_{n_i}^2}{s^2 + 2\zeta_i\omega_{n_i}s + \omega_{n_i}^2} \quad (10.19)$$

where ζ_i ($i = 1, \dots, n$) are the *relative damping ratios* and ω_{n_i} ($i = 1, \dots, n$) are the *natural frequencies*. For a 6 DOF system, the desired states can be expressed by a MIMO mass–damper–spring system:

$$\mathbf{M}_d \ddot{\boldsymbol{\eta}}_d + \mathbf{D}_d \dot{\boldsymbol{\eta}}_d + \mathbf{G}_d \boldsymbol{\eta}_d = \mathbf{G}_d \mathbf{r} \quad (10.20)$$

where \mathbf{M}_d , \mathbf{D}_d and \mathbf{G}_d are positive design matrices specifying the desired dynamics of the system. The model (10.20) can also be represented as a linear time invariant (LTI) system:

$$\dot{\mathbf{x}}_d = \mathbf{A}_d \mathbf{x}_d + \mathbf{B}_d \mathbf{r} \quad (10.21)$$

where $\mathbf{x}_d := [\boldsymbol{\eta}_d^\top, \dot{\boldsymbol{\eta}}_d^\top]^\top \in \mathbb{R}^{2n}$ is the desired state vector, $\mathbf{r} \in \mathbb{R}^r$ ($r \leq n$) is a bounded reference vector usually generated by a joystick or a keyboard. The state and input matrices are recognized as

$$\mathbf{A}_d = \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ -\mathbf{M}_d^{-1} \mathbf{G}_d & -\mathbf{M}_d^{-1} \mathbf{D}_d \end{bmatrix}, \quad \mathbf{B}_d = \begin{bmatrix} \mathbf{0} \\ \mathbf{M}_d^{-1} \mathbf{G}_d \end{bmatrix} \quad (10.22)$$

Velocity Reference Model

The velocity reference model should at least be of order two so as to obtain smooth reference signals for the desired velocity \mathbf{v}_d and acceleration $\dot{\mathbf{v}}_d$. Let \mathbf{r}^b denote the operator input expressed in $\{b\}$. The second-order LP filter (10.19) can be used for this purpose. Let

$$\ddot{\mathbf{v}}_d + 2\Delta\Omega\dot{\mathbf{v}}_d + \Omega^2 \mathbf{v}_d = \Omega^2 \mathbf{r}^b \quad (10.23)$$

where \mathbf{v}_d is the desired velocity, $\dot{\mathbf{v}}_d$ is the desired acceleration and $\ddot{\mathbf{v}}_d$ is interpreted as the desired “jerk”. For this model, $\Delta > 0$ and $\Omega > 0$ are diagonal design matrices of *relative damping ratios* and *natural frequencies*:

$$\Delta = \text{diag}\{\zeta_1, \zeta_2, \dots, \zeta_n\}$$

$$\Omega = \text{diag}\{\omega_{n_1}, \omega_{n_2}, \dots, \omega_{n_n}\}$$

The state space representation is

$$\mathbf{A}_d = \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ -\Omega^2 & -2\Delta\Omega \end{bmatrix}, \quad \mathbf{B}_d = \begin{bmatrix} \mathbf{0} \\ \Omega^2 \end{bmatrix} \quad (10.24)$$

Note that a step in the command \mathbf{r}^b will give a step in $\ddot{\mathbf{v}}_d$ while $\dot{\mathbf{v}}_d$ and \mathbf{v}_d will be low-pass filtered and therefore smooth signals in a tracking control system. We also notice that the steady-state velocity for a constant reference signal \mathbf{r}^b is

$$\lim_{t \rightarrow \infty} \mathbf{v}_d(t) = \mathbf{r}^b \quad (10.25)$$

Position and Attitude Reference Models

The position and attitude reference model η_d is typically chosen to be of third order for filtering the steps in r^n . This suggests that a first-order LP filter should be cascaded with a mass–damper–spring system. Moreover, consider the transfer function:

$$\frac{\eta_{d_i}}{r_i^n}(s) = \frac{\omega_{n_i}^2}{(1 + T_i s)(s^2 + 2\xi_i \omega_{n_i} s + \omega_{n_i}^2)} \quad (i = 1, \dots, n) \quad (10.26)$$

where a first-order LP filter with time constant $T_i = 1/\omega_{n_i} > 0$ has been added. This can also be written

$$\frac{\eta_{d_i}}{r_i^n}(s) = \frac{\omega_{n_i}^3}{s^3 + (2\xi_i + 1)\omega_{n_i} s^2 + (2\xi_i + 1)\omega_{n_i}^2 s + \omega_{n_i}^3}, \quad (i = 1, \dots, n) \quad (10.27)$$

or in a vectorial setting as

$$\eta_d^{(3)} + (2\Delta + I)\Omega\ddot{\eta}_d + (2\Delta + I)\Omega^2\dot{\eta}_d + \Omega^3\eta_d = \Omega^3 r^n \quad (10.28)$$

The state-space representation is

$$A_d = \begin{bmatrix} \mathbf{0} & I & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & I \\ -\Omega^3 & -(2\Delta + I)\Omega^2 & -(2\Delta + I)\Omega \end{bmatrix}, \quad B_d = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \Omega^3 \end{bmatrix} \quad (10.29)$$

In the case of n critically damped systems, $\xi_i = 1$ ($i = 1, \dots, n$), we have $\Delta = I$. Consequently,

$$\eta_d^{(3)} + 3\Omega\ddot{\eta}_d + 3\Omega^2\dot{\eta}_d + \Omega^3\eta_d = \Omega^3 r^n \quad (10.30)$$

‡

$$(s + \omega_{n_i})^3 \eta_{d_i} = \omega_{n_i}^3 r_i^n \quad (i = 1, \dots, n) \quad (10.31)$$

These reference models models also satisfy

$$\lim_{t \rightarrow \infty} \eta_d(t) = r^n \quad (10.32)$$

if $r^n = \text{constant}$.

Saturating Elements

One drawback with a linear reference model is that the time constants in the model often yield a satisfactory response for one operating point of the system while the response for other amplitudes of the operator input r_i results in completely different behavior. This is due to the exponential convergence of the signals in a linear system. One way to circumvent this problem is to use amplitude gain scheduling so that the reference model design parameters (ξ_i, ω_i) are scheduled with respect to the magnitude of the input signal r_i .

The performance of the linear reference model can also be improved by including saturation elements for velocity and acceleration according to

$$\text{sat}(x) = \begin{cases} \text{sgn}(x)x_{\max} & \text{if } |x| \geq x_{\max} \\ x & \text{else} \end{cases} \quad (10.33)$$

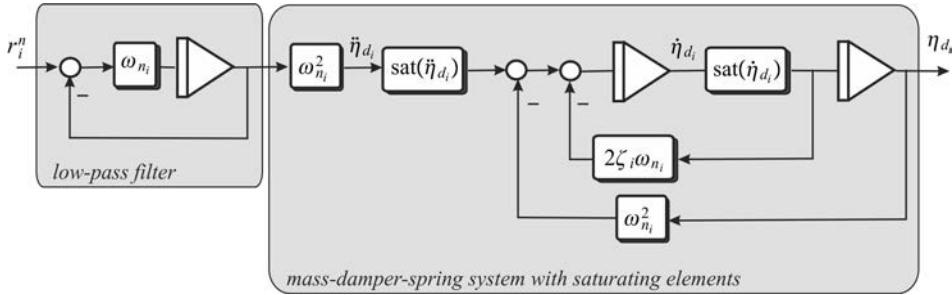


Figure 10.4 Reference model including saturating elements.

Hence, the saturation limits

$$v_i \leq v_i^{\max}, \quad \dot{v}_i \leq \dot{v}_i^{\max} \quad (10.34)$$

should reflect the physical limitations of the craft as illustrated in Example 10.2.

These techniques have been used in model reference adaptive control (MRAC) by Van Amerongen (1982, 1984) and adaptive control of underwater vehicles by Fjellstad *et al.* (1992). The position and attitude reference model should therefore be modified as shown in Figure 10.4.

Nonlinear Damping

Nonlinear damping can also be included in the reference model to reduce the velocity for large amplitudes or step inputs r_i . This suggests the modified model:

$$\ddot{\eta}_d^{(3)} + (2\Delta + I)\Omega\ddot{\eta}_d + (2\Delta + I)\Omega^2\dot{\eta}_d + d(\dot{\eta}_d) + \Omega^3\eta_d = \Omega^3r^n \quad (10.35)$$

where the nonlinear function $d(\dot{\eta}_d) = [d_1(\dot{\eta}_{d_1}), \dots, d_n(\dot{\eta}_{d_n})]^\top$ could be chosen as

$$d_i(\dot{\eta}_{d_i}) = \sum_j \delta_{ij} |\dot{\eta}_{d_i}|^{p_j} \dot{\eta}_{d_i} \quad (i = 1, \dots, n) \quad (10.36)$$

where $\delta_{ij} > 0$ are design parameters and $p_j > 0$ are some integers. The effect of nonlinear damping is demonstrated in Example 10.2.

Example 10.2 (Reference Model)

Consider the mass-damper-spring reference model:

$$\dot{\eta}_d = v_d \quad (10.37)$$

$$\ddot{v}_d + 2\xi\omega_n v_d + \delta |v_d| v_d + \omega_n^2 \eta_d = \omega_n^2 r \quad (10.38)$$

where $\xi = \omega_n = 1$. Figure 10.5 shows a comparison of responses using $\delta = 0$, $\delta = 1$ and a saturating element, $v_{\max} = 1$ for an operator step input $r = 10$. The Matlab example file *ExRefMod.m* in the MSS toolbox was used to generate the plots.

10.2.2 Trajectory Generation using a Marine Craft Simulator

The reference models in Section 10.2.1 are attractive due to their simplicity. The cutoff frequency of the reference model must never exceed the closed-loop bandwidth of the system in order to guarantee that

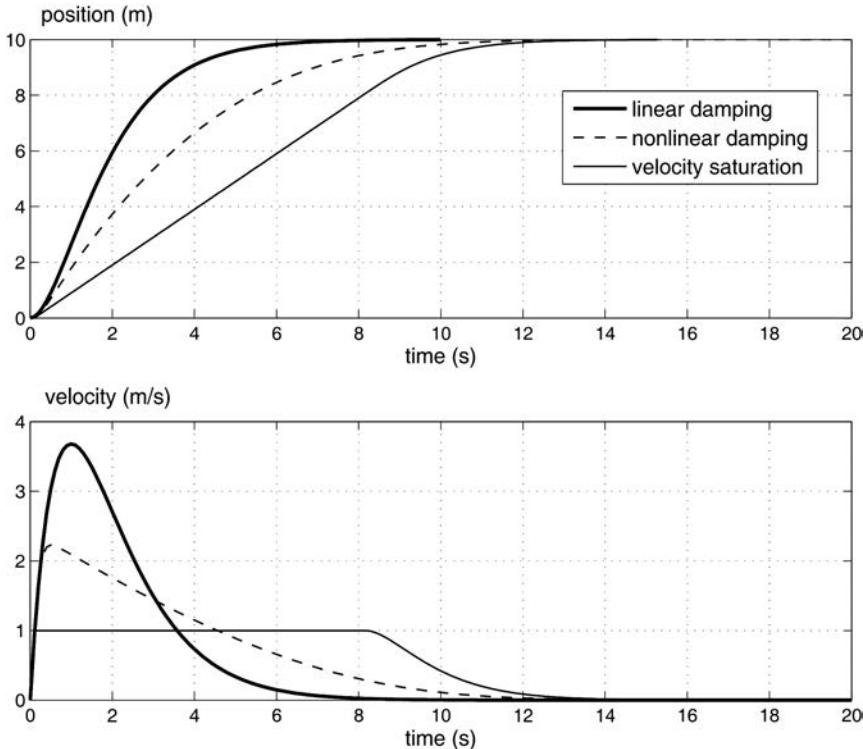


Figure 10.5 Desired position and velocity for a step input $r = 10$.

the craft is able to track the desired states. This is difficult to verify in a practical system due to factors such as nonlinearities, saturating elements and time delays. An alternative approach could be to generate a time-varying reference trajectory using a closed-loop model of the craft, where the time constants, relative damping ratios and natural frequencies are chosen to reflect physical limitations of the craft. For instance, the dynamic model can be chosen as

$$\dot{\eta}_d = J_{\Theta}(\eta_d)v_d \quad (10.39)$$

$$M\ddot{v}_d + N\dot{v}_d + g(\eta_d) = \tau \quad (10.40)$$

where the damping matrix is modeled as a diagonal matrix:

$$N = \text{diag}\{n_1, \dots, n_6\} > 0 \quad (10.41)$$

The system inertia matrix M is included in the model to guarantee proper scaling of the control inputs τ . Smooth reference trajectories $(\eta_d(t), v_d(t))$ are then obtained by simulating the model under closed-loop control, for instance by using a nonlinear PD controller (see Section 12.2):

$$\tau = g(\eta_d) - J_\Theta^\top(\eta_d) \left[K_p(\eta_d - \eta_{\text{ref}}) + K_d \dot{\eta}_d \right] \quad (10.42)$$

where η_{ref} is the setpoint and (η_d, v_d) represents the desired states. The control law (10.42) is in fact a *guidance controller* since it is applied to the reference model. In addition to this, it is useful to include saturation elements for velocity and acceleration to keep these quantities within their physical limits.

Example 10.3 (Generation of Reference Trajectories using a Marine Craft Simulator)

Consider a marine craft moving at forward speed $U \gg 0$ such that $u \approx U$ and $v \approx 0$. The desired reference trajectories can be modeled as

$$\dot{x}_d = u_d \cos(\psi_d) \quad (10.43)$$

$$\dot{y}_d = u_d \sin(\psi_d) \quad (10.44)$$

with the surge velocity given by

$$(m - X_{\dot{u}})\ddot{u}_d + \frac{1}{2}\rho C_d A |u_d| u_d = \tau \quad (10.45)$$

where $u_d \gg 0$ is the desired velocity, ρ is the density of water, C_d is the drag coefficient, A is the projected cross-sectional area of the submerged hull in the x direction and $(m - X_{\dot{u}})$ is the mass including the hydrodynamic added mass. Notice that the ship is moving so fast that quadratic drag dominates and linear damping due to skin friction can be neglected. The yaw dynamics is chosen as a first-order Nomoto model:

$$\dot{\psi}_d = r_d \quad (10.46)$$

$$T\dot{r}_d + r_d = K\delta \quad (10.47)$$

where K and T are the design parameters. The guidance system has two inputs, thrust τ and rudder angle δ . The guidance controllers for speed and yaw angle can be chosen of PI and PID types, respectively:

$$\tau = -K_{p\tau}(u_d - u_{\text{ref}}) - K_{i\tau} \int_0^t (u_d - u_{\text{ref}}) d\tau \quad (10.48)$$

$$\delta = -K_{p\delta}(\psi_d - \psi_{\text{ref}}) - K_{i\delta} \int_0^t (\psi_d - \psi_{\text{ref}}) d\tau - K_{d\delta} r_d \quad (10.49)$$

where ψ_{ref} is generated using an LOS algorithm (see Section 10.3):

$$\psi_{\text{ref}} = \text{atan2}(y_{\text{los}} - y_d(t), x_{\text{los}} - x_d(t)) \quad (10.50)$$

Numerical integration of (10.43)–(10.47) with feedback (10.48)–(10.49) yields a smooth reference trajectory (x_d, y_d, ψ_d) and speed assignment U_d .

10.2.3 Optimal Trajectory Generation

Optimization methods can be used for trajectory and path generation. This gives a systematic method for inclusion of static and dynamic constraints. However, the challenge is that an optimization problem must be solved online in order to generate a feasible time-varying trajectory. Implementation and solution of

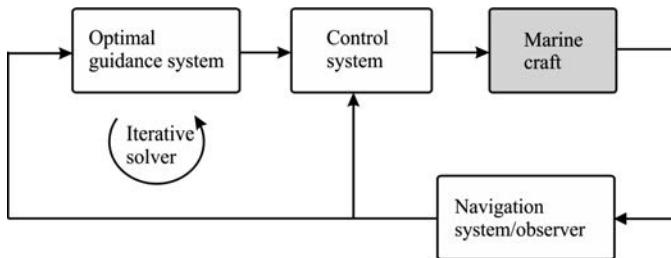


Figure 10.6 Optimal trajectory generation using an iterative solver to solve a minimum time or minimum power optimization problem.

optimization problems can be done using linear programming (LP), quadratic programming (QP) and nonlinear methods. All these methods require you to have a solver that can be implemented in your program; see Figure 10.6. For testing and development, the different algorithms can be implemented using the optimization toolbox in Matlab. The optimization problem can be formulated as minimum power or minimum time, for instance

$$J = \min_{\eta_d, v_d} \{ \text{power, time} \} \quad (10.51)$$

subject to

$$\begin{aligned} |U| &\leq U_{\max} \text{ (maximum speed)} \\ |r| &\leq r_{\max} \text{ (maximum turning rate)} \\ |u_i| &\leq u_{i,\max} \text{ (saturating limit of control } u_i) \\ |\dot{u}_i| &\leq \dot{u}_{i,\max} \text{ (saturating limit of rate } \dot{u}_i) \end{aligned}$$

which represents the constraints imposed by the vehicle dynamics. It is also possible to add constraints for obstacle avoidance and minimum fuel consumption.

10.3 Path Following for Straight-Line Paths

A trajectory describes the motion of a moving object through space as a function of time. The object might be a craft, projectile or a satellite, for example. A trajectory can be described mathematically either by the geometry of the path (see Section 10.4) or as the position of the object over time. *Path following* is the task of following a predefined path independent of time; that is there are no temporal constraints. This means that no restrictions are placed on the temporal propagation along the path. Spatial constraints, however, can be added to represent obstacles and other positional constraints.

A frequently used method for path following is *line-of-sight (LOS) guidance*. A LOS vector from the craft to the next waypoint or a point on the path between two waypoints can be used for both course and heading control. If the craft is equipped with a heading autopilot the angle between the LOS vector and the predetermined path can be used as a setpoint for the heading autopilot. This will force the craft to track the path. Guidance laws composed of *speed* and *LOS steering laws*, which can be combined in various ways to achieve different motion control objectives, are presented in the forthcoming section, which is adapted from Breivik and Fossen (2004b, 2005b, 2009) and Breivik *et al.* (2008).

10.3.1 Path Generation based on Waypoints

Systems for waypoint guidance are used both for ships and underwater vehicles. These systems consist of a waypoint generator with a human interface. The selected waypoints are stored in a waypoint database and used for generation of a trajectory or a path for the moving craft to follow. Both trajectory and path-following control systems can be designed for this purpose. Sophisticated features such as weather routing, obstacle avoidance and mission planning can be incorporated in the design of waypoint guidance systems. Some of these features will be discussed in the forthcoming section.

Waypoint Representation

The route of a ship or an underwater vehicle is usually specified in terms of waypoints. Each waypoint is defined using Cartesian coordinates (x_k, y_k, z_k) for $k = 1, \dots, n$. The waypoint *database* therefore consists of

$$\text{wpt.pos} = \{(x_0, y_0, z_0), (x_1, y_1, z_1), \dots, (x_n, y_n, z_n)\}$$

For surface craft, only two coordinates (x_k, y_k) are used. Additionally, other waypoint properties such as speed and heading can be defined, that is

$$\text{wpt.speed} = \{U_0, U_1, \dots, U_n\}$$

$$\text{wpt.heading} = \{\psi_0, \psi_1, \dots, \psi_n\}$$

For surface craft this means that the craft should pass through waypoint (x_i, y_i) at forward speed U_i with heading angle ψ_i . The three states (x_i, y_i, ψ_i) are also called the *pose*. The heading angle is usually unspecified during cross-tracking, whereas it is more important during a crab wise maneuver close to offshore installations (dynamic positioning).

The waypoint database can be generated using many criteria. These are usually based on:

- **Mission:** The craft should move from some starting point (x_0, y_0, z_0) to the terminal point (x_n, y_n, z_n) via the waypoints (x_i, y_i, z_i) .
- **Environmental data:** Information about wind, waves and ocean currents can be used for energy optimal routing (or avoidance of bad weather for safety reasons).
- **Geographical data:** Information about shallow waters and islands should be included.
- **Obstacles:** Floating constructions and other obstacles must be avoided.
- **Collision avoidance:** Avoiding moving craft close to your own route by introducing safety margins.
- **Feasibility:** Each waypoint must be feasible, in that it must be possible to maneuver to the next waypoint without exceeding the maximum speed and turning rate.

Online replanning can be used to update the waypoint database in case of time-varying conditions such as changing weather or moving craft (collision avoidance). Optimality with regard to weather is discussed in Section 10.4.1. This is referred to as weather routing.

Path Generation using Straight Lines and Circular Arcs

In practice it is common to represent the desired path using straight lines and circle arcs to connect the waypoints, as shown in Figure 10.7. This is related to the famous result of Dubins (1957), which can be summarized as:

The shortest path (minimum time) between two configurations (x, y, ψ) of a craft moving at constant speed U is a path formed by straight lines and circular arc segments.

Since a craft and not a point mass is considered, the start and end configurations of the craft are specified in terms of the positions (x, y) , heading angle ψ and speed U . In addition, it is assumed that there are bounds

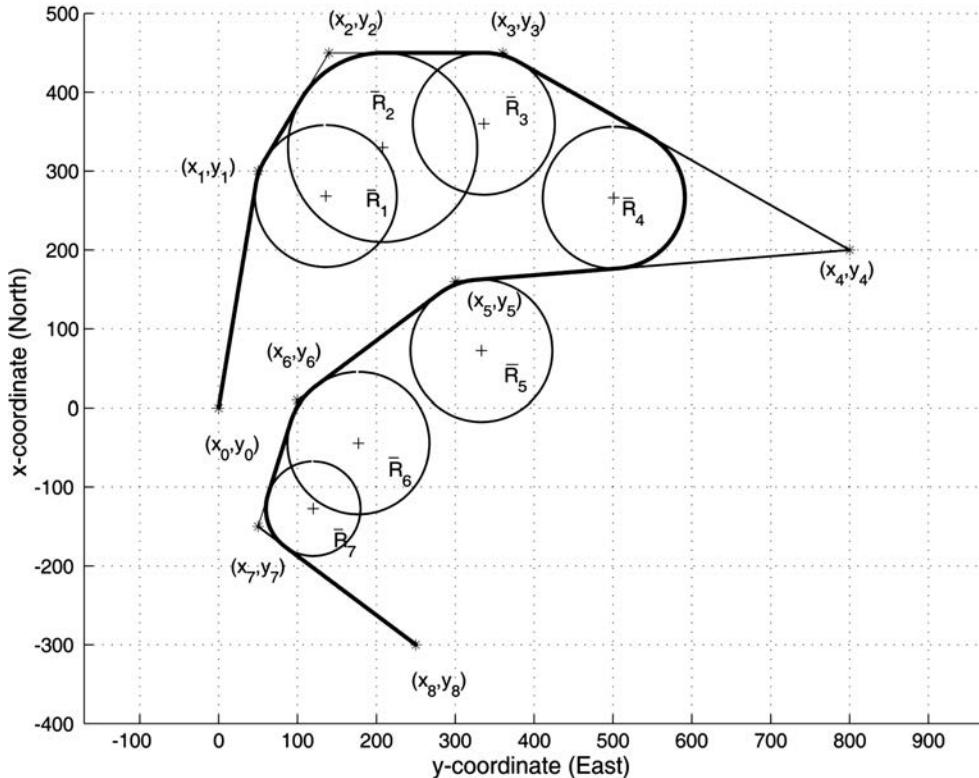


Figure 10.7 Straight lines and inscribed circles used for waypoint guidance.

on the turning rate r or the radius. The so-called Dubins path can also be proven by using *Pontryagin's maximum principle*. Generation of Dubins paths including obstacle avoidance are discussed by Tsourdos *et al.* (2010). Extensions to the case with turn rate and acceleration limits (convected Dubins path) are made by Kostov and Degtiariova-Kostova (1993) and Scheuer and Laugier (1998). Path generation for the case of uniform currents are discussed by McGee *et al.* (2006) and Techy and Woolsey (2009, 2010). In the case of time-varying speed, a dynamic optimization problem including the marine craft surge dynamics must be solved.

In this section, the discussion is limited to Dubins paths formed by straight lines and circles as shown in Figure 10.7, where the inscribed circle between two straight lines describes the desired turn. The radius of the inscribed circle is denoted \bar{R}_i ($i = 1, \dots, n$).

The drawback of this strategy, in comparison with a cubic interpolation strategy, for instance, is that a jump in the desired yaw rate r_d is experienced. This is due to the fact that the desired yaw rate along the straight line is $r_d = 0$ while it is $r_d = \text{constant}$ on the circle arc during steady turning. Hence, there will be a jump in the desired yaw rate during transition from the straight line to the circle arc. This produces a small offset during cross-tracking. If a smooth reference trajectory, for instance generated by interpolation, is used, these drawbacks are overcome. However, it is convenient to use straight lines and circle arcs due to their simplicity. Another consideration is that the human operator can specify a circle with radius R_i about each waypoint (see Figure 10.7). These values are stored in the database as

$$\text{wpt.radius} = \{R_0, R_1, \dots, R_n\}$$

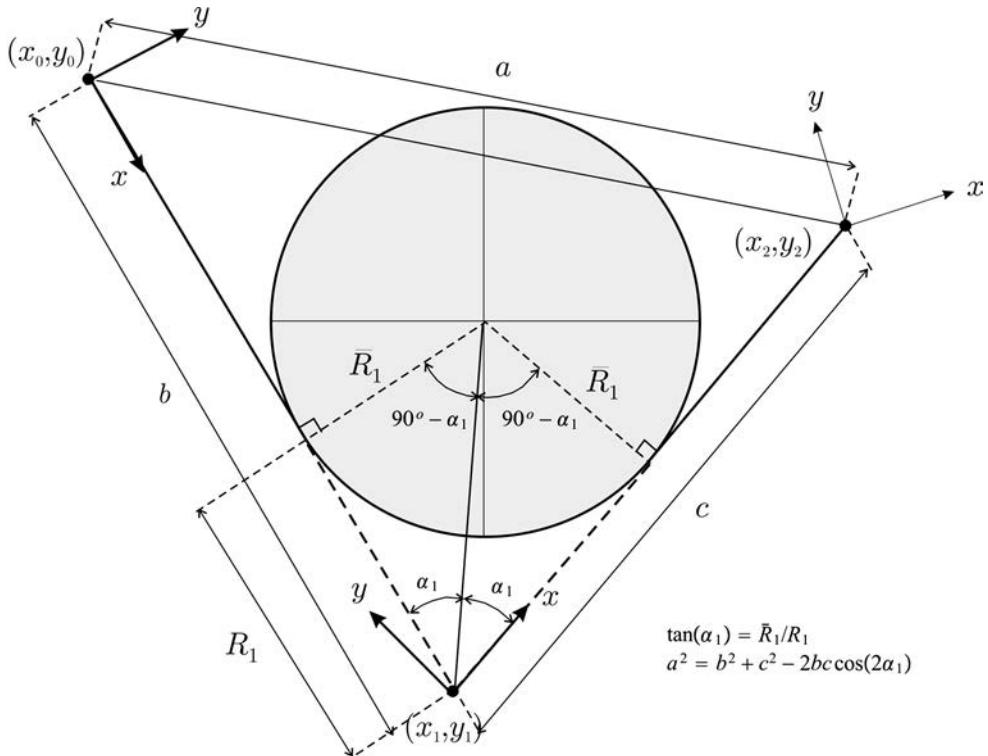


Figure 10.8 Circle with radius \bar{R}_1 inscribed between the points $(x_0, y_0), (x_1, y_1)$ and (x_2, y_2) .

The point where the circle arc intersects the straight line represents the turning point of the ship. Hence, the radius of the inscribed circle can be computed from R_i as

$$\bar{R}_i = R_i \tan(\alpha_i) \quad (i = 1, \dots, n) \quad (10.52)$$

where α_i is defined in Figure 10.8.

10.3.2 LOS Steering Laws

This section is based on Breivik and Fossen (2009) and Breivik (2010). For 2-D horizontal plane motions, the speed of the craft is defined as

$$U(t) := \|\mathbf{v}(t)\| = \sqrt{\dot{x}(t)^2 + \dot{y}(t)^2} \geq 0 \quad (10.53)$$

while steering is related to the angle

$$\chi(t) := \text{atan2}(\dot{y}(t), \dot{x}(t)) \in \mathbb{S} := [-\pi, \pi] \quad (10.54)$$

where $\text{atan2}(y, x)$ is the four-quadrant version of $\arctan(y/x) \in [-\pi/2, \pi/2]$. Path following is ensured by proper assignments to $\chi(t)$ (steering control) as long as $U(t) > 0$ (positive speed) since the scenario only involves a spatial constraint.

Consider a straight-line path implicitly defined by two waypoints $\mathbf{p}_k^n = [x_k, y_k]^\top \in \mathbb{R}^2$ and $\mathbf{p}_{k+1}^n = [x_{k+1}, y_{k+1}]^\top \in \mathbb{R}^2$, respectively. Also, consider a path-fixed reference frame with origin in \mathbf{p}_k^n whose x axis has been rotated by a positive angle:

$$\alpha_k := \text{atan2}(y_{k+1} - y_k, x_{k+1} - x_k) \in \mathbb{S} \quad (10.55)$$

relative to the x axis. Hence, the coordinates of the craft in the path-fixed reference frame can be computed by

$$\boldsymbol{\varepsilon}(t) = \mathbf{R}_p(\alpha_k)^\top (\mathbf{p}^n(t) - \mathbf{p}_k^n) \quad (10.56)$$

where

$$\mathbf{R}_p(\alpha_k) := \begin{bmatrix} \cos(\alpha_k) & -\sin(\alpha_k) \\ \sin(\alpha_k) & \cos(\alpha_k) \end{bmatrix} \in SO(2) \quad (10.57)$$

and $\boldsymbol{\varepsilon}(t) = [s(t), e(t)]^\top \in \mathbb{R}^2$ with

$s(t)$ = along-track distance (tangential to path)

$e(t)$ = cross-track error (normal to path)

For path-following purposes, only the cross-track error is relevant since $e(t) = 0$ means that the craft has converged to the straight line. Expanding (10.56), the along-track distance and cross-track error can be explicitly stated by

$$s(t) = [x(t) - x_k] \cos(\alpha_k) + [y(t) - y_k] \sin(\alpha_k) \quad (10.58)$$

$$e(t) = -[x(t) - x_k] \sin(\alpha_k) + [y(t) - y_k] \cos(\alpha_k) \quad (10.59)$$

and the associated control objective for straight-line path following becomes

$$\lim_{t \rightarrow \infty} e(t) = 0 \quad (10.60)$$

In order to ensure that $e(t) \rightarrow 0$, both course and heading angle commands can be used.

Two different guidance principles can be used to steer along the LOS vector (Breivik and Fossen, 2009):

- *Enclosure-based steering*
- *Lookahead-based steering*

and at the same time stabilize $e(t)$ to the origin. The two steering methods essentially operate by the same principle, but as will be made clear, the lookahead-based approach motivated by missile guidance has several advantages over the enclosure-based approach.

Enclosure-Based Steering

Consider a circle with radius $R > 0$ enclosing $\mathbf{p}^n = [x, y]^\top$. If the circle radius is chosen sufficiently large, the circle will intersect the straight line at two points. The enclosure-based strategy for driving $e(t)$ to zero is then to direct the velocity toward the intersection point $\mathbf{p}_{\text{los}}^n = [x_{\text{los}}, y_{\text{los}}]^\top$ that corresponds to

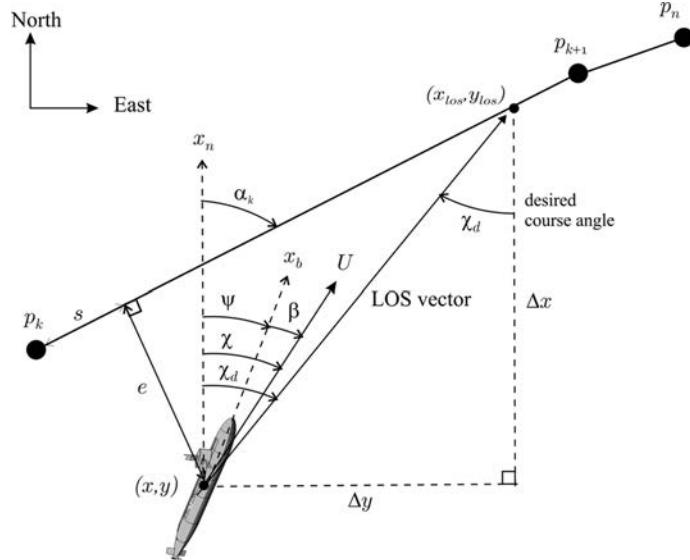


Figure 10.9 LOS guidance where the desired course angle χ_d (angle between x_n and the desired velocity vector) is chosen to point toward the LOS intersection point $(x_{\text{los}}, y_{\text{los}})$.

the desired direction of travel, which is implicitly defined by the sequence in which the waypoints are ordered. Such a solution involves directly assigning χ_d as shown in Figure 10.9. Since

$$\tan(\chi_d(t)) = \frac{\Delta y(t)}{\Delta x(t)} = \frac{y_{\text{los}} - y(t)}{x_{\text{los}} - x(t)} \quad (10.61)$$

the desired course angle can be computed as

$$\chi_d(t) = \text{atan2}(y_{\text{los}} - y(t), x_{\text{los}} - x(t)) \quad (10.62)$$

In order to calculate the two unknowns in $\mathbf{p}_{\text{los}}^n = [x_{\text{los}}, y_{\text{los}}]^\top$, the following two equations must be solved:

$$[x_{\text{los}} - x(t)]^2 + [y_{\text{los}} - y(t)]^2 = R^2 \quad (10.63)$$

$$\begin{aligned} \tan(\alpha_k) &= \frac{y_{k+1} - y_k}{x_{k+1} - x_k} \\ &= \frac{y_{\text{los}} - y_k}{x_{\text{los}} - x_k} = \text{constant} \end{aligned} \quad (10.64)$$

where (10.63) represents the *Pythagoras theorem*, while (10.64) states that the slope of the line between the two waypoints is constant. LOS guidance has been applied to surface ships by McGookin *et al.*

(2000b) and Fossen *et al.* (2003b). These equations are solved in the following, temporarily dropping the time dependence of the variables for notational convenience.

Denote the difference between the x and y positions of the two waypoints as $\Delta x := x_{k+1} - x_k$ and $\Delta y := y_{k+1} - y_k$, respectively. The equations are first solved analytically assuming that $|\Delta x| > 0$ and, second, for the case $\Delta x = 0$.

Case 1: For $|\Delta x| > 0$, Equation (10.64) results in

$$y_{\text{los}} = \left(\frac{\Delta y}{\Delta x} \right) (x_{\text{los}} - x_k) + y_k \quad (10.65)$$

when choosing to solve for y_{los} . For simplicity and brevity in the calculations that follow, denote

$$d := \left(\frac{\Delta y}{\Delta x} \right), \quad e := x_k, \quad f := y_k$$

Expanding (10.63) yields

$$x_{\text{los}}^2 - 2xx_{\text{los}} + x^2 + y_{\text{los}}^2 - 2yy_{\text{los}} + y^2 = R^2 \quad (10.66)$$

where

$$\begin{aligned} y_{\text{los}}^2 &= \left[\left(\frac{\Delta y}{\Delta x} \right) (x_{\text{los}} - x_k) + y_k \right]^2 \\ &= [dx_{\text{los}} + (f - de)]^2 \\ &= (dx_{\text{los}} + g)^2 \\ &= d^2 x_{\text{los}}^2 + 2dgx_{\text{los}} + g^2 \end{aligned} \quad (10.67)$$

where

$$g := f - de = y_k - \left(\frac{\Delta y}{\Delta x} \right) x_k$$

has been used. Subsequently, consider

$$2yy_{\text{los}} = 2y(dx_{\text{los}} + g) = 2dyx_{\text{los}} + 2gy \quad (10.68)$$

such that (10.67) and (10.68) inserted into (10.66) gives:

$$(1 + d^2)x_{\text{los}}^2 + 2(dg - dy - x)x_{\text{los}} + (x^2 + y^2 + g^2 - 2gy - R^2) = 0 \quad (10.69)$$

which is a standard, analytically solvable second-order equation. Then, denote

$$\begin{aligned} a &:= 1 + d^2 \\ b &:= 2(dg - dy - x) \\ c &:= x^2 + y^2 + g^2 - 2gy - R^2 \end{aligned}$$

from which the solution of (10.69) becomes

$$x_{\text{los}} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \quad (10.70)$$

where if $\Delta x > 0$, then $x_{\text{los}} = -b + \sqrt{b^2 - 4ac}/2a$, and if $\Delta x < 0$, then $x_{\text{los}} = -b - \sqrt{b^2 - 4ac}/2a$.

Having calculated x_{los} , y_{los} is easily obtained from (10.65). Note that when $\Delta y = 0$, $y_{\text{los}} = y_k (= y_{k+1})$.

Case 2: If $\Delta x = 0$, only (10.63) is valid, which means that

$$y_{\text{los}} = y \pm \sqrt{r^2 - (x_{\text{los}} - x)^2} \quad (10.71)$$

where $x_{\text{los}} = x_k (= x_{k+1})$. If $\Delta y > 0$, then $y_{\text{los}} = y + \sqrt{R^2 - (x_{\text{los}} - x)^2}$, and if $\Delta y < 0$, then $y_{\text{los}} = y - \sqrt{R^2 - (x_{\text{los}} - x)^2}$. When $\Delta x = 0$, $\Delta y = 0$ is not an option.

Lookahead-Based Steering

For lookahead-based steering, the course angle assignment is separated into two parts:

$$\chi_d(e) = \chi_p + \chi_r(e) \quad (10.72)$$

where

$$\chi_p = \alpha_k \quad (10.73)$$

is the *path-tangential angle* (see Figure 10.9), while

$$\chi_r(e) := \arctan \left(\frac{-e}{\Delta} \right) \quad (10.74)$$

is a *velocity-path relative angle*, which ensures that the velocity is directed toward a point on the path that is located a *lookahead distance* $\Delta(t) > 0$ ahead of the direct projection of $p''(t)$ on to the path (Papoulias, 1991).

As can be immediately noticed, this lookahead-based steering scheme is less computationally intensive than the enclosure-based approach. It is also valid for all cross-track errors, whereas the enclosure-based strategy requires $R \geq |e(t)|$. Furthermore, Figure 10.10 shows that

$$e(t)^2 + \Delta(t)^2 = R^2 \quad (10.75)$$

which means that the enclosure-based approach corresponds to a lookahead-based scheme with a time variation

$$\Delta(t) = \sqrt{R^2 - e(t)^2} \quad (10.76)$$

varying between 0 and R for $|e(t)| = R$ and $|e(t)| = 0$, respectively.

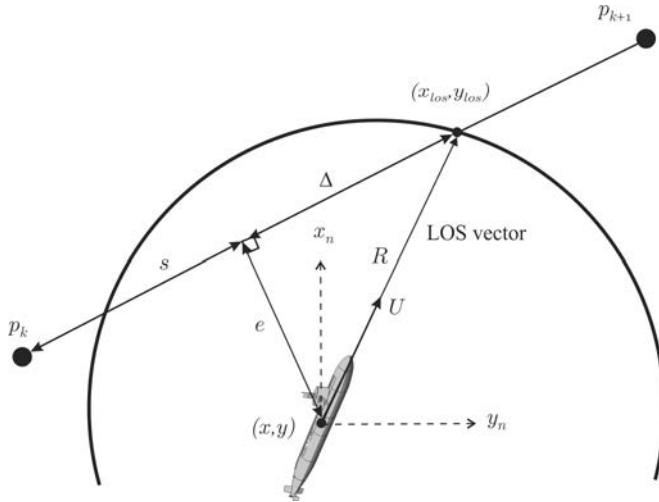


Figure 10.10 Circle of acceptance with constant radius R illustrating the geometric relationship $e(t)^2 + \Delta(t)^2 = R^2$.

The steering law (10.74) can also be interpreted as a saturating control law:

$$\chi_r(e) = \arctan(-K_p e) \quad (10.77)$$

where $K_p(t) = 1/\Delta(t) > 0$. Notice that the lookahead-based steering law is equivalent to a saturated proportional control law, effectively mapping $e \in \mathbb{R}$ into $\chi_r(e) \in [-\pi/2, \pi/2]$.

As can be inferred from the geometry of Figure 10.10, a small lookahead distance implies aggressive steering, which intuitively is confirmed by a correspondingly large proportional gain in the saturated control interpretation. This interpretation also suggests the possibility of introducing integral action into the steering law (10.74), such that

$$\chi_r(e) = \arctan\left(-K_p e - K_i \int_0^t e(\tau) d\tau\right) \quad (10.78)$$

where $K_i > 0$ represents the integral gain. Integral action can be particularly useful for underactuated craft that can only steer by attitude information, enabling them to follow straight-line paths while under the influence of ocean currents and nonzero sideslip angles β , even without having access to velocity information. Thus, considering horizontal path following along straight lines, the desired yaw angle can be computed by

$$\chi_d(e) = \alpha_k + \chi_r(e) \quad (10.79)$$

with $\chi_r(e)$ as in (10.78). In practice, to avoid overshoot and windup effects, care must be taken when using integral action in the steering law. Specifically, the integral term should only be used when a steady-state off-track condition is detected.

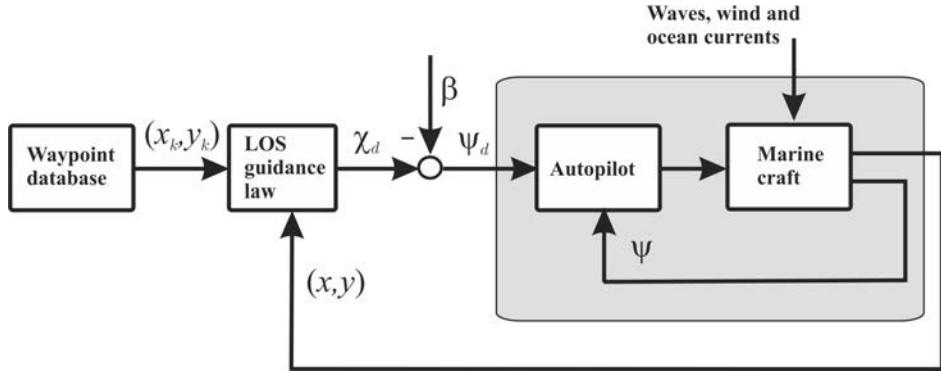


Figure 10.11 LOS guidance principle where the sideslip angle β either can be chosen as zero and compensated for by using integral action or nonzero by using velocity measurements.

Path-Following Controllers

Consider the LOS intersection point $\mathbf{p}_{\text{los}}^n$ in Figure 10.9. Different principles for path following can be applied depending on whether you have access to velocity measurements or not:

Method A (Body x axis and LOS vector aligned): Assume that the velocity is unknown and compute the desired heading angle according to the *enclosure-based steering law* (10.62):

$$\psi_d(t) = \text{atan2}(y_{\text{los}} - y(t), x_{\text{los}} - x(t)) \quad (10.80)$$

such that the body x axis of the craft points in the direction of the LOS intersection point $\mathbf{p}_{\text{los}}^n$. In this approach, the sideslip angle β is assumed to be unknown and the control objective is $\psi \rightarrow \psi_d$ (see Figure 10.11). Consequently, a *heading autopilot* of PID type is

$$\tau = -K_p \tilde{\psi} - K_d \dot{\tilde{\psi}} - K_i \int_0^t \tilde{\psi}(\tau) d\tau \quad (10.81)$$

where $\tilde{\psi} = \psi - \psi_d$ can be used. The price to be paid is that the craft will behave like an object hanging in a rope and the craft's lateral distance to the path will depend on the magnitude of the environmental forces and thus the sideslip angle β . This is due to the fact that $\psi = \chi$ only if $\beta = 0$. If such deviations cannot be accepted, the speed and LOS vectors should be aligned using Method B (see Figure 10.12).

Method B (Velocity and LOS vectors aligned): Compute the desired course angle χ_d such that the velocity vector is along the path (LOS vector) using the *lookahead-based steering law*:

$$\begin{aligned} \chi_d(e) &= \chi_p + \chi_r(e) \\ &= \alpha_k + \arctan(-K_p e) \end{aligned} \quad (10.82)$$

The control objective $\chi \rightarrow \chi_d$ is satisfied by transforming the course angle command χ_d to a heading angle command ψ_d by using (2.96). This requires knowledge of β since (see Figure 10.11)

$$\psi_d = \chi_d - \beta \quad (10.83)$$

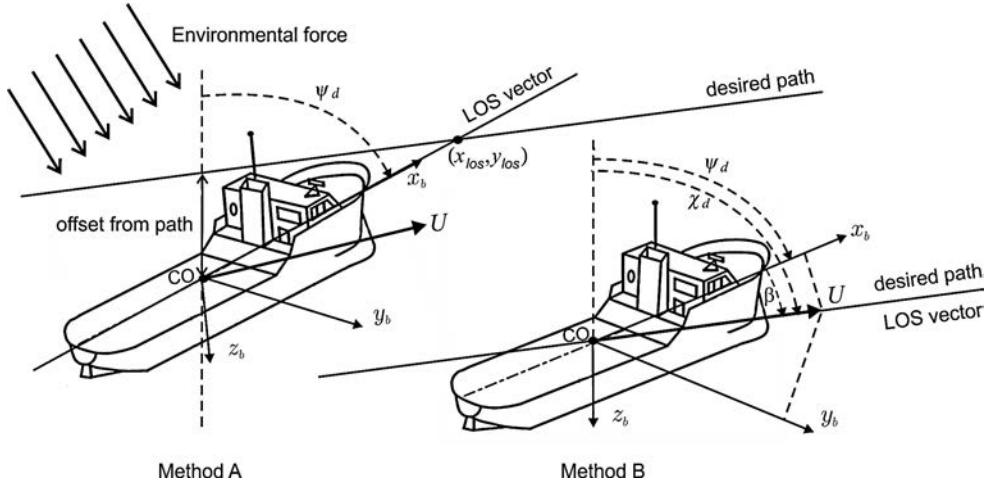


Figure 10.12 Body x -axis aligned with the LOS vector (Method A) or alternatively velocity and LOS vectors aligned (Method B). Notice that Method A gives a lateral offset to the path.

Hence, the velocity and LOS vectors can be aligned using the heading controller (12.178) with the following error signal:

$$\begin{aligned}\tilde{\psi} &= \psi - \psi_d \\ &= \psi - \chi_d + \beta\end{aligned}\quad (10.84)$$

as illustrated in Figure 10.11. If the velocities of the craft are measured, the sideslip angle can be computed by

$$\beta = \arcsin\left(\frac{v}{U}\right) \quad (10.85)$$

Guidance laws of PI type, for instance (10.78), avoid velocity measurements by treating β as an unknown slowly varying disturbance satisfying $\dot{\beta} \approx 0$.

Circle of Acceptance for Surface Vessels

When moving along a piece wise linear path made up of n straight-line segments connected by $n+1$ waypoints, a switching mechanism for selecting the next waypoint is needed. Waypoint (x_{k+1}, y_{k+1}) can be selected on the basis of whether or not the craft lies within a *circle of acceptance* with radius R_{k+1} around (x_{k+1}, y_{k+1}) . Moreover, if the craft positions (x, y) at time t satisfy

$$[x_{k+1} - x(t)]^2 + [y_{k+1} - y(t)]^2 \leq R_{k+1}^2 \quad (10.86)$$

the next waypoint (x_{k+1}, y_{k+1}) should be selected. A guideline could be to choose R_{k+1} equal to two ship lengths, that is $R_{k+1} = 2L_{pp}$.

Note that for the enclosure-based approach, such a switching criterion entails the additional (conservative) requirement $r \geq R_{k+1}$. A perhaps more suitable switching criterion solely involves the along-track

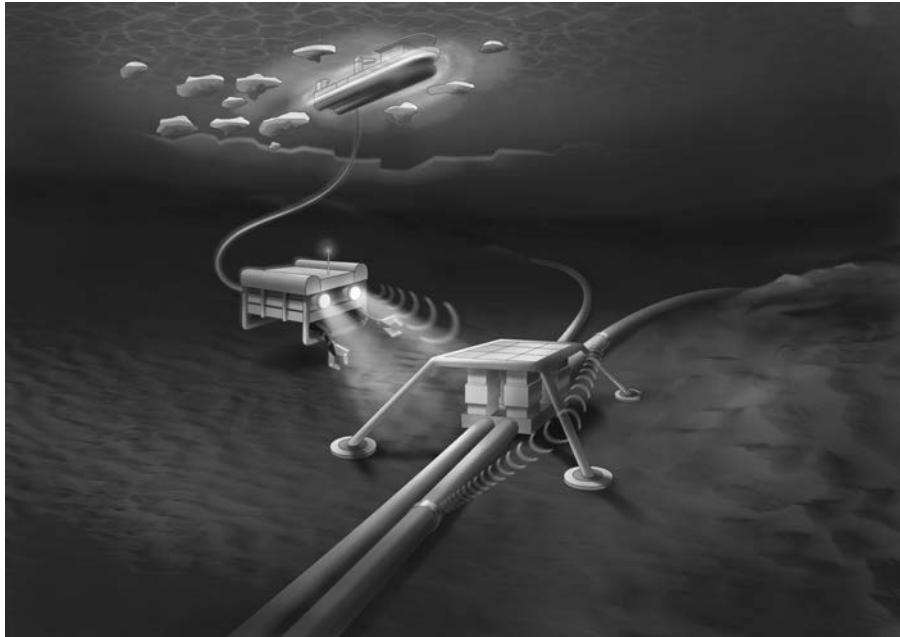


Figure 10.13 Remotely operated vehicle (ROV) performing offshore inspection and maintenance. Illustration by Bjarne Stenberg/SINTEF.

distance s , such that if the total along-track distance between waypoints \mathbf{p}_k^n and \mathbf{p}_{k+1}^n is denoted s_{k+1} , a switch is made when

$$s_{k+1} - s(t) \leq R_{k+1} \quad (10.87)$$

which is similar to (12.189) but has the advantage that $\mathbf{p}^n(t)$ does not need to enter the waypoint enclosing circle for a switch to occur; that is no restrictions are put on the cross-track error. Thus, if no intrinsic value is associated with visiting the waypoints and their only purpose is to implicitly define a piece-wise linear path, there is no reason to apply the circle-of-acceptance switching criterion (12.189).

Extension to 3-D LOS Guidance for Underwater Vehicles

It is straightforward to generalize the concepts of LOS guidance to 3-D maneuvering. Also for this case, the desired course angle χ_d can be chosen as (10.62) with the LOS intersection point given by (10.63) and (10.64) under the assumption that the vehicle performs slow maneuvers in the vertical plane such that a depth controller can easily achieve $z = z_d$. This works quite well for vehicles moving at low speed since it is not necessary to pitch the vehicle in order to move vertically; see Figure 10.13. A typical example is a working ROV with a broad, flattened front (bluff body) moving vertically using a vertical thruster. The circle of acceptance must, however, be replaced by a *sphere of acceptance* (Healey and Lienard, 1993):

$$[x_{k+1} - x(t)]^2 + [y_{k+1} - y(t)]^2 + [z_{k+1} - z(t)]^2 \leq R_{k+1}^2 \quad (10.88)$$

A more sophisticated approach will be to compute the azimuth and elevation angles needed to move in 3-D to the next waypoint (Breivik and Fossen, 2009). This approach is used for “flying” vehicles equipped with fins for diving and depth control. These vehicles move at a higher speed in order to produce lifting forces (no vertical thrusters) and consequently they behave like an aircraft, where it is possible to control the coupled surge, heave and pitch motions (longitudinal motions).

10.4 Path-Following for Curved Paths

This section relaxes the condition that the path consists of straight lines between waypoints. Instead, it is assumed that the guidance systems can make use of a predefined parametrized path. The path-following controller is a *kinematic controller* that generates the desired states for the motion control system using the parametrization of the path. The drawback is that the path must be parametrized and known in advance. In many cases this is not practical and a simpler path consisting of waypoints and straight lines must be used. The solution for this is presented in Section 10.3. Section 10.4.1 discusses path generation while a path-following controller for parametrized paths is derived in Section 10.4.2.

For a parametrized path, the following definitions are adopted from Skjetne *et al.* (2004):

Definition 10.2 (Parametrized Path)

A parametrized path is defined as a geometric curve $\eta_d(\varpi) \in \mathbb{R}^q$ with $q \geq 1$ parametrized by a continuous path variable ϖ .

For marine craft it is common to use a 3-D representation:

$$\mathbf{p}_d^n(\varpi) = [x_d(\varpi), y_d(\varpi), z_d(\varpi)]^\top \in \mathbb{R}^3 \quad (10.89)$$

where the first two coordinates describe the position in the horizontal plane and the last coordinate is the depth. For surface vessels only x_d and y_d are needed while underwater vehicles use all three coordinates. The first- and second-order derivatives of $\mathbf{p}^n(\varpi)$ with respect to ϖ are denoted as \mathbf{p}' and \mathbf{p}'' , respectively.

A frequently used solution of the path-following problem is to solve it as the geometric task of a *maneuvering problem*, given by the following definition:

Definition 10.3 (Maneuvering Problem)

The maneuvering problem involves solving two tasks:

1. Geometric Task: Force the position $\mathbf{p}^n(t)$ to converge to a desired path $\mathbf{p}_d^n(\varpi(t))$,

$$\lim_{t \rightarrow \infty} [\mathbf{p}^n(t) - \mathbf{p}_d^n(\varpi(t))] = \mathbf{0} \quad (10.90)$$

for any continuous function $\varpi(t)$.

2. Dynamic Task: Force the speed $\dot{\varpi}$ to converge to a desired speed U_d according to

$$\lim_{t \rightarrow \infty} \left[\dot{\varpi}(t) - \frac{U_d(\varpi(t))}{\sqrt{(x'_d)^2 + (y'_d)^2}} \right] = 0 \quad (10.91)$$

The dynamic task follows from

$$U_d(t) = \sqrt{\dot{x}_d^2(t) + \dot{y}_d^2(t)} = \sqrt{x'_d(\varpi)^2 + y'_d(\varpi)^2} \dot{\varpi}(t) \quad (10.92)$$

Definition 10.3 implies that the dynamics $\varpi(t)$ along the path can be specified independently of the error dynamics. A special case of the maneuvering problem is

$$\dot{\varpi}(t) = 1, \quad \varpi(0) = 0 \quad (10.93)$$

which is recognized as the tracking problem since the solution of (10.93) is $\varpi = t$. A solution to the maneuvering problem for fully actuated craft is found in Skjetne *et al.* (2004).

10.4.1 Path Generation using Interpolation Methods

The path can be generated using spline or polynomial interpolation methods to generate a curve $(x_d(\varpi), y_d(\varpi))$ through a set of N predefined waypoints. Notice that a trajectory $(x_d(t), y_d(t))$ is obtained by choosing $\dot{\varpi} = k$ such that $\varpi = kt$ where $k \in \mathbb{R}$.

Cubic Spline and Hermite Interpolation

In Matlab, several methods for interpolation are available.

Matlab

The different methods for interpolation are found by typing

```
help polyfun
```

Two useful methods for path generation are the cubic spline interpolant (`spline.m`) and the piecewise cubic Hermite interpolating polynomial (`pchip.m`).

The main difference between Hermite and cubic spline and interpolation is how the slopes at the end points are handled. For simplicity let us consider the problem of trajectory generation. The cubic Hermite interpolant ensures that the first-order derivatives $(\dot{x}_d(t), \dot{y}_d(t))$ are continuous. In addition, the slopes at each endpoint are chosen in such a way that $(x_d(t), y_d(t))$ are shape preserving and respect monotonicity.

Cubic spline interpolation is usually done by requiring that the second-order derivatives $(\ddot{x}_d(t), \ddot{y}_d(t))$ at the endpoints of the polynomials are equal, which gives a smooth spline. Consequently, the cubic spline will be more accurate than the Hermite interpolating polynomial if the data values are of a smooth function. The cubic Hermite interpolant, on the contrary, has less oscillations if the data are nonsmooth.

The results of interpolating a set of predefined waypoints to a trajectory $(x_d(\varpi), y_d(\varpi))$ using the cubic Hermite interpolant and cubic spline interpolation methods are shown in Figure 10.14. It is seen that different behaviors are obtained due to the conditions on the first- and second-order derivatives at the endpoints.

Polynomial Interpolation

Instead of using the Matlab functions `pchip.m` and `spline.m` a cubic spline can be interpolated through a set of waypoints by considering the *cubic polynomials*

$$x_d(\varpi) = a_3\varpi^3 + a_2\varpi^2 + a_1\varpi + a_0 \quad (10.94)$$

$$y_d(\varpi) = b_3\varpi^3 + b_2\varpi^2 + b_1\varpi + b_0 \quad (10.95)$$

where $(x_d(\varpi), y_d(\varpi))$ are the position of the craft and where ϖ is a path variable given by

$$\dot{\varpi} = f(\varpi, t) \quad (10.96)$$

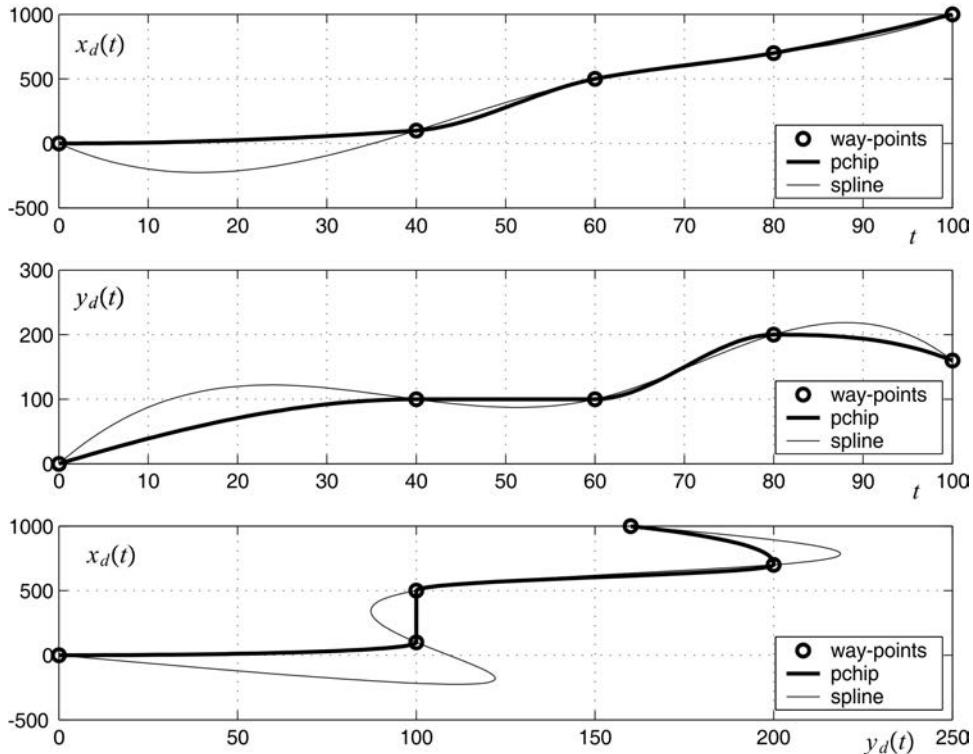


Figure 10.14 Methods for waypoint interpolation; see ExSpline.m in the MSS toolbox.

The partial derivatives of \$x_d(\varpi)\$ and \$y_d(\varpi)\$ with respect to \$\varpi\$ are

$$x'_d(\varpi) = \frac{dx_d(\varpi)}{d\varpi} = 3a_3\varpi^2 + 2a_2\varpi + a_1 \quad (10.97)$$

$$y'_d(\varpi) = \frac{dy_d(\varpi)}{d\varpi} = 3b_3\varpi^2 + 2b_2\varpi + b_1 \quad (10.98)$$

Hence, the desired speed \$U_d(t)\$ of the craft can be computed as

$$\dot{x}_d(t) = \frac{dx_d(\varpi)}{d\varpi} \dot{\varpi}(t) \quad (10.99)$$

$$\dot{y}_d(t) = \frac{dy_d(\varpi)}{d\varpi} \dot{\varpi}(t) \quad (10.100)$$

resulting in

$$\begin{aligned} U_d(t) &= \sqrt{\dot{x}_d^2(t) + \dot{y}_d^2(t)} \\ &= \sqrt{x'_d(\varpi)^2 + y'_d(\varpi)^2} \dot{\varpi}(t) \end{aligned} \quad (10.101)$$

Similarly, an expression for the acceleration $\dot{U}_d(t)$ can be found.

Matlab

The script `ExSpline.m` generates the plots in Figure 10.14:

```
% ExSpline - Cubic Hermite and spline interpolation of waypoints

wpt.pos.x = [0 100 500 700 1000];
wpt.pos.y = [0 100 100 200 160];
wpt.time = [0 40 60 80 100];

t = 0:1:max(wpt.time); % time
x_p = pchip(wpt.time,wpt.pos.x,t); % cubic Hermite interpolation
y_p = pchip(wpt.time,wpt.pos.y,t);
x_s = spline(wpt.time,wpt.pos.x,t); % spline interpolation
y_s = spline(wpt.time,wpt.pos.y,t);

subplot(311), plot(wpt.time,wpt.pos.x,'o',t,[x_p; x_s])
subplot(312), plot(wpt.time,wpt.pos.y,'o',t,[y_p; y_s])
subplot(313), plot(wpt.pos.y,wpt.pos.x,'o',y_p,x_p,y_s,x_s)
```

The unknown parameters $a_0, a_1, a_2, a_3, b_0, b_1, b_2, b_3$ in (10.94) and (10.95) can also be computed using a cubic spline algorithm, as shown below.

Cubic Spline Algorithm for Path Generation

The path through the waypoints (x_{k-1}, y_{k-1}) and (x_k, y_k) must satisfy

$$x_d(\varpi_{k-1}) = x_{k-1}, \quad x_d(\varpi_k) = x_k \quad (10.102)$$

$$y_d(\varpi_{k-1}) = y_{k-1}, \quad y_d(\varpi_k) = y_k \quad (10.103)$$

where $k = 1, \dots, n$. In addition, smoothness is obtained by requiring that

$$\lim_{\varpi \rightarrow \varpi_k^-} x'_d(\varpi) = \lim_{\varpi \rightarrow \varpi_k^+} x'_d(\varpi) \quad (10.104)$$

$$\lim_{\varpi \rightarrow \varpi_k^-} x''_d(\varpi) = \lim_{\varpi \rightarrow \varpi_k^+} x''_d(\varpi) \quad (10.105)$$

For this problem, it is possible to add only two boundary conditions (velocity or acceleration) for the x and y equations, respectively. Hence,

$$x'_d(\varpi_0) = x'_0, \quad x'_d(\varpi_n) = x'_n \quad (10.106)$$

$$y'_d(\varpi_0) = y'_0, \quad y'_d(\varpi_n) = y'_n \quad (10.107)$$

or

$$x_d''(\varpi_0) = x_0'', \quad x_d''(\varpi_n) = x_n'' \quad (10.108)$$

$$y_d''(\varpi_0) = y_0'', \quad y_d''(\varpi_n) = y_n'' \quad (10.109)$$

The polynomial $x_d(\varpi_k)$ is given by the parameters $\mathbf{a}_k = [a_{3k}, a_{2k}, a_{1k}, a_{0k}]^\top$, resulting in $4(n - 1)$ unknown parameters. The number of constraints are also $4(n - 1)$ if only velocity or acceleration constraints are chosen at the end points. The unknown parameters for n waypoints are collected into a vector:

$$\mathbf{x} = [\mathbf{a}_k^\top, \dots, \mathbf{a}_{n-1}^\top]^\top \quad (10.110)$$

Hence, the cubic interpolation problem can be written as a linear equation:

$$\mathbf{y} = \mathbf{A}(\varpi_{k-1}, \dots, \varpi_k)\mathbf{x}, \quad k = 1, 2, \dots, n \quad (10.111)$$

where

$$\mathbf{y} = [x_{\text{start}}, x_0, x_1, x_1, 0, 0, x_2, x_2, 0, 0, \dots, x_n, x_{\text{final}}]^\top \quad (10.112)$$

The start and end points can be specified in terms of velocity or acceleration constraints $x_{\text{start}} \in \{x'_0, x''_0\}$ and $x_{\text{final}} \in \{x'_n, x''_n\}$, respectively. This gives

$$\mathbf{A}(\varpi_{k-1}, \dots, \varpi_k) = \left[\begin{array}{ccccc} \mathbf{c}_{\text{start}} & \mathbf{0}_{1 \times 4} & \mathbf{0}_{1 \times 4} & \dots & \mathbf{0}_{1 \times 4} \\ \mathbf{p}(\varpi_0) & \mathbf{0}_{1 \times 4} & \mathbf{0}_{1 \times 4} & & \mathbf{0}_{1 \times 4} \\ \hline \mathbf{p}(\varpi_1) & \mathbf{0}_{1 \times 4} & \mathbf{0}_{1 \times 4} & & \mathbf{0}_{1 \times 4} \\ 0 & \mathbf{p}(\varpi_1) & \mathbf{0}_{1 \times 4} & & \mathbf{0}_{1 \times 4} \\ -\mathbf{v}(\varpi_1) & \mathbf{v}(\varpi_1) & \mathbf{0}_{1 \times 4} & & \mathbf{0}_{1 \times 4} \\ -\mathbf{a}(\varpi_1) & \mathbf{a}(\varpi_1) & \mathbf{0}_{1 \times 4} & & \mathbf{0}_{1 \times 4} \\ \hline \mathbf{0}_{1 \times 4} & \mathbf{p}(\varpi_2) & \mathbf{0}_{1 \times 4} & & \mathbf{0}_{1 \times 4} \\ \mathbf{0}_{1 \times 4} & \mathbf{0}_{1 \times 4} & \mathbf{p}(\varpi_2) & & \mathbf{0}_{1 \times 4} \\ \mathbf{0}_{1 \times 4} & -\mathbf{v}(\varpi_2) & \mathbf{v}(\varpi_2) & & \mathbf{0}_{1 \times 4} \\ \mathbf{0}_{1 \times 4} & -\mathbf{a}(\varpi_2) & \mathbf{a}(\varpi_2) & & \mathbf{0}_{1 \times 4} \\ \hline \vdots & & & & \ddots \\ \hline \mathbf{0}_{1 \times 4} & \mathbf{0}_{1 \times 4} & \mathbf{0}_{1 \times 4} & & \mathbf{p}(\varpi_n) \\ \mathbf{0}_{1 \times 4} & \mathbf{0}_{1 \times 4} & \mathbf{0}_{1 \times 4} & \dots & \mathbf{c}_{\text{final}} \end{array} \right] \quad (10.113)$$

where $\mathbf{c}_{\text{start}} \in \{x'_d(\varpi_0), x''_d(\varpi_0)\}$, $\mathbf{c}_{\text{final}} \in \{x'_d(\varpi_n), x''_d(\varpi_n)\}$ and

$$\mathbf{p}(\varpi_k) = [\varpi_k^3, \varpi_k^2, \varpi_k, 1] \quad (10.114)$$

$$\mathbf{v}(\varpi_k) = \mathbf{p}'(\varpi_k) = [3\varpi_k^2, 2\varpi_k, 1, 0] \quad (10.115)$$

$$\mathbf{a}(\varpi_k) = \mathbf{p}''(\varpi_k) = [6\varpi_k, 2, 0, 0] \quad (10.116)$$

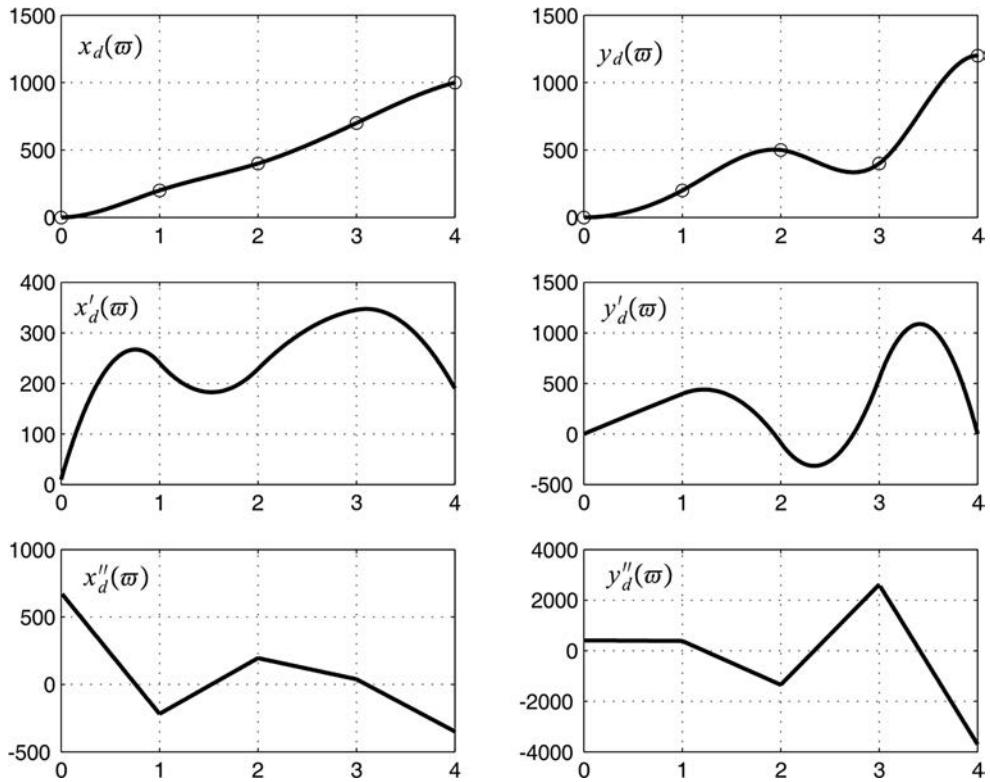


Figure 10.15 Polynomials $x_d(\varpi)$ and $y_d(\varpi)$ and their first- and second-order derivatives.

Equation (10.111) can be solved for $\varpi_k = 0, 1, \dots, n$ according to

$$\mathbf{x} = \mathbf{A}^{-1} \mathbf{y} \quad (10.117)$$

The formulae for $\mathbf{b}_k = [b_{3k}, b_{2k}, b_{1k}, b_{0k}]^\top$ are obtained in a similar manner.

Matlab

Formula (10.117) has been implemented in the script `ExPathGen.m` and `pva.m`. The results are for the following set of waypoints:

```
wpt.pos.x = [0 200 400 700 1000]
wpt.pos.y = [0 200 500 400 1200]
```

where $\varpi = 0, \dots, 4$ are shown in Figures 10.15 and 10.16.

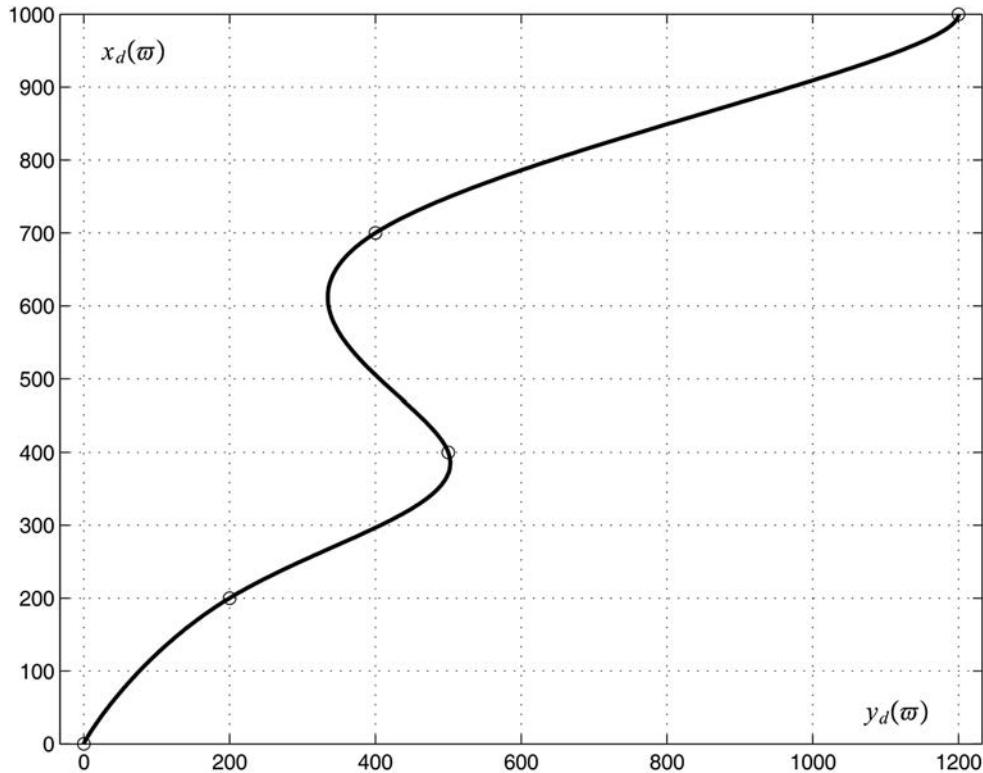


Figure 10.16 The xy plot based on a cubic spline.

Transformation of Path to Reference Trajectories using Desired Speed Profiles

In Figure 10.16 it is seen that the solution between two successive waypoints

$$x_d(\varpi) = a_3\varpi^3 + a_2\varpi^2 + a_1\varpi + a_0 \quad (10.118)$$

$$y_d(\varpi) = b_3\varpi^3 + b_2\varpi^2 + b_1\varpi + b_0 \quad (10.119)$$

is indeed a *time-independent* path when $x_d(\varpi)$ is plotted against $y_d(\varpi)$ for increasing ϖ values.

The path can be transformed to a *time-varying* trajectory by defining a *speed profile*. The speed profile assigns dynamics to $\varpi(t)$ such that the desired path transforms to a time-dependent reference trajectory at the same time as the desired speed and acceleration profiles are preserved. From (10.101) it is seen that

$$\dot{\varpi}(t) = \frac{U_d(t)}{\sqrt{x'_d(\varpi)^2 + y'_d(\varpi)^2}}, \quad \varpi(t_k) = k \quad (10.120)$$

where $\varpi(t_k) = k$ is the initial condition of the differential equation and $U_d(t)$ is the desired speed profile. Let U_{ref} be the input to a first-order system:

$$T \dot{U}_d(t) + U_d(t) = U_{\text{ref}}, \quad T > 0 \quad (10.121)$$

A smooth transition from the desired speed $U_d(t_k)$ at waypoint k to the next waypoint $k + 1$ can be made by using

$$U_{\text{ref}} = U_d(t_{k+1}) \quad (10.122)$$

This is illustrated in the following example.

Example 10.4 (Transformation of Path to Reference Trajectories)

Consider the first two waypoints in the example file `ExPathGen.m`:

$$(x_0, y_0) = (0, 0)$$

$$(x_1, y_1) = (200, 200)$$

The cubic polynomials satisfying (10.117) are

$$x_d(\varpi) = -29.89 \varpi^3 + 135.63 \varpi^2 + 94.25 \varpi$$

$$y_d(\varpi) = 108.05 \varpi^3 - 2.30 \varpi^2 + 94.25 \varpi$$

for $\varpi \in [0, 1]$. Let the speed dynamics time constant be $T = 10$ s. Assume that the craft is initially at rest ($U_d(t_0) = 0$) and that the desired speed of waypoint number 1 is $U_{\text{ref}} = U_d(t_1) = 5.0$ m/s. The numerical solutions of

$$\dot{\varpi}(t) = \frac{U_d(t)}{\sqrt{x'_d(\varpi)^2 + y'_d(\varpi)^2}} \quad (10.123)$$

$$T \dot{U}_d(t) + U_d(t) = U_{\text{ref}} \quad (10.124)$$

for waypoints 0 and 1 corresponding to $\varpi_0(t_0) = 0$ and $\varpi_1(t_1) = 1$ with $t_0 = 0$ and t_1 unknown, is shown in Figure 10.17; see `ExPathGen.m`. It is seen that the desired speed of 5.0 m/s is reached in approximately 67 s. Hence, the terminal time must be chosen as $t_1 \geq 67$ s (corresponding to $\varpi(t_1) = 1$) in order to satisfy the desired speed dynamics. If $t_1 < 67$ s there is not enough time to reach the desired speed of waypoint 1 unless the time constant T is reduced. The time constant should reflect what is physically possible for the craft. Notice that the path $(x_d(\varpi), y_d(\varpi))$ has been transformed to a time-varying reference trajectory $(x_d(t), y_d(t))$ by assigning a speed profile (10.123) to be solved numerically with the path planner (10.117). This gives design flexibility since the path can be generated off-line using a waypoint database while speed is assigned to the path when the dynamics of the actual craft is considered.

Nonlinear Constrained Optimization

Another solution to trajectory and path generation is to use nonlinear constrained optimization techniques. These methods allow an object function to be specified where minimum time and energy are design goals. In addition, the speed and acceleration constraints of the craft can be added. The drawback is that nonlinear constraint optimization problems are much harder to solve numerically than the methods described

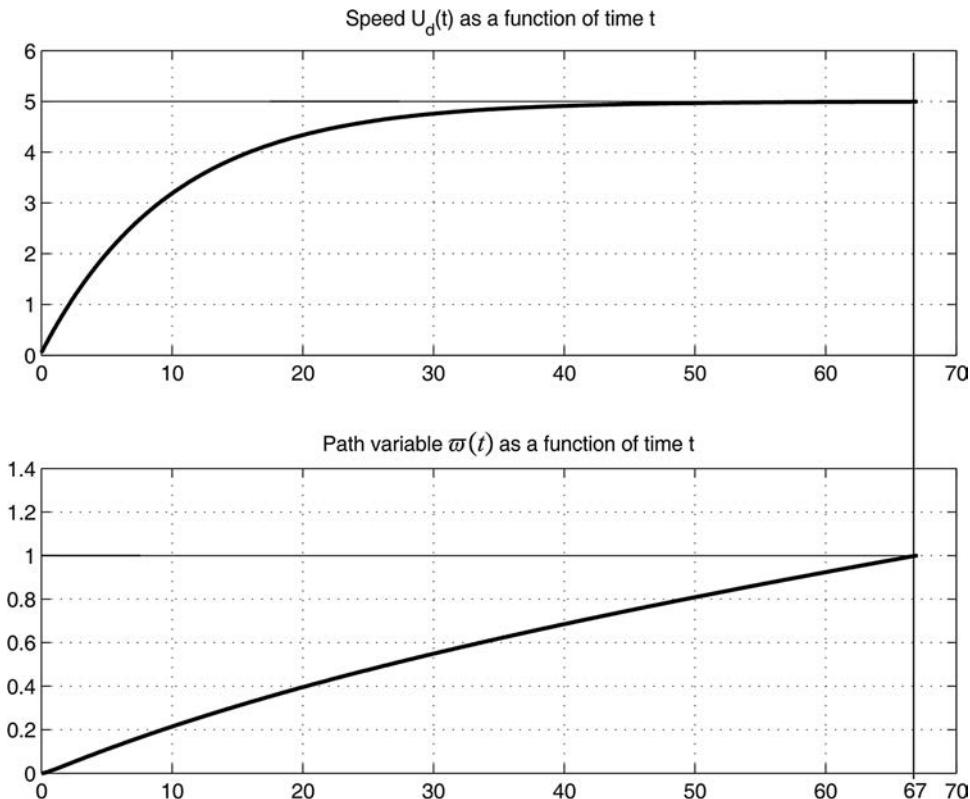


Figure 10.17 Upper plot shows that the speed $U_d(t)$ reaches the desired value of 5.0 m/s in approximately 67 s. The lower plot shows that the path variable $w(t)$ is incremented from 0 to 1 during the speed transition.

in the previous sections. The Matlab optimization toolbox will be used to demonstrate how this can be done.

In general, trajectory-tracking and path-planning problems can be formulated as

$$\begin{aligned}
 J &= \min_{\mathbf{x}} \{f(\mathbf{x})\} \\
 \text{subject to} \quad g_k(\mathbf{x}) &\leq 0 \quad (k = 1, \dots, n_g) \\
 h_j(\mathbf{x}) &= 0 \quad (j = 1, \dots, n_h) \\
 x_{i,\min} &\leq x_i \leq x_{i,\max} \quad (i = 1, \dots, n_x)
 \end{aligned} \tag{10.125}$$

where $f(\mathbf{x})$ should be minimized with respect to the parameter vector \mathbf{x} with $g_i(\mathbf{x})$ and $h_j(\mathbf{x})$ as non-linear inequality and equality constraints, respectively. An attractive simplification is to use quadratic

programming. Consequently,

$$\begin{aligned} J = \min_{\mathbf{x}} \quad & \left\{ \frac{1}{2} \mathbf{x}^\top \mathbf{H} \mathbf{x} + \mathbf{f}^\top \mathbf{x} \right\} \\ \text{subject to} \quad & \mathbf{A} \mathbf{x} \leq \mathbf{b} \\ & x_{i,\min} \leq x_i \leq x_{i,\max} \quad (i = 1, \dots, n_x) \end{aligned} \quad (10.126)$$

For simplicity, consider two waypoints (x_k, y_k) and (x_{k+1}, y_{k+1}) satisfying

$$x(t_k) = x_k, \quad y(t_k) = y_k \quad (10.127)$$

$$x(t_{k+1}) = x_{k+1}, \quad y(t_{k+1}) = y_{k+1} \quad (10.128)$$

Choosing the speed constraints as

$$\dot{x}_d(t) = U_d(t) \cos(\psi_d(t)) \quad (10.129)$$

$$\dot{y}_d(t) = U_d(t) \sin(\psi_d(t)) \quad (10.130)$$

where the angle $\psi_d(t)$ is computed as $\psi_d(t_k) = \text{atan2}(y_{k+1} - y_k, x_{k+1} - x_k)$, that is with direction toward the next waypoint. Hence,

$$\dot{x}_d(t_k) = U_k \cos(\psi_k) \quad (10.131)$$

$$\dot{y}_d(t_k) = U_k \sin(\psi_k) \quad (10.132)$$

For two waypoints this results in

$$\mathbf{y} = \mathbf{A}(t_k, t_{k+1}) \mathbf{x} \quad (10.133)$$

where

$$\mathbf{y} = [x_k, x_{k+1}, y_k, y_{k+1}, U_k \cos(\psi_k), U_k \sin(\psi_k), U_{k+1} \cos(\psi_{k+1}), U_{k+1} \sin(\psi_{k+1})]^\top \quad (10.134)$$

and

$$\mathbf{A}(t_k, t_{k+1}) = \begin{bmatrix} t_k^3 & t_k^2 & t_k & 1 & 0 & 0 & 0 & 0 \\ t_{k+1}^3 & t_{k+1}^2 & t_{k+1} & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & t_k^3 & t_k^2 & t_k & 1 \\ 0 & 0 & 0 & 0 & t_{k+1}^3 & t_{k+1}^2 & t_{k+1} & 1 \\ 3t_k^2 & 2t_k & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 3t_k^2 & 2t_k & 1 & 0 \\ 3t_{k+1}^2 & 2t_{k+1} & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 3t_{k+1}^2 & 2t_{k+1} & 0 & 0 \end{bmatrix} \quad (10.135)$$

The criterion to minimize is

$$J = \min_{\mathbf{x}} \left\{ [\mathbf{A}(t_k, t_{k+1}) \mathbf{x} - \mathbf{y}]^\top [\mathbf{A}(t_k, t_{k+1}) \mathbf{x} - \mathbf{y}] \right\} \quad (10.136)$$

for given pairs (t_k, t_{k+1}) of time. Expanding this expression yields

$$\bar{J} = \frac{1}{2} (J - \mathbf{y}^\top \mathbf{y}) = \min_{\mathbf{x}} \left\{ \frac{1}{2} \mathbf{x}^\top \mathbf{A}^\top(t_k, t_{k+1}) \mathbf{A}(t_k, t_{k+1}) \mathbf{x} - \mathbf{y}^\top \mathbf{A}(t_k, t_{k+1}) \mathbf{x} \right\} \quad (10.137)$$

implying that

$$\mathbf{H} = \mathbf{A}^\top(t_k, t_{k+1}) \mathbf{A}(t_k, t_{k+1}) \quad (10.138)$$

$$\mathbf{f} = -\mathbf{y}^\top \mathbf{A}(t_k, t_{k+1}) \quad (10.139)$$

In this expression, the starting time t_k is given while the arrival time t_{k+1} is unknown. The cubic polynomials (10.94)–(10.95) imply that there are eight additional unknown parameters to optimize:

$$\mathbf{x} = [a_3, a_2, a_1, a_0, b_3, b_2, b_1, b_0]^\top \quad (10.140)$$

giving a total of nine unknown parameters. In addition, linear constraints $\mathbf{Ax} \leq \mathbf{b}$ can be added. The reference trajectory can be found using quadratic programming.

Matlab

Trajectory generation using the optimization toolbox is demonstrated in the following example:

Example 10.5 (Trajectory Generation using Quadratic Programming)

Consider two waypoints:

$$(x_0, y_0) = (10, 10)$$

$$(x_1, y_1) = (200, 100)$$

with the speed constraint

$$U_d(t) \leq 10 \text{ m/s}$$

in the MSS toolbox script

```
ExQuadProg
```

The desired waypoint speeds are $U_0(t_0) = 0$ m/s and $U_1(t_1) = 5$ m/s with $t_0 = 0$ s. The arrival time t_1 is computed in a loop by solving the quadratic optimization problem (10.126) for each time $t_1 = t_0 + dt$ where dt is incremented by 1.0 s each time. This process is terminated when the first solution $U_d(t) \leq 10$ m/s is reached (this can be easily changed if other requirements are more important). The optimal solution:

$$x_d(t) = -0.0102 t^3 + 0.5219 t^2 - 4.28 \times 10^{-12} t + 10.0$$

$$y_d(t) = -0.0048 t^3 + 0.2472 t^2 - 1.04 \times 10^{-12} t + 10.0$$

for $t \in [t_0, t_1]$ is obtained after 29 loops ($t_1 = 29$ s) using `quadprog.m` in the Matlab optimization toolbox. The results are shown in Figure 10.18.

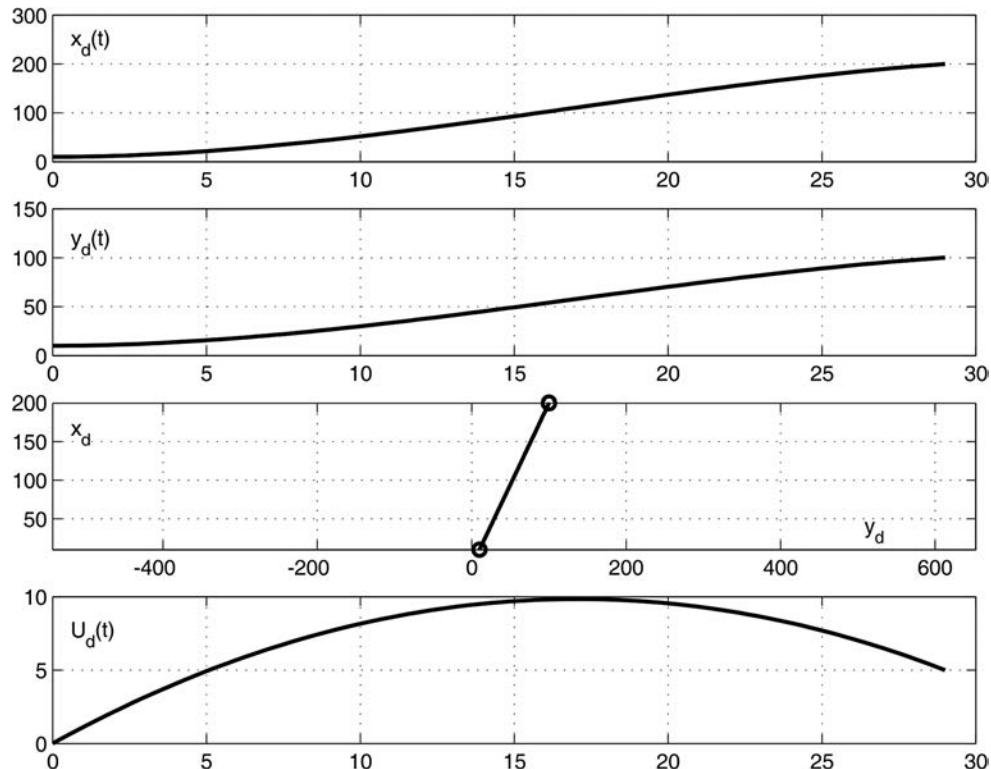


Figure 10.18 The two upper plots show the cubic polynomials $x_d(t)$ and $y_d(t)$. In the third plot $y_d(t)$ is plotted against $x_d(t)$ while the lower plot is speed $U_d(t)$.

Weather Routing

A weather routing or voyage planning system (VPS) computes the most efficient route using meteorological and oceanographic data, information about the craft's hull and propulsion system and shipping economics to ensure that the craft reaches port on time. The data from this analysis can be waypoints with optimal speed and heading information. The routing software of a modern weather routing system includes features such as:

- Surface analysis and forecast models
- Sea state and wind wave models
- Upper air models
- Formation description of low-pressure systems
- Hurricanes and tropical weather models
- Ocean current models
- Vessel performance models
- Cargo condition, trim, draft and deck load
- Link to Internet sources for weather data
- Interface to a satellite system transmitting weather data

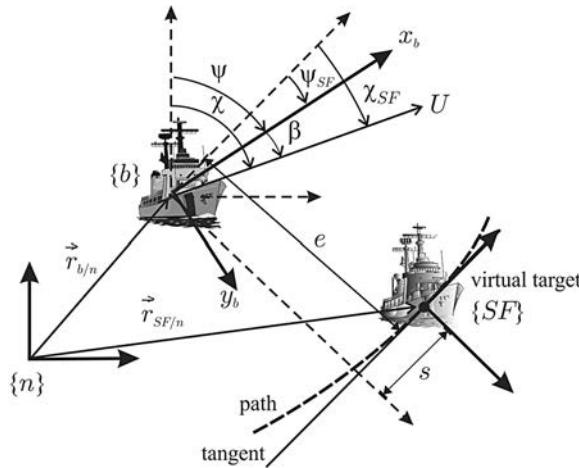


Figure 10.19 Kinematic description of the Serret–Frenet frame.

- Optimization of routes based on a fixed estimated time of arrival (ETA)
- Routing of vessels around hazardous weather conditions

An optimal route is computed using a numerical optimization offline. This can be done by a computer onboard the craft or by a company onshore transmitting the results to the craft electronically on a 24-hour basis. Several companies offer continuous voyage monitoring with status reports and performance evaluations. This allows for replanning during changing weather conditions. Global weather information is available from several forecast centers.

Some useful references for weather routing of ships are Calvert (1989), Hagiwara (1989), Padadakis and Perakis (1990), Lo (1991), Barbier *et al.* (1994), Lo and McCord (1995), McCord and Smith (1995) and Lo and McCord (1998).

10.4.2 Path-Following Kinematic Controller

The path-following controller is a *kinematic controller* that generates the desired states for the motion control system. For a parametrized path in 2-D,

$$\mathbf{p}_d^n(\varpi) = \begin{bmatrix} x_d(\varpi) \\ y_d(\varpi) \end{bmatrix} \in \mathbb{R}^2 \quad (10.141)$$

the kinematic controller can be designed using a dynamic model of the craft by specifying a reference frame that moves along the path; see Figure 10.19. This reference frame is usually chosen as the *Serret–Frenet frame* (see Frenet, 1847, Serret, 1851). During path following, the craft speed is denoted U and the kinematic controller is designed to: (i) regulate the distance e between the vehicle and the path to zero and (ii) regulate the angle χ_{SF} between the craft speed vector and the tangent to the path to zero (see Samson, 1992, Micaelli and Samson, 1993).

Definition 10.4 (Serret–Frenet Frame)

The virtual target defined by the projection of an actual craft on to a path-tangential reference frame (Serret–Frenet frame {SF}) evolves according to (Lapierre and Soetanto, 2007)

$$\dot{s} = U \cos(\chi_{\text{SF}}) - (1 - \kappa e) \dot{s}_a \quad (10.142)$$

$$\dot{e} = U \sin(\chi_{\text{SF}}) - \kappa s \dot{s}_a \quad (10.143)$$

$$\dot{\chi}_{\text{SF}} = r + \dot{\beta} - \kappa \dot{s}_a \quad (10.144)$$

where U is the speed of the craft and (e, s) is the location on the path of $\{\text{SF}\}$ relative to $\{b\}$. If $s = 0$, the variable e represents the closest distance between the actual craft and the origin of $\{\text{SF}\}$ tangential to the path. Hence, s can be viewed as an extra controller design parameter for evolution along the path. The arc length that the target has moved along the path is denoted s_a while χ_{SF} is the angle between the x axis of $\{\text{SF}\}$ and the speed vector; see Figure 10.19. Finally, κ is the path curvature.

Proof: From Figure 10.19, it is seen that the distance vectors between $\{n\}$, $\{b\}$ and $\{\text{SF}\}$ satisfies

$$\vec{r}_{b/n} = \vec{r}_{\text{SF}/n} + \vec{r}_{b/\text{SF}} \quad (10.145)$$

Hence, the time differentiation of $\vec{r}_{b/\text{SF}}$ with $\{b\}$ as the moving reference frame gives

$$\frac{^i \mathbf{d}}{dt} \vec{r}_{b/\text{SF}} = \frac{^b \mathbf{d}}{dt} \vec{r}_{b/\text{SF}} + \vec{\omega}_{b/i} \times \vec{r}_{b/\text{SF}} \quad (10.146)$$

such that

$$\vec{v}_{b/n} = \vec{v}_{\text{SF}/n} + \left(\frac{^b \mathbf{d}}{dt} \vec{r}_{b/\text{SF}} + \vec{\omega}_{\text{SF}/n} \times \vec{r}_{b/\text{SF}} \right) \quad (10.147)$$

Expressing this in $\{\text{SF}\}$ gives

$$\vec{v}_{b/n}^{\text{SF}} = \vec{v}_{\text{SF}/n}^{\text{SF}} + \left(\frac{^b \mathbf{d}}{dt} \vec{r}_{b/\text{SF}}^{\text{SF}} + \vec{\omega}_{\text{SF}/n}^{\text{SF}} \times \vec{r}_{b/\text{SF}}^{\text{SF}} \right) \quad (10.148)$$

where $\vec{r}_{b/\text{SF}}^{\text{SF}} = [s, e, 0]^T$ and $\vec{v}_{b/n}^{\text{SF}} = \mathbf{R}_{z, \chi_{\text{SF}}} [U, 0, 0]^T$ is the velocity of the vehicle expressed in $\{\text{SF}\}$. From this it follows that

$$\begin{aligned} \mathbf{R}_{z, \chi_{\text{SF}}} \begin{bmatrix} U \\ 0 \\ 0 \end{bmatrix} &= \vec{v}_{\text{SF}/n}^{\text{SF}} + \left(\frac{^b \mathbf{d}}{dt} (\vec{r}_{b/\text{SF}}^{\text{SF}}) + \vec{\omega}_{\text{SF}/n}^{\text{SF}} \times \vec{r}_{b/\text{SF}}^{\text{SF}} \right) \\ &= \begin{bmatrix} \dot{s}_a \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} \dot{s} \\ \dot{e} \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \kappa \dot{s}_a \end{bmatrix} \times \begin{bmatrix} s \\ e \\ 0 \end{bmatrix} \end{aligned} \quad (10.149)$$

Expanding this expression yields

$$U \cos(\chi_{\text{SF}}) = \dot{s} + (1 - \kappa e) \dot{s}_a \quad (10.150)$$

$$U \sin(\chi_{\text{SF}}) = \dot{e} + \kappa s \dot{s}_a \quad (10.151)$$

which proves (10.142) and (10.143). The rotation rate of the angle $\psi - \psi_{\text{SF}}$ between $\{n\}$ and $\{\text{SF}\}$ (see Figure 10.19) is $(\dot{\psi} - \dot{\psi}_{\text{SF}}) = \kappa \dot{s}_a$. Since $\chi_{\text{SF}} = \psi_{\text{SF}} + \beta$, it follows that

$$\begin{aligned}\dot{\chi}_{\text{SF}} &= \dot{\psi} - \kappa \dot{s}_a + \dot{\beta} \\ &= r + \dot{\beta} - \kappa \dot{s}_a\end{aligned}\quad (10.152)$$

which proves (10.144).

Remark 10.1

If $\dot{s} = s = 0$, the $\{\text{SF}\}$ equations become

$$\dot{s}_a = \frac{U \cos(\chi_{\text{SF}})}{1 - \kappa e} \quad (10.153)$$

$$\dot{e} = U \sin(\chi_{\text{SF}}) \quad (10.154)$$

where the term $1 - \kappa e$ in the denominator creates a singularity. Hence, the control law requires that the initial position of the craft must be restricted to a tube around the path with radius less than $1/\kappa_{\max}$. A discussion on the limitation of this approach is found in Breivik and Fossen (2004a). The constraint $1 - \kappa e \neq 0$ is, however, removed by using (10.142)–(10.143) where an additional controller parameter s allows the origin of the $\{\text{SF}\}$ frame to evolve along the path (Lapierre and Soetanto, 2007).

Remark 10.2

In Encarnacao et al. (2000), the ocean current velocities are included in the kinematic equations of motion together with a state estimator to obtain the optimal sideslip angle during path following. This section presents a different approach where the current velocities are modeled as physical forces, with moments in the expression for $\dot{\beta}$ representing the equation of motion in sway. Furthermore, the ocean currents are compensated for by using integral action in the kinematic controller to reduce sensitivity to model parameters.

Marine Craft Model for Kinematic Controller

For a conventional marine craft with no actuation in the transverse direction, the sway force can be approximated by a maneuvering model (see Section 7.1.1):

$$(m - Y_v)\dot{v} + mur = Y(u_r, v_r, r) \quad (10.155)$$

where the hydrodynamic force is due to added mass and linear damping:

$$Y(u_r, v_r, r) = X_u u_r r + Y_v v_r + Y_r r \quad (10.156)$$

The relative velocities satisfy (see Section 2.4.2)

$$u_r = u - u_c \quad (10.157)$$

$$v_r = v - v_c \quad (10.158)$$

Since ocean currents are slowly varying and the craft speed is constant, the sway acceleration can be approximated by time differentiation of

$$v = U \sin(\beta) \quad (10.159)$$

under the assumptions that $\dot{U} = 0$. This gives

$$\dot{v} = U \cos(\beta) \dot{\beta} \quad (10.160)$$

Combining (10.155)–(10.156) and (10.160) gives

$$\begin{aligned}\dot{\beta} &= \frac{1}{U \cos(\beta)} \dot{v} \\ &= \frac{1}{(m - Y_v) U \cos(\beta)} [X_{\dot{u}} u_r r + Y_v v_r + Y_r r - m u_r]\end{aligned}\quad (10.161)$$

Consequently,

$$\dot{\beta} = \frac{1}{(m - Y_v) U \cos(\beta)} [(Y_r - (m - X_{\dot{u}}) U \cos(\beta) - X_{\dot{u}} u_c) r + Y_v U \sin(\beta) - Y_v v_c] \quad (10.162)$$

Kinematic Controller

The {SF} frame plays the role of the virtual target body axes and tracks the real craft. The error coordinates for control design purposes become s, e and $\tilde{\chi}_{\text{SF}} = \chi_{\text{SF}} - \chi_d$ which all should be driven to zero. The desired approach angle can be chosen as a function of e (Micaelli and Samson, 1993):

$$\chi_d(e) = -\chi_a \frac{e^{2ke-1}}{e^{2ke+1}} \quad (10.163)$$

where $k > 0$ and $0 < \chi_a < \pi/2$ satisfying $e\chi_d(e) \leq 0$ for all e .

An alternative approach is motivated by the LOS algorithm (10.74) (see Breivik and Fossen, 2004a, and Børhaug and Pettersen, 2006):

$$\chi_d(e) = \arctan\left(\frac{-e}{\Delta}\right) \quad (10.164)$$

where $\Delta > 0$ is a constant parameter. Again notice that $e\chi_d(e) \leq 0$ for all e .

Theorem 10.1 (Kinematic Path-Following Controller)

A feedback linearization controller for (10.144) (Lapierre and Soetanto, 2007)

$$r = \dot{\chi}_d - \dot{\beta} + \kappa \dot{s}_a - K_1 \tilde{\chi}_{\text{SF}} \quad (10.165)$$

$$\dot{s}_a = U \cos(\chi_{\text{SF}}) + K_2 s \quad (10.166)$$

where the yaw rate r and path tangential speed $U_d = \dot{s}_a$ are used as control variables, renders the equilibrium point $(s, e, \tilde{\chi}_{\text{SF}}) = (0, 0, 0)$ UGAS and ULES for $K_1 > 0$ and $K_2 > 0$.

Proof. Convergence and stability can be proven by noticing that the error dynamics forms a cascade of two systems. For the first system:

$$\dot{\tilde{\chi}}_{\text{SF}} + K_1 \tilde{\chi}_{\text{SF}} = 0 \quad (10.167)$$

Consequently, the angle $\chi_{SF} \rightarrow \chi_d$. For the second system in the cascade, consider the Lyapunov function candidate:

$$V = \frac{1}{2}(s^2 + e^2) > 0, \quad s \neq 0, e \neq 0 \quad (10.168)$$

The time derivative of V under the assumption that $\chi_{SF} = \chi_d$ is

$$\begin{aligned} \dot{V} &= s(U \cos(\chi_d) - (1 - \kappa e)\dot{s}_a) + e(U \sin(\chi_d) - \kappa s\dot{s}_a) \\ &= sU \cos(\chi_d) + eU \sin(\chi_d) - s(U \cos(\chi_d) + K_2 s) \\ &= -K_2 s^2 + eU \sin(\chi_d) \end{aligned}$$

Exploiting the fact that the desired course angle given by (10.164) satisfies

$$\sin(\chi_d) = \frac{-e}{\sqrt{e^2 + \Delta^2}} \quad (10.169)$$

finally gives

$$\begin{aligned} \dot{V} &= -K_2 s^2 - \frac{U}{\sqrt{e^2 + \Delta^2}} e^2 \\ &< 0, \quad s \neq 0, e \neq 0 \end{aligned} \quad (10.170)$$

for $\Delta > 0$ and $U > 0$. Since the LFC is positive definite and radially unbounded, while its derivative with respect to time is negative, standard Lyapunov arguments for cascaded systems proves that the equilibrium point $(s, e, \tilde{\chi}_{SF}) = (0, 0, 0)$ is UGAS. In addition, the Jacobian of the error dynamics about the equilibrium point has strictly negative eigenvalues, which proves ULES.

Remark 10.3

A differential equation for the path variable ϖ can be derived by considering the path curvature $\kappa(\varpi)$ given by

$$\kappa(\varpi) = \frac{|x'_d y''_d - y'_d x''_d|}{\sqrt{(x'_d)^2 + (y'_d)^2}} \quad (10.171)$$

where $x_d = x_d(\varpi)$ and $y_d = y_d(\varpi)$. The arc length s_a satisfies

$$ds_a^2 = dx^2 + dy^2 \quad (10.172)$$

and by dividing by $d\varpi^2$, this can be rewritten as

$$d\varpi = \frac{1}{\sqrt{x'_d(\varpi)^2 + y'_d(\varpi)^2}} ds_a \quad (10.173)$$

Hence, from (10.166) it follows that

$$\dot{\varpi} = \frac{U \cos(\chi_{\text{SF}}) + K_2 s}{\sqrt{x'_d(\varpi)^2 + y'_d(\varpi)^2}} \quad (10.174)$$

Implementation Aspects

When implementing the kinematic controller (10.165)–(10.166), an expression for $\dot{\beta}$ must be computed from the sway dynamics (10.162). This expression depends on the model parameters. Consider the expression

$$\begin{aligned} r + \dot{\beta} &= r + \frac{1}{(m - Y_v)U \cos(\beta)} [(Y_r - (m - X_{\dot{u}})U \cos(\beta) - X_{\dot{u}}u_c))r + Y_v U \sin(\beta) - Y_v v_c] \\ &= \left(1 - \frac{(m - X_{\dot{u}})}{(m - Y_v)} + \frac{Y_r - X_{\dot{u}}u_c}{(m - Y_v)U \cos(\beta)}\right)r + \frac{Y_v}{(m - Y_v)} \left(\tan(\beta) - \frac{v_c}{U \cos(\beta)}\right) \\ &\approx \left(1 - \frac{(m - X_{\dot{u}})}{(m - Y_v)}\right)r + \frac{Y_v}{(m - Y_v)} \left(\tan(\beta) - \frac{v_c}{U \cos(\beta)}\right) \\ &= \dot{\chi}_d + \kappa \dot{s}_a - K_1 \tilde{\chi}_{\text{SF}} \end{aligned} \quad (10.175)$$

where the physical property

$$(m - Y_v)U \cos(\beta) \gg Y_r - X_{\dot{u}}u_c \quad (10.176)$$

has been exploited. Solving for $r = r_d$ gives the kinematic controller

$$r_d = \left(1 - \frac{(m - X_{\dot{u}})}{(m - Y_v)}\right)^{-1} \left[\dot{\chi}_d + \kappa U_d - K_1 \tilde{\chi}_{\text{SF}} - \frac{Y_v}{(m - Y_v)} \left(\tan(\beta) - \frac{v_c}{U \cos(\beta)}\right) \right] \quad (10.177)$$

$$U_d = U \cos(\chi_{\text{SF}}) + K_2 s \quad (10.178)$$

where the desired yaw rate is denoted r_d and the desired speed $U_d = \dot{s}_a$ is the path-tangential speed. The sideslip angle

$$\beta = \arcsin\left(\frac{v}{U}\right) \quad (10.179)$$

and current velocity v_c must be measured or estimated in a state observer. Alternatively, β and v_c can be treated as slowly varying parameters, which can be compensated for by adding integral action. This suggests that

$$r_d = \left[1 - \frac{(m - X_{\dot{u}})}{(m - Y_v)}\right]^{-1} \left[\dot{\chi}_d + \kappa U_d - 2\lambda \tilde{\chi}_{\text{SF}} - \lambda^2 \int_0^t \tilde{\chi}_{\text{SF}}(\tau) d\tau \right] \quad (10.180)$$

where $\lambda > 0$ is a constant parameter used to tune the bandwidth of the error system:

$$\dot{\tilde{\chi}}_{SF} + 2\lambda \tilde{\chi}_{SF} + \lambda^2 \int_0^t \tilde{\chi}_{SF}(\tau) d\tau = \frac{Y_v}{(m - Y_i)} \left(\tan(\beta) - \frac{v_c}{U \cos(\beta)} \right) \quad (10.181)$$

For a marine craft at constant course, the integral term will balance the forcing term in the steady state such that

$$\lambda^2 \int_0^t \tilde{\chi}_{SF}(\tau) d\tau = \frac{Y_v}{(m - Y_i)} \left(\tan(\beta) - \frac{v_c}{U \cos(\beta)} \right) \quad (10.182)$$

and $\tilde{\chi}_{SF} \rightarrow 0$.

11

Sensor and Navigation Systems

Conventional ship and underwater vehicle control systems are implemented with a state estimator for processing of the sensor and navigation data. The quality of the raw measurements are usually monitored and handled by a signal processing unit or a program for quality check and wild-point removal. The processed measurements are transmitted to the sensor and navigation computer which uses a state estimator capable of noise filtering, prediction and reconstruction of unmeasured states. The most famous algorithm is the Kalman filter which was introduced in the 1960s (Kalman, 1960; Kalman and Bucy, 1961). Later, other methods based on passivity and nonlinear theory have been developed.

Wave filtering is one of the most important issues to take into account when designing ship control systems (Fossen, 1994; Fossen and Perez, 2009). Environmental forces due to waves, wind and ocean currents are considered disturbances to the motion control system. These forces, which can be described in stochastic terms, are conceptually separated into low-frequency (LF) and wave-frequency (WF) components; see Figure 11.1.

Waves produce a pressure change on the hull surface, which in turn induces forces. These pressure-induced forces have an oscillatory component that depends linearly on the wave elevation. Hence, these forces have the same frequency as that of the waves and are therefore referred to as *wave-frequency forces*. Wave forces also have a component that depends nonlinearly on the wave elevation (Newman, 1977; Faltinsen, 1990). Nonlinear wave forces are due to the quadratic dependence of the pressure on the fluid-particle velocity induced by the passing of the waves. If, for example, two sinusoidal incident waves have different frequencies, then their quadratic relationship gives pressure forces with frequencies at both the sum and difference of the incident wave frequencies. They also contribute with zero-frequency or mean forces. Hence, the nonlinear wave forces have a wider frequency range than the incident waves. The mean wave forces cause the craft to drift. The forces with a frequency content at the difference of the wave frequencies can have LF content, which can lead to resonance in the horizontal motion of marine craft with mooring lines or under positioning control (Faltinsen, 1990). The high-frequency wave-pressure-induced forces, which have a frequency content at the sum of the wave frequencies, are normally too high to be considered in ship-motion control, but these forces can contribute to structural vibration in the hull. For further details about wave loads and their effects on ship motion, see Newman (1977) and Faltinsen (1990).

In low-to-medium sea states, the frequency of oscillations of the linear wave forces do not normally affect the operational performance of the craft. Hence, controlling only LF motion avoids correcting the motion for every single wave, which can result in unacceptable operational conditions for the propulsion system due to power consumption and potential wear of the actuators. Operations that require the control of only the LF motion include dynamic positioning, heading autopilots and thruster-assisted position

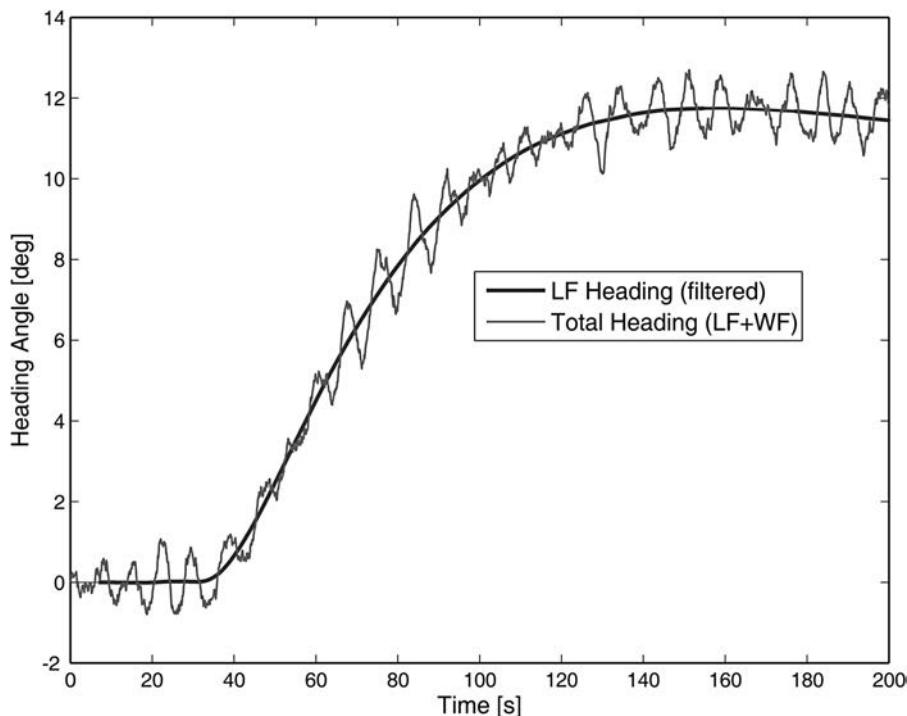


Figure 11.1 Separation of the total motion of a marine craft into LF and WF motion components.

mooring. Dynamic positioning refers to the use of the propulsion system to regulate the horizontal position and heading of the craft. In thruster-assisted position mooring, the propulsion system is used to reduce the mean loading on the mooring lines. Additional operations that require the control of only the LF motion include slow maneuvers that arise, for example, from following underwater remotely operated vehicles. Operations that require the control of only the WF motions include heave compensation for deploying loads on the sea floor (Perez and Steinmann, 2007) as well as ride control of passenger vessels, where reducing roll and pitch motion helps avoid motion sickness (Perez, 2005).

It is important that only the slowly varying forces are counteracted by the steering and propulsion systems; the oscillatory motion due to the waves (first-order wave-induced forces) should be prevented from entering the feedback loop. This is done by using *wave filtering* techniques (Balchen *et al.*, 1976). A wave filter is usually a model-based observer that separates the position and heading measurements into LF and WF position and heading signals; see Figure 11.1.

Definition 11.1 (Wave Filtering)

Wave filtering can be defined as the reconstruction of the LF motion components from noisy measurements of position, heading and in some cases velocity and acceleration by means of a state observer or a filter.

Remark 11.1

If a state observer is applied, estimates of the WF motion components (first-order wave-induced forces) can also be computed.

Wave filtering is crucial in ship motion control systems since the WF part of the motion should not be compensated for by the control system unless wave-induced vibration damping is an issue. This is the case for high-speed craft. If the WF part of the motion enters the feedback loop, this will cause unnecessary use of the actuators (thrust modulation) and reduce the tracking performance, which, again, results in increased fuel consumption.

In this chapter, model-based wave filtering and observer design using linear wave response models are discussed. This is one of the most important features of a high-precision ship control system. The best commercial autopilot and DP systems all have some kind of wave filtering in order to reduce wear and tear on the steering machine, as well as thrust modulation.

11.1 Low-Pass and Notch Filtering

Low-pass and notch filters can be used to reduce motions induced by ocean waves in the feedback loop. This assumes that the filters can be implemented in series, as shown in Figure 11.2. For wave periods in the interval $5 \text{ s} < T_0 < 20 \text{ s}$, the dominating wave frequency (modal frequency) f_0 of a wave spectrum will be in the range (see Section 8.2)

$$0.05 \text{ Hz} < f_0 < 0.2 \text{ Hz} \quad (11.1)$$

The circular frequency $\omega_0 = 2\pi f_0$ corresponding to periods $T_0 > 5 \text{ s}$ is

$$\omega_0 < 1.3 \text{ rad/s} \quad (11.2)$$

Waves within the frequency band (11.1) can be accurately described by first- and second-order wave theory. The first-order wave forces produce large *oscillations* about a *mean* wave force, which can be computed from second-order wave theory (see Figure 11.1). The mean wave (drift) force is slowly varying and is usually compensated for by using *integral action* in the control law, while *wave filtering* must be performed to remove first-order components from the feedback loop.

For instance, first-order wave forces around $f_0 = 0.1 \text{ Hz}$ can be close to or outside the control bandwidth of the marine craft depending of the craft considered. For a large oil tanker, the crossover frequency can be as low as 0.01 rad/s , as shown in Figure 11.3, while smaller vessels such as cargo ships and the Mariner class vessel are close to 0.05 rad/s .

A feedback control system will typically move the bandwidth of these vessels up to 0.1 rad/s , which still is below the wave spectrum shown in Figure 11.3. However, the wave forces will be inside the bandwidth of the servos and actuators of the craft. Hence, the wave forces must be filtered out before

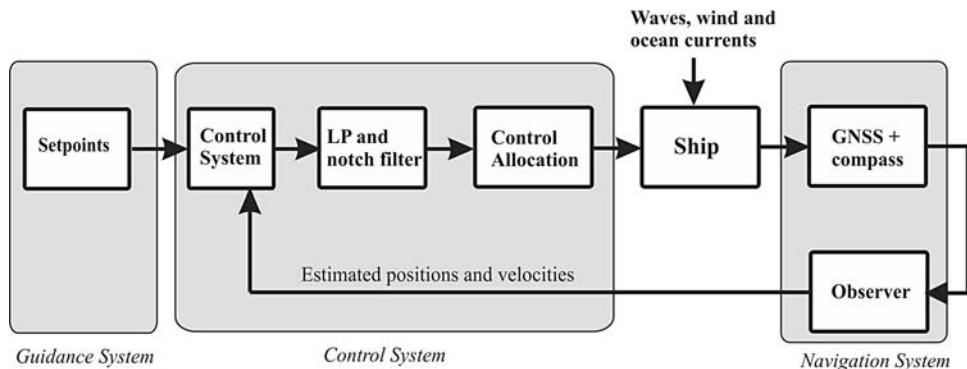


Figure 11.2 LP and notch filters in series with the control system.

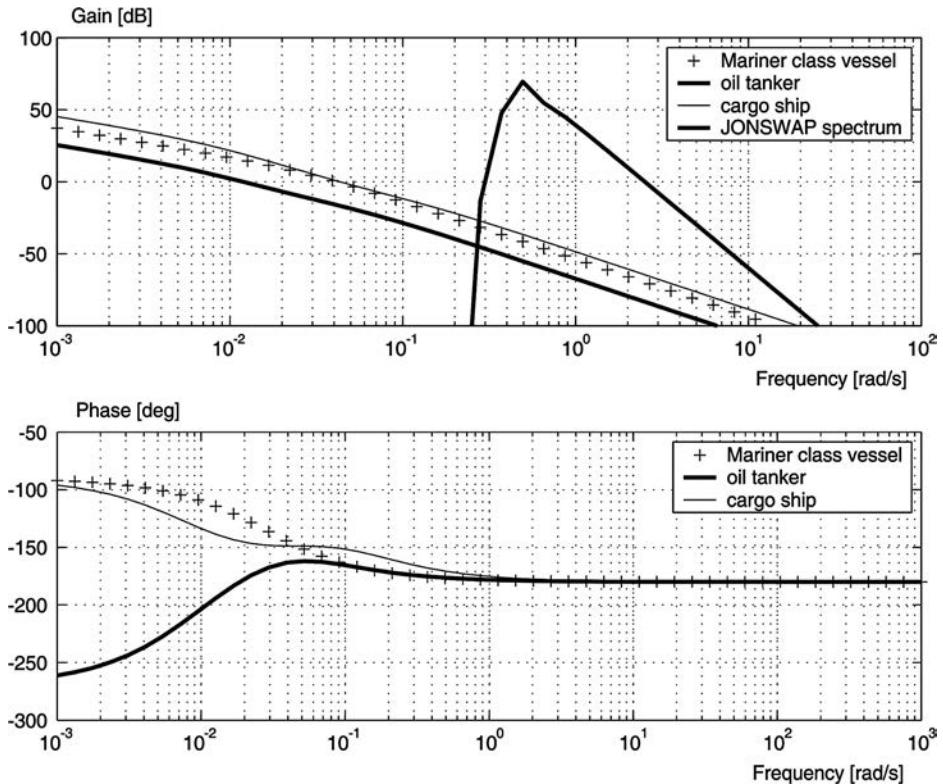


Figure 11.3 Bode plots showing $\psi/\delta(s)$ for three different vessels and the JONSWAP wave spectrum for $\omega_0 = 0.5$ rad/s and $H_s = 5$ m.

feedback is applied in order to avoid unnecessary control action. In other words, we do not want the rudder and thruster actuators of the ship to compensate for the first-order WF motion. This is usually referred to as *wave filtering*.

11.1.1 Low-Pass Filtering

For sea states where the WF motion is much higher than the bandwidth ω_b of the controller, a low-pass filter can be used to filter out the WF motions if ω_b satisfies

$$\omega_b \ll \omega_e \quad (11.3)$$

where

$$\omega_e = \left| \omega_0 - \omega_0^2 \frac{U}{g} \cos(\beta) \right| \quad (11.4)$$

is the frequency of encounter (see Figure 8.12). This is typically the case for large vessels such as oil tankers. In the autopilot case, the design objective can be understood by considering the *measurement equation*

$$y(s) = \underbrace{h_{\text{ship}}(s)\delta(s)}_{\psi(s)} + \underbrace{h_{\text{wave}}(s)w(s)}_{\psi_w(s)} \quad (11.5)$$

where $y(s)$ is the compass measurement, $w(s)$ is a zero-mean Gaussian white noise process and $\delta(s)$ is the rudder input. The signal $\psi(s)$ represents the LF motion, while $\psi_w(s)$ is the WF motion. Linear theory suggests that, see (8.112) and (7.46),

$$h_{\text{wave}}(s) = \frac{K_w s}{s^2 + 2\lambda\omega_0 s + \omega_0^2} \quad (11.6)$$

$$h_{\text{ship}}(s) = \frac{K(1 + T_3 s)}{s(1 + T_1 s)(1 + T_2 s)} \quad (11.7)$$

Feedback directly from y will therefore include the WF motion. For a large tanker, proper *wave filtering* can be obtained by using a low-pass filter to produce an estimate of $\psi(s)$ such that

$$\hat{\psi}(s) = h_{lp}(s)y(s) \quad (11.8)$$

Consequently, the feedback control law δ should be a function of $\hat{\psi}$ and not y in order to avoid first-order wave-induced rudder motions.

For instance, a first-order low-pass filter with time constant T_f can be designed according to

$$h_{lp}(s) = \frac{1}{1 + T_f s}, \quad \omega_b < \frac{1}{T_f} < \omega_e \quad (11.9)$$

This filter will suppress forces over the frequency $1/T_f$. This criterion is obviously hard to satisfy for smaller craft since ω_b can be close to or even larger than ω_e .

Higher-order low-pass filters can be designed by using a *Butterworth filter*, for instance. The n th-order Butterworth filter:

$$h_{lp}(s) = \frac{1}{p(s)} \quad (11.10)$$

is found by solving the Butterworth polynomial:

$$p(s)p(-s) = 1 + (s/j\omega_f)^{2n} \quad (11.11)$$

for $p(s)$. Here n denotes the order of the filter while ω_f is the cutoff frequency. For $n = 1, \dots, 4$ the solutions are

$$(n=1) \quad h_{lp}(s) = \frac{1}{1 + s/\omega_f}$$

$$(n=2) \quad h_{lp}(s) = \frac{\omega_f^2}{s^2 + 2\xi\omega_f s + \omega_f^2}; \quad \xi = \sin(45^\circ)$$

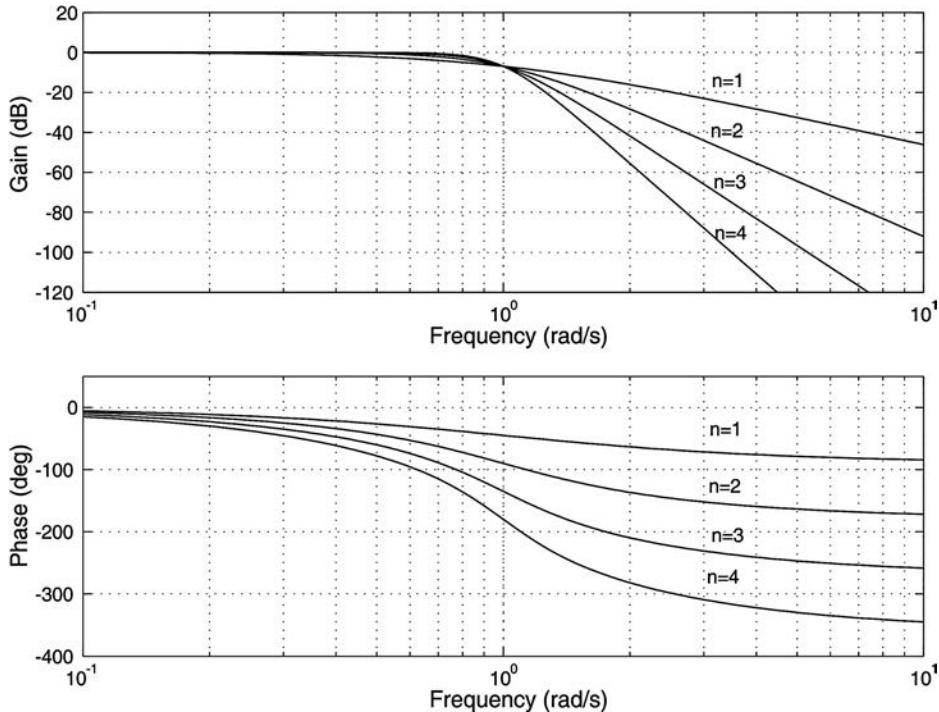


Figure 11.4 Bode plot showing the Butterworth filter for $n = 1, \dots, 4$ with cutoff frequency $\omega_f = 1.0$ rad/s.

$$(n = 3) \quad h_{lp}(s) = \frac{\omega_f^2}{s^2 + 2\zeta\omega_f s + \omega_f^2} \cdot \frac{1}{1 + s/\omega_f}; \quad \zeta = \sin(30^\circ)$$

$$(n = 4) \quad h_{lp}(s) = \prod_{i=1}^2 \frac{\omega_f^2}{s^2 + 2\zeta_i\omega_f s + \omega_f^2}; \quad \zeta_1 = \sin(22.5^\circ), \quad \zeta_2 = \sin(67.5^\circ)$$

A higher-order low-pass filter implies better disturbance suppression of the price of additional phase lags (see Figure 11.4).

11.1.2 Cascaded Low-Pass and Notch Filtering

For smaller craft the bandwidth of the controller ω_b can be close to or within the range $\omega_{\min} < \omega_e < \omega_{\max}$ of the wave spectrum. This problem can be handled by using a low-pass filter in cascade with a notch filter:

$$\hat{\psi}(s) = h_{lp}(s)h_n(s)y(s) \quad (11.12)$$

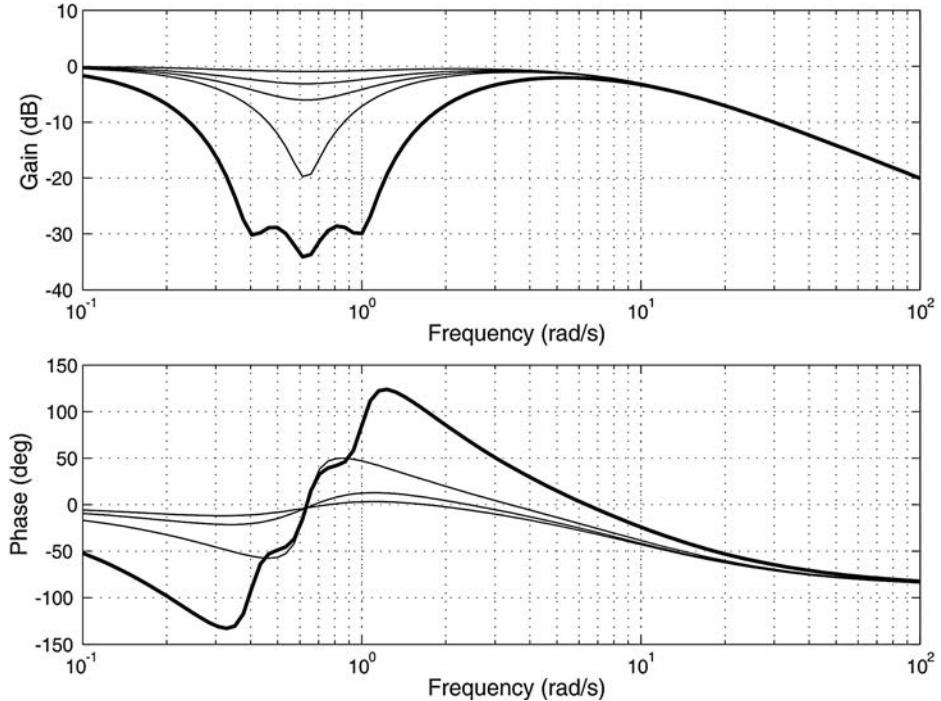


Figure 11.5 Bode plot showing the notch filter for $\zeta \in \{0.1, 0.5, 0.9\}$ and $\omega_0 = 0.63$ rad/s in cascade with a low-pass filter with time constant $T_f = 0.1$ s. The thick line represents three cascaded notch filters at $\omega_1 = 0.4$ rad/s, $\omega_2 = 0.63$ rad/s and $\omega_3 = 1.0$ rad/s.

where

$$h_n(s) = \frac{s^2 + 2\zeta\omega_n s + \omega_n^2}{(s + \omega_n)^2} \quad (11.13)$$

Here $0 < \zeta < 1$ is a design parameter used to control the magnitude of the notch while the notch frequency ω_n should be chosen equal to the peak frequency ω_0 of the spectrum for a marine craft at zero speed (dynamic positioning). The low-pass and notch filters are shown in Figure 11.5 for different values of ζ .

For a marine craft moving at forward speed U the optimal notch frequency will be

$$\omega_n = \omega_e \quad (11.14)$$

This frequency can be computed online by using a frequency tracker or adaptive filtering techniques. Since the estimate of ω_n can be poor and one single-notch filter only covers a small part of the actual frequency range of the wave spectrum, an alternative filter structure consisting of three cascaded notch filters with fixed center frequencies has been suggested; see Grimble and Johnson (1989):

$$h_n(s) = \prod_{i=1}^3 \frac{s^2 + 2\zeta\omega_i s + \omega_i^2}{(s + \omega_i)^2} \quad (11.15)$$

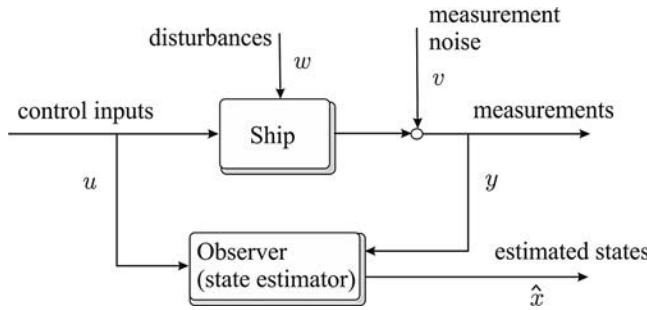


Figure 11.6 Block diagram showing the system model and the observer signal flow.

The center frequencies of the notch filters are typically chosen as $\omega_1 = 0.4$ rad/s, $\omega_2 = 0.63$ rad/s and $\omega_3 = 1.0$ rad/s. This is shown in Figure 11.5. Notice that additional phase lag is introduced when using a cascaded notch filter.

11.2 Fixed Gain Observer Design

The simplest state estimator is designed as a fixed gain observer where the ultimate goal of the observer is to reconstruct the unmeasured state vector \hat{x} from the measurements u and y of a dynamical system (see Figure 11.6). In order for this to succeed, the system must be *observable*.

11.2.1 Observability

Observability can be understood as a measure for how well internal states x of a system can be inferred by knowledge of its external outputs u and y . The *observability* and *controllability* of a system are mathematical duals. More specifically, a system is said to be observable if, for any possible sequence of state and control vectors, the current state can be determined in finite time using only the outputs. In other words, this means that from the outputs of the system it is possible to determine the behavior of the entire system. If a system is not observable, this means that the current values of some of its states cannot be determined through output sensors. This implies that their value is unknown to the controller and, consequently, that it will be unable to fulfil the control specifications referred to these outputs.

For time-invariant linear systems, a convenient observability test is given by the following definition:

Definition 11.2 (Observability)

Consider the linear time-invariant system:

$$\dot{x} = Ax + Bu \quad (11.16)$$

$$y = Hx \quad (11.17)$$

The state and output matrix (A, H) must satisfy the observability condition to ensure that the state x can be reconstructed from the output y and the input u . The observability condition requires that the matrix (Gelb et al., 1988)

$$\mathcal{O} = [H^\top \mid A^\top H^\top \mid \cdots \mid (A^\top)^{n-1} H^\top] \quad (11.18)$$

must be of full column rank such that (at least) a left inverse exists.

11.2.2 Luenberger Observer

Consider an *observable* linear time-invariant system:

$$\dot{x} = Ax + Bu + Ew \quad (11.19)$$

$$y = Hx + v \quad (11.20)$$

where w and v are zero-mean white noise terms. An observer copying the dynamics (11.19)–(11.20) under the assumption that the zero-mean noise terms w and v can be neglected is

$$\dot{\hat{x}} = A\hat{x} + Bu + \gamma(y, \hat{y}) \quad (11.21)$$

$$\hat{y} = H\hat{x} \quad (11.22)$$

where $\gamma(y, \hat{y})$ is an *injection term* to be constructed such that $\hat{x} \rightarrow x$ as $t \rightarrow \infty$. Note that the variables of a state observer are commonly denoted by a “hat” to distinguish them from the variables of the equations satisfied by the physical system.

The *Luenberger observer* is obtained by choosing the injection term γ as

$$\gamma(y, \hat{y}) = K\varepsilon, \quad \varepsilon = y - \hat{y} = H\tilde{x} \quad (11.23)$$

where K = constant is a matrix of observer gains.

Assume that $w = v = \mathbf{0}$. Defining the estimation error as $\tilde{x} := x - \hat{x}$ implies that the difference between (11.19) and (11.21) can be written

$$\dot{\tilde{x}} = A\tilde{x} - \gamma(y, \hat{y}) \quad (11.24)$$

For the Luenberger observer, the error dynamics becomes

$$\dot{\tilde{x}} = (A - KH)\tilde{x} \quad (11.25)$$

Asymptotical convergence of \tilde{x} to zero can be obtained for a constant K if the system (A, H) is observable, as explained in Section 11.2.1.

Matlab

If the observability matrix O is nonsingular, the poles of the error dynamics can be placed in the left half-plane by using the Matlab function:

```
K = place(A', H, p)'
```

where $p = [p_1, \dots, p_n]$ is a vector describing the desired locations of the observer error poles (must be distinct). Notice that both K and A are transposed, since the dual problem of the regulator problem is solved.

Examples 11.1–11.2 in Section 11.2.3 demonstrate how the Luenberger observer can be used in ship control when only compass measurements are available. Emphasis is placed on wave filtering and the estimation of the yaw rate.

11.2.3 Case Study: Luenberger Observer for Heading Autopilots using only Compass Measurements

An alternative to LP and notch filtering of wave-induced forces is to apply a state estimator (observer). A state estimator can be designed to separate the LF components of the motion from the noisy measurements by using a model of the ship and the WF motions. In fact, a model-based wave filter is well suited to separate the LF and WF motions from each other, even for marine craft, where the control bandwidth is close to or higher than the encounter frequency. It will now be shown how this can be done by considering a ship autopilot for heading control. It is assumed that the heading angle ψ is measured using a gyro or a magnetic compass while angular rate is left unmeasured, even though it is possible to use a gyro to measure the yaw rate $\dot{\psi}$.

Example 11.1 (Nomoto Ship Model Exposed to Wind, Waves and Ocean Currents)

Assume that a first-order Nomoto model describes the LF motion of the ship with sufficient accuracy:

$$\dot{\psi} = r \quad (11.26)$$

$$\dot{r} = -\frac{1}{T}r + \frac{K}{T}(\delta - b) + w_r \quad (11.27)$$

$$\dot{b} = -\frac{1}{T_b}b + w_b \quad (11.28)$$

where the rudder offset b is modeled as a first-order Markov process with $T_b \gg T$. In the limiting case, that is $T_b \rightarrow \infty$, this reduces to a Wiener process ($\dot{b} = w_b$). The rudder bias model is needed to counteract slowly varying moments on the ship due to wave drift forces, LF wind and ocean currents. Consequently, the bias term b ensures that $\delta = b$ gives $r = 0$ and $\psi = \text{constant}$ in the steady state. The linear wave model (8.122)–(8.123) can be used to model the wave response:

$$\dot{\xi}_w = \psi_w \quad (11.29)$$

$$\dot{\psi}_w = -\omega_0^2 \xi_w - 2\lambda\omega_0 \psi_w + K_w w_w \quad (11.30)$$

The process noise terms w_r , w_b and w_w are modeled as zero-mean Gaussian white noise processes. By combining the ship and wave models, the compass measurement equation can be expressed by the sum

$$y = \psi + \psi_w + v \quad (11.31)$$

where v represents zero-mean Gaussian measurement noise. Notice that neither the yaw rate r nor the wave states ξ_w and ψ_w are measured. The resulting SISO state-space model for $u = \delta$, $\mathbf{x} = [\xi_w, \psi_w, \psi, r, b]^\top$ and $\mathbf{w} = [w_w, w_r, w_b]^\top$ becomes

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{b}u + \mathbf{E}\mathbf{w} \quad (11.32)$$

$$y = \mathbf{h}^\top \mathbf{x} + v \quad (11.33)$$

where

$$\mathbf{A} = \left[\begin{array}{cc|ccc} 0 & 1 & 0 & 0 & 0 \\ -\omega_0^2 & -2\lambda\omega_0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -\frac{1}{T} & -\frac{K}{T} \\ 0 & 0 & 0 & 0 & -\frac{1}{T_b} \end{array} \right], \quad \mathbf{b} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \frac{K}{T} \\ 0 \end{bmatrix} \quad (11.34)$$

$$\mathbf{E} = \left[\begin{array}{c|cc} 0 & 0 & 0 \\ \underbrace{2\lambda\omega_0\sigma}_{K_w} & 0 & 0 \\ \hline 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{array} \right], \quad \mathbf{h}^\top = [0, 1, 1, 0, 0] \quad (11.35)$$

Matlab

The following example shows how the Luenberger observer gains of a ship autopilot system can be computed in Matlab.

Example 11.2 (Luenberger Observer Gains)

It is straightforward to see that the autopilot model with WF, wind and ocean current model (11.34)–(11.35) is observable from the input δ to the compass measurement y . Let $K = 1$, $T = 50$, $\lambda = 0.1$, $\omega_0 = 1$ and $T_b = 1000$; then

```
K = 1; T=50; lambda = 0.1; wo =1; Tb = 1000;

A = [ 0 1 0 0 0
      -wo*wo -2*lambda*wo 0 0 0
      0 0 0 1 0
      0 0 0 -1/T -K/T
      0 0 0 0 -1/Tb      ]
      h = [0,1,1,0,0] '
      n = rank(obsv(A,h'))
```

results in $n = 5$ corresponding to $\text{rank}(\mathcal{O}) = 5$. Hence, the system is observable according to Definition 11.2, implying that the states r , b , ψ_w and ξ_w can be reconstructed from a single measurement $y = \psi + \psi_w + v$ using a Luenberger observer:

$$\dot{\hat{x}} = \mathbf{A}\hat{x} + \mathbf{b}u + \mathbf{k}(y - \hat{y}) \quad (11.36)$$

$$\hat{y} = \mathbf{h}^\top \hat{x} \quad (11.37)$$

The filter gains can be computed by using pole placement in Matlab, for instance:

```
k = place(A',h,[p1,p2,p3,p4,p5])'
```

where p_1, p_2, p_3, p_4 and p_5 are the desired closed-loop poles of the error dynamics (11.25).

11.3 Kalman Filter Design

The Kalman filter is an efficient recursive filter that estimates the state of a linear or nonlinear dynamic system from a series of noisy measurements. It is widely used in sensor and navigation systems since it can reconstruct unmeasured states as well as remove white and colored noise from the state estimates. It is also possible to include wild-point removal capabilities. In cases of temporarily loss of measurements, the filter equations behave such as a predictor. As soon as new measurements are available, the predictor is corrected and updated online to give the minimum variance estimate. This feature is particularly useful when satellite signals are lost since the Kalman filter can predict the motion using only gyros and accelerometers. Inertial navigation systems and observers for inertial measurement units are discussed in Section 11.5.

Together with the linear quadratic regulator (LQR), the Kalman filter solves the linear quadratic Gaussian (LQG) control problem; see Section 13.1. This section summarizes the most useful results for the design of discrete-time and continuous-time Kalman filters for marine craft.

The key assumption when designing a Kalman filter is that the system model is observable. This is necessary in order to obtain convergence of the estimated states $\hat{\mathbf{x}}$ to \mathbf{x} . Moreover, if the system model is *observable* (see Definition 11.2), the state vector $\mathbf{x} \in \mathbb{R}^n$ can be reconstructed recursively through the measurement vector $\mathbf{y} \in \mathbb{R}^m$ and the control input vector $\mathbf{u} \in \mathbb{R}^p$ as shown in Figure 11.6.

11.3.1 Discrete-Time Kalman Filter

The discrete-time Kalman filter (Kalman, 1960) is defined in terms of the discretized system model:

$$\mathbf{x}(k+1) = \Phi \mathbf{x}(k) + \Delta \mathbf{u}(k) + \Gamma \mathbf{w}(k) \quad (11.38)$$

$$\mathbf{y}(k) = \mathbf{H} \mathbf{x}(k) + \mathbf{v}(k) \quad (11.39)$$

where

$$\Phi = \exp(Ah) \approx \mathbf{I} + Ah + \frac{1}{2}(Ah)^2 + \cdots + \frac{1}{N!}(Ah)^N \quad (11.40)$$

$$\Delta = A^{-1}(\Phi - \mathbf{I})B, \quad \Gamma = A^{-1}(\Phi - \mathbf{I})E \quad (11.41)$$

and h is the sampling time.

Matlab

The discretized system matrices can be computed as

```
[PHI, DELTA] = c2d(A, B, h)
[PHI, GAMMA] = c2d(A, E, h)
```

where $\text{PHI}=\Phi$, $\text{DELTA}=\Delta$ and $\text{GAMMA}=\Gamma$. Notice that Euler integration implies choosing $N = 1$ such that $\Phi(k) = \mathbf{I} + Ah$.

Table 11.1 Discrete-time Kalman filter

Design matrices	$\mathbf{Q}(k) = \mathbf{Q}^\top(k) > 0, \mathbf{R}(k) = \mathbf{R}^\top(k) > 0$ (usually constant)
Initial conditions	$\bar{\mathbf{x}}(0) = \mathbf{x}_0$ $\bar{\mathbf{P}}(0) = E[(\mathbf{x}(0) - \hat{\mathbf{x}}(0))(\mathbf{x}(0) - \hat{\mathbf{x}}(0))^\top] = \mathbf{P}_0$
Kalman gain matrix	$\mathbf{K}(k) = \bar{\mathbf{P}}(k)\mathbf{H}^\top(k) [\mathbf{H}(k)\bar{\mathbf{P}}(k)\mathbf{H}^\top(k) + \mathbf{R}(k)]^{-1}$
State estimate update	$\hat{\mathbf{x}}(k) = \bar{\mathbf{x}}(k) + \mathbf{K}(k) [y(k) - \mathbf{H}(k)\bar{\mathbf{x}}(k)]$
Error covariance update	$\hat{\mathbf{P}}(k) = [\mathbf{I} - \mathbf{K}(k)\mathbf{H}(k)] \bar{\mathbf{P}}(k) [\mathbf{I} - \mathbf{K}(k)\mathbf{H}(k)]^\top + \mathbf{K}(k)\mathbf{R}(k)\mathbf{K}^\top(k), \quad \hat{\mathbf{P}}(k) = \hat{\mathbf{P}}(k)^\top > 0$
State estimate propagation	$\bar{\mathbf{x}}(k+1) = \Phi(k)\hat{\mathbf{x}}(k) + \Delta(k)\mathbf{u}(k)$
Error covariance propagation	$\bar{\mathbf{P}}(k+1) = \Phi(k)\hat{\mathbf{P}}(k)\Phi^\top(k) + \Gamma(k)\mathbf{Q}(k)\Gamma^\top(k)$

The linear discrete-time Kalman filter algorithm is given in Table 11.1. This algorithm, however, requires that the state estimation error covariance matrix $\hat{\mathbf{P}}(k) = \hat{\mathbf{P}}(k)^\top > 0$ is computed online. Since the matrix is symmetrical, the number of differential equations will be $n(n + 1)/2$ for $\mathbf{P}(k) \in \mathbb{R}^{n \times n}$. In addition, there are n state estimates corresponding to $\hat{\mathbf{x}}(k)$.

11.3.2 Continuous-Time Kalman Filter

Consider the linear continuous-time system:

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} + \mathbf{E}\mathbf{w} \quad (11.42)$$

where the process noise \mathbf{w} is assumed to be a zero-mean Gaussian white noise process with covariance matrix $\mathbf{Q} = \mathbf{Q}^\top > 0$. In the one-dimensional case \mathbf{Q} corresponds to the squared *standard deviation* σ^2 . Furthermore, let the measurement equation (sensor system) be represented by

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{v} \quad (11.43)$$

where the measurement noise \mathbf{v} is assumed to be a zero-mean Gaussian white noise process with covariance matrix $\mathbf{R} = \mathbf{R}^\top > 0$.

If the system (11.42)–(11.43) is *observable* (see Definition 11.2), the state vector $\mathbf{x} \in \mathbb{R}^n$ can be reconstructed recursively through the measurement vector $\mathbf{y} \in \mathbb{R}^m$ and the control input vector $\mathbf{u} \in \mathbb{R}^p$ (see Figure 11.6). The continuous-time KF algorithms are given in Table 11.2.

Table 11.2 Continuous-time Kalman filter

Design matrices	$\mathbf{Q}(t) = \mathbf{Q}^\top(t) > 0, \mathbf{R}(t) = \mathbf{R}^\top(t) > 0$ (usually constant)
Initial conditions	$\hat{\mathbf{x}}(0) = \mathbf{x}_0$ $\mathbf{P}(0) = E[(\mathbf{x}(0) - \hat{\mathbf{x}}(0))(\mathbf{x}(0) - \hat{\mathbf{x}}(0))^\top] = \mathbf{P}_0$
Kalman gain matrix propagation	$\mathbf{K}(t) = \mathbf{P}(t)\mathbf{H}^\top(t)\mathbf{R}^{-1}(t)$
State estimate propagation	$\dot{\hat{\mathbf{x}}}(t) = \mathbf{A}(t)\hat{\mathbf{x}}(t) + \mathbf{B}(t)\mathbf{u}(t) + \mathbf{K}(t)[y(t) - \mathbf{H}(t)\hat{\mathbf{x}}(t)]$
Error covariance propagation	$\dot{\mathbf{P}}(t) = \mathbf{A}(t)\mathbf{P}(t) + \mathbf{P}(t)\mathbf{A}^\top(t) + \mathbf{E}(t)\mathbf{Q}(t)\mathbf{E}^\top(t) - \mathbf{P}(t)\mathbf{H}^\top(t)\mathbf{R}^{-1}(t)\mathbf{H}(t)\mathbf{P}(t), \quad \mathbf{P}(t) = \mathbf{P}^\top(t) > 0$

In the linear case it is computationally advantageous to use the steady-state solution of the KF. This filter will in fact have the same structure as the fixed-gain observers of Section 11.2.3. The only difference is the method for computation of the filter gain matrix.

Continuous-Time Steady-State Kalman Filter

A frequently used simplification of the continuous-time Kalman filter is the steady-state solution obtained for the linear time-invariant (LTI) system:

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} + \mathbf{E}\mathbf{w} \quad (11.44)$$

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{v} \quad (11.45)$$

where \mathbf{w} and \mathbf{v} are zero-mean Gaussian white noise processes. The steady-state Kalman filter is given by

$$\dot{\hat{\mathbf{x}}} = \mathbf{A}\hat{\mathbf{x}} + \mathbf{B}\mathbf{u} + \mathbf{K}_\infty(\mathbf{y} - \mathbf{H}\hat{\mathbf{x}}) \quad (11.46)$$

$$\mathbf{K}_\infty = \mathbf{P}_\infty \mathbf{H}^\top \mathbf{R}^{-1} \quad (11.47)$$

where $\mathbf{P}_\infty = \mathbf{P}_\infty^\top > 0$ is the positive definite solution of the *algebraic matrix Riccati equation*

$$\mathbf{A}\mathbf{P}_\infty + \mathbf{P}_\infty\mathbf{A}^\top + \mathbf{E}\mathbf{Q}\mathbf{E}^\top - \mathbf{P}_\infty\mathbf{H}^\top \mathbf{R}^{-1}\mathbf{H}\mathbf{P}_\infty = \mathbf{0} \quad (11.48)$$

11.3.3 Extended Kalman Filter

The Kalman filter can also be applied to nonlinear systems in the form

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \mathbf{B}\mathbf{u} + \mathbf{E}\mathbf{w} \quad (11.49)$$

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{v} \quad (11.50)$$

where $\mathbf{f}(\mathbf{x})$ is a nonlinear vector field. For this system, the state vector can be estimated using the discrete-time extended Kalman filter (EKF) algorithm of Table 11.3.

The discrete-time quantities $\mathcal{F}(\hat{\mathbf{x}}(k), \mathbf{u}(k))$, $\Phi(k)$ and $\Gamma(k)$ in Table 11.3 can be found by using *forward Euler* integration, for instance. Moreover,

$$\mathcal{F}(\hat{\mathbf{x}}(k), \mathbf{u}(k)) \approx \hat{\mathbf{x}}(k) + h[\mathbf{f}(\hat{\mathbf{x}}(k)) + \mathbf{B}\mathbf{u}(k)] \quad (11.51)$$

$$\Phi(k) \approx \mathbf{I} + h \left. \frac{\partial \mathbf{f}(\mathbf{x}(k), \mathbf{u}(k))}{\partial \mathbf{x}(k)} \right|_{\mathbf{x}(k)=\hat{\mathbf{x}}(k)} \quad (11.52)$$

$$\Gamma(k) \approx h\mathbf{E} \quad (11.53)$$

Table 11.3 Discrete-time extended Kalman filter (EKF)

Design matrices	$\mathbf{Q}(k) = \mathbf{Q}^\top(k) > 0, \mathbf{R}(k) = \mathbf{R}^\top(k) > 0$ (usually constant)
Initial conditions	$\bar{\mathbf{x}}(0) = \mathbf{x}_0$ $\bar{\mathbf{P}}(0) = E[(\mathbf{x}(0) - \hat{\mathbf{x}}(0))(\mathbf{x}(0) - \hat{\mathbf{x}}(0))^\top] = \mathbf{P}_0$
Kalman gain matrix	$\mathbf{K}(k) = \bar{\mathbf{P}}(k)\mathbf{H}^\top(k) [\mathbf{H}(k)\bar{\mathbf{P}}(k)\mathbf{H}^\top(k) + \mathbf{R}(k)]^{-1}$
State estimate update	$\hat{\mathbf{x}}(k) = \bar{\mathbf{x}}(k) + \mathbf{K}(k) [\mathbf{y}(k) - \mathbf{H}(k)\bar{\mathbf{x}}(k)]$
Error covariance update	$\hat{\mathbf{P}}(k) = [\mathbf{I} - \mathbf{K}(k)\mathbf{H}(k)] \bar{\mathbf{P}}(k) [\mathbf{I} - \mathbf{K}(k)\mathbf{H}(k)]^\top$ $+ \mathbf{K}(k)\mathbf{R}(k)\mathbf{K}^\top(k), \quad \hat{\mathbf{P}}(k) = \hat{\mathbf{P}}(k)^\top > 0$
State estimate propagation	$\bar{\mathbf{x}}(k+1) = \mathcal{F}(\hat{\mathbf{x}}(k), \mathbf{u}(k))$
Error covariance propagation	$\bar{\mathbf{P}}(k+1) = \Phi(k)\hat{\mathbf{P}}(k)\Phi^\top(k) + \Gamma(k)\mathbf{Q}(k)\Gamma^\top(k)$

where $h > 0$ is the sampling time. The EKF has been widely used in many applications. In Jouffroy and Fossen (2010) it has been shown that the continuous-time EKF is incremental GES under the assumption that the \mathbf{P} matrix of the Riccati equation is uniformly positive definite and upper bounded, that is

$$p_{\min}\mathbf{I} \leq \mathbf{P}(t) \leq p_{\max}\mathbf{I} \quad (11.54)$$

for two strictly positive constants p_{\min} and p_{\max} . This guarantees that the estimates converge exponentially to the actual states.

11.3.4 Corrector–Predictor Representation for Nonlinear Observers

When implementing nonlinear observers, the *corrector–predictor representation* of the discrete-time EKF can be used to handle effectively slow measurement rates, multiple measurement rates and *dead-reckoning*. Consider the continuous-time nonlinear observer:

$$\dot{\hat{\mathbf{x}}} = \mathbf{f}(\hat{\mathbf{x}}, \mathbf{u}) + \gamma(\mathbf{y}, \hat{\mathbf{y}}) \quad (11.55)$$

with the linear injection term

$$\gamma(\mathbf{y}, \hat{\mathbf{y}}) = \mathbf{K}(\mathbf{y} - \hat{\mathbf{y}}) \quad (11.56)$$

The discrete-time corrector–predictor formulation for the nonlinear system model (11.55) in terms of Euler integration becomes

$$\begin{aligned} \text{Corrector} \quad \hat{\mathbf{x}}(k) &= \bar{\mathbf{x}}(k) + \mathbf{K}_d [\mathbf{y}(k) - \bar{\mathbf{y}}(k)] \\ \text{Predictor} \quad \bar{\mathbf{x}}(k+1) &= \bar{\mathbf{x}}(k) + h\mathbf{f}(\hat{\mathbf{x}}(k), \mathbf{u}(k)) \end{aligned} \quad (11.57)$$

where $\mathbf{K}_d = h\mathbf{K}$ and h is the sampling time. At each time t_k a measurement $\mathbf{y}(k)$ is available, the corrector updates the state $\bar{\mathbf{x}}(k)$ to $\hat{\mathbf{x}}(k)$. The updated state is used by the predictor to predict the states at time t_{k+1} based on the nonlinear system model $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u})$.

Example 11.3 (Corrector–Predictor for Ship Navigation using Two Measurement Rates)

Consider a ship navigation system where the predictor runs at 100 Hz. The IMU and GPS measurements, y_{IMU} and y_{GPS} , are received at 100 Hz (same as the sampling time) and 10 Hz, respectively.

The corrector-predictor representation can be modified to handle two measurement frequencies by modifying the gain \mathbf{K}_d or the measurement vector \mathbf{y} according to

$h = 0.01$ sampling time
 $\text{GPS} = 10$ counter for GPS measurements
 $\bar{\mathbf{x}} = \mathbf{x}_0$ initial state vector

while estimating

Method A	Method B
$\mathbf{y}_{\text{IMU}} = \text{measurement}$	$\mathbf{y}_{\text{IMU}} = \text{measurement}$
$\mathbf{y}_{\text{GPS}} = \text{measurement}$	
$\mathbf{K}_d = [hk_{\text{IMU}}, 0]^\top$	$\mathbf{K}_d = [hk_{\text{IMU}}, 10hk_{\text{GPS}}]^\top$
if GPS = 10	if GPS = 10
$\mathbf{K}_d = [hk_{\text{IMU}}, 10hk_{\text{GPS}}]^\top$	$\mathbf{y}_{\text{GPS}} = \text{measurement}$
GPS = 0	GPS = 0
end	end
if dead-reckoning (no updates)	if dead-reckoning (no updates)
$\mathbf{K}_d = [0, 0]^\top$	$\mathbf{K}_d = [0, 0]^\top$
end	end
$\mathbf{y} = [\mathbf{y}_{\text{IMU}}^\top, \mathbf{y}_{\text{GPS}}^\top]^\top$	$\mathbf{y} = [\mathbf{y}_{\text{IMU}}^\top, \mathbf{y}_{\text{GPS}}^\top]^\top$
$\hat{\mathbf{x}} = \bar{\mathbf{x}} + \mathbf{K}_d [\mathbf{y} - \mathbf{H}\bar{\mathbf{x}}]$	$\hat{\mathbf{x}} = \bar{\mathbf{x}} + \mathbf{K}_d [\mathbf{y} - \mathbf{H}\bar{\mathbf{x}}]$
$\mathbf{u} = \text{control system (optionally)}$	$\mathbf{u} = \text{control system (optionally)}$
$\bar{\mathbf{x}} = \bar{\mathbf{x}} + h\mathbf{f}(\hat{\mathbf{x}}, \mathbf{u})$	$\bar{\mathbf{x}} = \bar{\mathbf{x}} + h\mathbf{f}(\hat{\mathbf{x}}, \mathbf{u})$
GPS = GPS + 1	GPS = GPS + 1
end	

Dead-reckoning refers to the case where there are no updates (signal loss) for a period of time. During sensor failures, the best thing to do is to trust the model without any updates. Hence, the corrector is bypassed by setting $\hat{\mathbf{x}}(k) = \bar{\mathbf{x}}(k)$ and prediction is based on the system model only:

$$\hat{\mathbf{x}}(k+1) = \bar{\mathbf{x}}(k) + h\mathbf{f}(\bar{\mathbf{x}}(k), \mathbf{u}(k)) \quad (11.58)$$

Notice that observers with linear injection terms can use both methods while observers with nonlinear injection terms only can use the second method.

11.3.5 Case Study: Kalman Filter for Heading Autopilots using only Compass Measurements

This section explains how observers and wave filters for heading autopilots can be designed.

Heading Sensors

The main sensor components for a heading controlled marine craft are:

- Magnetic and/or gyroscopic compasses measuring ψ
- Yaw rate gyro measuring r

In many commercial systems only the compass is used for feedback control since the yaw rate can be estimated quite well by a state estimator.

A compass is the primary device for direction-finding on the surface of the Earth. Compasses may operate on magnetic or gyroscopic principles or by determining the direction of the Sun or a star. The discussions will be restricted to magnetic and gyroscopic compasses, since these are the primary devices onboard commercial ships and rigs.

The magnetic compass is an old Chinese invention, which probably dates back to 100 AD. Knowledge of the compass as a directional device came to western Europe sometime in the 12th century and it is today a standard unit in all commercial and navy ships.

A magnetic compass is in fact an extremely simple device (as opposed to a gyroscopic compass). It consists of a small, lightweight magnet balanced on a nearly frictionless pivot point. The magnet is generally called a needle. The magnetic field inside the Earth has its south end at the North Pole and opposite. Hence, the North end of the compass needle points towards the North Pole (opposite magnets attract). The magnetic field of the Earth is, however, not perfectly aligned along the Earth's rotational axis. It is skewed slightly off center. This skew or bias is called the *declination* and it must be compensated for. It is therefore common to indicate what the declination is on navigational maps. Sensitivity to magnetic variations and declination cause problems in ship navigation. These problems were overcome after the introduction of the gyroscopic compass.

The first recorded construction of the gyroscope is usually credited to *C. A. Bohnenberger* in 1810 while the first electrically driven gyroscope was demonstrated in 1890 by *G. M. Hopkins* (see Allensworth, 1999; Bennet, 1979). A gyroscope is a disk mounted on a base in such a way that the disk can spin freely on its x and y axes; that is the disk will remain in a fixed position in whatever directions the base is moved. A properly mounted gyroscope will always turn to match its plane of rotation with that of the Earth, just as a magnetic compass turns to match the Earth's magnetic field.

The large variations in the magnetic character of ships caused by electrical machinery and weapon systems made the construction of accurate declination or deviation tables for the magnetic compass very difficult. In parallel works, *Dr H. Anschütz* of Germany and *Elmer Sperry* of the USA worked on a practical application of *Hopkins'* gyroscope. In 1908 Anschütz patented the first North-seeking gyrocompass, while Elmer Sperry was granted a patent for his ballistic compass, which includes vertical damping, three years later.

In 1910, when the Anschütz gyro compass appeared, the problem with magnetic variations in ship navigation was eliminated. However, this compass proved to be quite unsatisfactory during rolling of the ship, since it produced an "intercardinal rolling error". Therefore in 1912 Anschütz redesigned the compass to overcome this defect. One year later, the Sperry compass entered the market and it became a serious competitor with the Anschütz. Today gyroscopic compasses are produced by a large number of companies for both commercial and navy ships.

System Model for Heading Autopilot Observer Design

As in the case of positioning, we consider the first-order wave-induced motion as an output disturbance. Hence the measured yaw angle can be decomposed into

$$y = \psi + \psi_w + v \quad (11.59)$$

where ψ is the response due to the control action and LF disturbance, ψ_w represents the first-order wave-induced motion and v is zero-mean Gaussian white measurement noise introduced by the compass. To estimate ψ and r from y , and at the same time obtain wave filtering, one can use a WF

model to predict the wave motions ψ_w . The main tool for this is a linear time-invariant Kalman filter based on

$$\dot{\xi}_w = \psi_w \quad (11.60)$$

$$\dot{\psi}_w = -\omega_0^2 \xi_w - 2\lambda\omega_0 \psi_w + w_1 \quad (11.61)$$

where λ and ω_0 are the relative damping ratio and peak frequency of the filter used to represent the wave-induced yaw motion. The yaw dynamics of a marine craft is given by the Nomoto model (see Section 7.2):

$$\dot{\psi} = r \quad (11.62)$$

$$\dot{r} = -\frac{1}{T}r + \frac{1}{m}(\tau_{\text{wind}} + \tau_N) + b + w_2 \quad (11.63)$$

$$\dot{b} = w_3 \quad (11.64)$$

where b is a bias term and w_1 , w_2 and w_3 are zero-mean Gaussian white noise processes. The constant $m = I_z - N_r$ is introduced for convenience such that the rudder angle δ generates a yaw moment τ_N given by

$$\tau_N = m \frac{K}{T} \delta = N_\delta \delta \quad (11.65)$$

while τ_{wind} represents an optional term for wind feedforward. Notice that neither the yaw rate r nor the wave states ξ_w and ψ_w are measured. The resulting state-space model is

$$\dot{x} = Ax + bu + Ew \quad (11.66)$$

$$y = h^\top x + v \quad (11.67)$$

where

$$x = [\xi_w, \psi_w, \psi, r, b]^\top \quad (11.68)$$

$$u = \tau_{\text{wind}} + \tau_N \quad (11.69)$$

$$w = [w_1, w_2, w_3]^\top \quad (11.70)$$

and

$$A = \left[\begin{array}{cc|ccc} 0 & 1 & 0 & 0 & 0 \\ -\omega_0^2 & -2\lambda\omega_0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1/T & 1 \\ 0 & 0 & 0 & 0 & 0 \end{array} \right], \quad b = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1/m \\ 0 \end{bmatrix} \quad (11.71)$$

$$E = \left[\begin{array}{c|cc} 0 & 0 & 0 \\ 1 & 0 & 0 \\ \hline 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{array} \right], \quad h^\top = [0, 1, 1, 0, 0] \quad (11.72)$$

In order to implement a Kalman filter for a heading autopilot, the system model can be used in a discrete- or continuous-time filter, as presented in Sections 11.3.1–11.3.2. The main problem in the realization

of the state estimator is that the parameters T, m, ω_0 and λ are uncertain. The parameters T and m can be estimated from tests performed in calm water while the parameters ω_0 and λ of the first-order WF model and the covariance of the driving noise w_1 can be estimated from maneuvering trials or parameter estimation.

Holzhüter (1992) claims that the damping coefficient in the wave model can be chosen rather arbitrarily as long as it is low (typically $\lambda = 0.01\text{--}0.1$), whereas the wave frequency ω_0 can be treated as a tunable or gain scheduled parameter. In some cases it can be advantageous to estimate ω_0 online by applying a frequency tracker or adaptive control theory (Strand and Fossen, 1999).

Matlab

The following example illustrates how the Kalman filter gains can be computed in Matlab for a ship exposed to waves.

Example 11.4 (Continuous-Time Steady-State KF for Ship Autopilots)

For the ship-wave system (11.66)–(11.67), the SISO continuous-time state estimator takes the form

$$\dot{\hat{x}} = A\hat{x} + bu + k_\infty(y - h^\top x) \quad (11.73)$$

where the Kalman filter gain is

$$k_\infty = \frac{1}{r} P_\infty h \quad (11.74)$$

The covariance matrix $P_\infty = P_\infty^\top > 0$ is given by the ARE:

$$AP_\infty + P_\infty A^\top + EQE^\top - \frac{1}{r} P_\infty hh^\top P_\infty = \mathbf{0} \quad (11.75)$$

The KF gain k_∞ is computed in Matlab as

$$\begin{aligned} R &= r \\ Q &= \text{diag}\{q_{11}, q_{22}, q_{33}\} \\ [k, P] &= \text{lqe}(A, E, h, Q, R) \end{aligned}$$

where the tuning of the filter is done by choosing the four design parameters r , q_{11} , q_{22} and q_{33} . The first of these, r , represents the compass covariance, which can be computed by logging a time series $\psi(t)$ of the compass at a constant heading. Hence, the Matlab command

$$r = \text{cov}(\psi)$$

gives a good estimate of the measurement noise. The disadvantage with the KF approach is that information about the process noise w_1, w_2 and w_3 represented by the weights q_{11}, q_{22} and q_{33} are necessary. These three quantities are usually found by trial and error. The variance of the process and measurement noise will vary with each sea state, implying that several sets of KF gains must be computed.

Example 11.5 (Kalman-Filter-Based Wave Filter for the Mariner Class Vessel)

To illustrate the performance of Kalman filter-based wave filtering, we consider the case study in Fossen and Perez (2009) of an autopilot application taken from the Marine Systems Simulator (MSS, 2011). This

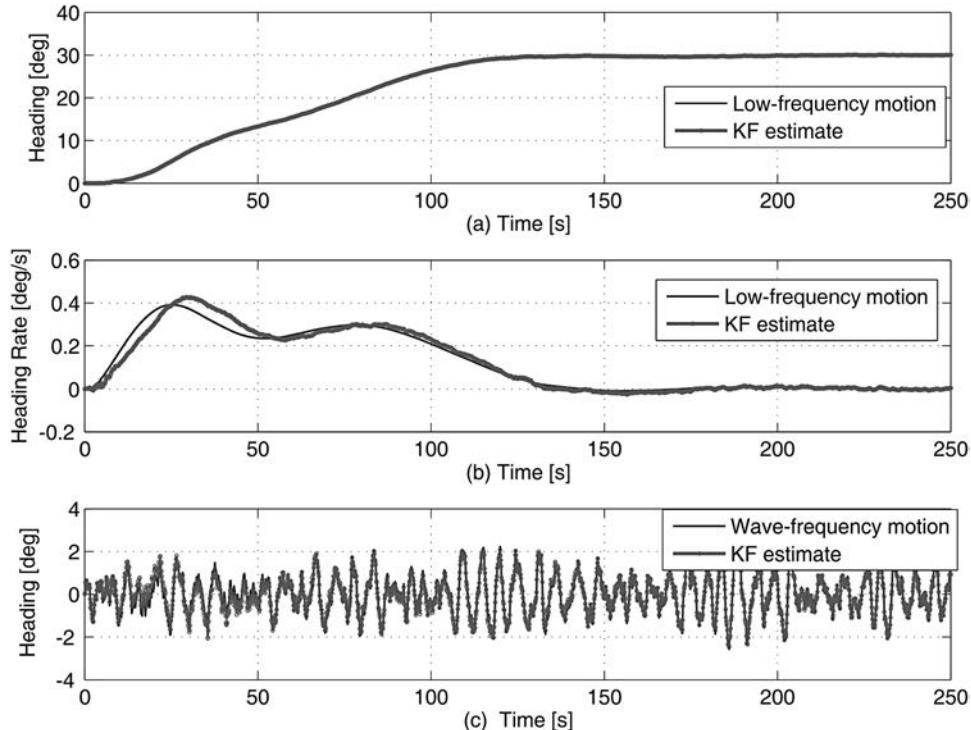


Figure 11.7 Kalman filter performance for a heading autopilot designed for the Mariner class cargo ship: (a) shows the true LF heading ψ and estimate $\hat{\psi}$, (b) shows the true LF heading rate r and estimate \hat{r} and (c) shows the WF component of the heading ψ_w and its estimate $\hat{\psi}_w$.

simulation package implemented in Matlab-Simulink provides models of marine craft and a library of Simulink blocks for heading autopilot control system design. A Simulink block for Kalman filter-based wave filtering is also included.

The marine craft considered is a 160 m Mariner class vessel with a nominal service speed of 15 knots, or 7.7 m/s. The parameters of a complete and validated nonlinear model for the Mariner class vessel are given in Fossen (1994). From the step tests performed on the nonlinear model, a first-order Nomoto model is identified with the parameters $K = 0.185 \text{ s}^{-1}$ and $T = 107.3 \text{ s}$. Based on the time constant, a sampling period of 0.5 s is chosen for the implementation of the Kalman filter. The standard deviation of the noise of the compass is 0.5 degrees. From a record of heading motion while the rudder is kept constant, the parameters of the first-order wave-induced motion model are estimated, namely $\zeta = 0.1$, $\omega_0 = 1.2 \text{ rad/s}$, and the standard deviation of the noise driving the filter is $\sigma_{w1} = \sqrt{300} \text{ rad/s}$.

Figure 11.7 demonstrates the performance of the Kalman filter. The two upper plots show the true LF heading angle and rate together with the Kalman filter estimates while the lower plot shows the first-order wave-induced heading angle and its estimate.

11.3.6 Case Study: Kalman Filter for Dynamic Positioning Systems using GNSS and Compass Measurements

Kalman filtering (or *optimal state estimation* in the sense of minimum variance) allows the user to estimate the state x of a dynamic system from a noise-contaminated input–output pair (u, y) . The interested reader

is advised to consult Brown and Hwang (1998) or Gelb *et al.* (1988) for details on Kalman filter design. Applications specific to the field of guidance and control can be found in Lin (1992).

Dynamic positioning (DP) systems have been commercially available for marine craft since the 1960s. The first DP systems were designed using conventional PID controllers in cascade with low-pass and/or notch filters to suppress the wave-induced motion components. From the middle of the 1970s more advanced filtering techniques were available thanks to the Kalman filter (Kalman, 1960). This motivated Balchen and coauthors to develop optimal wave filtering and state estimation; see Balchen *et al.* (1976, 1980a, 1980b). KF-based wave filtering has also been discussed by Grimble *et al.* (1980a, 1980b), Fung and Grimble (1981, 1983), Fotakis *et al.* (1982), Sælid and Jenssen (1983), Sælid *et al.* (1983), Reid *et al.* (1984), Holzhüter and Strauch (1987), Holzhüter (1992), Sørensen *et al.* (1995, 1996, 2000), Fossen and Strand (2000) and Fossen and Perez (2009).

In this section, the *Kalman filter* is presented for DP applications. Both observers include wave filtering, bias state estimation and reconstruction of the LF motion components, and estimates of the nonmeasured body velocities are considered. Positioning feedback systems are described more closely in Sections 12.2.10 and 13.1.6. Before observer design is discussed, a general introduction to navigation systems is given.

Global Navigation Satellite Systems (GNSS)

Several Global Navigation Satellite Systems (GNSS) provide autonomous geospatial positioning with global coverage. The United States *NAVSTAR Global Positioning System* (GPS) has been fully operative since 1995 (see Hofmann-Wellenhof *et al.*, 1994; Parkinson and Spilker, 1995). In addition to GPS, the Russian *GLObal'naya NAVigatsionnaya Sputnikovaya Sistema* (GLONASS) has been restored; see Kayton and Fried (1997) and Leick (1995), for instance. A more recent and more accurate system is the European Union's *GALILEO* positioning system, which will be complementary to GPS and GLONASS. For this purpose, integrated GNSS receivers capable of combining signals from one or more systems can be used. This also improves redundancy in marine control systems. The GNSS measurements are usually used in a motion control system that operates in the three planar degrees of freedom, namely *surge* (forward motion), *sway* (transverse motion) and *yaw* (rotation about the vertical axis, also called heading). The position of the marine craft is normally measured by differential GNSS, while the heading is measured by a gyrocompass. Additional types of sensors are usually available to ensure reliability of the positioning system, namely inertial measurement units, hydro acoustic position sensors, taut wires and laser sensors.

- *Differential and Augmented GNSS*: The main idea of a *differential* GNSS system is that a fixed receiver located, for example, *on shore* with a known position is used to calculate the GNSS position errors. The position errors are then transmitted to the GNSS receiver on board the ship and used as corrections to the actual ship position. In a *differential* GNSS the horizontal positioning errors are squeezed down to less than 1 m, which is the typical accuracy of a ship positioning system today (Hofmann-Wellenhof *et al.*, 1994).
- *Carrier Differential GNSS*: A GNSS receiver in lock is able to track the phase shift of the carrier and output the fractional phase measurement at each epoch. However, the overall phase measurement contains an unknown number of carrier cycles. This is called the integer ambiguity (N). This ambiguity exists because the receiver merely begins counting carrier cycles from the time a satellite signal is placed in an active track. For GPS, the precision of the phase measurement is about 0.01 cycles ($\approx 0.01 \times 19 \text{ cm} = 1.9 \text{ mm}$), and if N is determined, it allows for highly accurate position measurements. Ambiguity resolution is a very active research area, and there are several receivers known as *real-time kinematic* (RTK) receivers on the market today that utilize carrier measurements to achieve position accuracy in the order of a few centimeters. These position measurements are, however, not as robust as GPS and DGPS.

System Model for Dynamic Positioning Observer Design

In Section 7.3 it was shown that the 3 DOF nonlinear model for DP can be written as

$$\dot{\eta} = \mathbf{R}(\psi)\mathbf{v} \quad (11.76)$$

$$\mathbf{M}\dot{\mathbf{v}} + \mathbf{C}_{RB}(\mathbf{v})\mathbf{v} + \mathbf{D} \exp(-\alpha V_{rc})\mathbf{v}_r + \mathbf{d}(V_{rc}, \gamma_{rc}) = \boldsymbol{\tau} + \boldsymbol{\tau}_{wind} \quad (11.77)$$

where $\eta = [N, E, \psi]^\top$, $\mathbf{v} = [u, v, r]^\top$ and

$$\mathbf{d}(V_{rc}, \gamma_{rc}) = \begin{bmatrix} -\frac{1}{2}\rho A_{Fc} C_X(\gamma_{rc}) V_{rc}^2 \\ -\frac{1}{2}\rho A_{Lc} C_Y(\gamma_{rc}) V_{rc}^2 \\ -\frac{1}{2}\rho A_{Lc} L_{od} C_N(\gamma_{rc}) V_{rc}^2 - N_{|r|r} |r| \end{bmatrix} \quad (11.78)$$

This model is highly nonlinear but it is possible to use the extended KF to estimate the velocities and ocean currents using only position and heading measurements. However, the model can be simplified. In particular, $C_{X_c}(\gamma_{rc})$, $C_{Y_c}(\gamma_{rc})$ and $C_{N_c}(\gamma_{rc})$ are difficult to estimate with accuracy and extensive hydrodynamic tests are expensive to perform. In such cases, it is common practice to simplify the observer model in terms of a linear damping matrix and a bias term in the form (Fossen and Strand, 1999b)

$$\mathbf{D} \exp(-\alpha V_{rc})\mathbf{v}_r + \mathbf{d}(V_{rc}, \gamma_{rc}) \approx \mathbf{D}\mathbf{v} - \mathbf{R}^\top(\psi)\mathbf{b} \quad (11.79)$$

where

$$\mathbf{D} = \begin{bmatrix} d_{11} & 0 & 0 \\ 0 & d_{22} & d_{23} \\ 0 & d_{32} & d_{33} \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} \quad (11.80)$$

In this model, the effects of the ocean currents as well as unmodeled nonlinear dynamics are lumped into a bias term \mathbf{b} . The resulting DP model becomes

$$\dot{\xi} = \mathbf{A}_w \xi + \mathbf{E}_w w_1 \quad (11.81)$$

$$\dot{\eta} = \mathbf{R}(\psi)\mathbf{v} \quad (11.82)$$

$$\dot{\mathbf{b}} = \mathbf{w}_2 \quad (\text{alternatively } \dot{\mathbf{b}} = -T^{-1}\mathbf{b} + \mathbf{w}_2) \quad (11.83)$$

$$\mathbf{M}\dot{\mathbf{v}} = -\mathbf{D}\mathbf{v} + \mathbf{R}^\top(\psi)\mathbf{b} + \boldsymbol{\tau} + \boldsymbol{\tau}_{wind} + \mathbf{w}_3 \quad (11.84)$$

$$\mathbf{y} = \eta + \mathbf{C}_w \xi + \mathbf{v} \quad (11.85)$$

where the output $\eta_w = \mathbf{C}_w \xi$ represents three linear wave response models in surge, sway and yaw with state vector $\xi \in \mathbb{R}^6$ and $\mathbf{A}_w \in \mathbb{R}^{6 \times 6}$, $\mathbf{E}_w \in \mathbb{R}^{6 \times 3}$ and $\mathbf{C}_w \in \mathbb{R}^{3 \times 6}$ are constant matrices of appropriate dimensions describing the sea state. The variables w_i ($i = 1, 2, 3$) are zero-mean Gaussian noise vectors representing model uncertainty. The measurement \mathbf{y} is the sum of the LF and WF components corresponding to the GNSS and compass measurements. The bias is usually modeled as a *Wiener process* (random walk):

$$\dot{\mathbf{b}} = \mathbf{w}_2 \quad (11.86)$$

An alternative model is the *first-order Markov* model:

$$\dot{\mathbf{b}} = -T^{-1}\mathbf{b} + \mathbf{w}_2 \quad (11.87)$$

where $T = \text{diag}\{T_1, T_2, T_3\} \in \mathbb{R}^{3 \times 3}$ is a user-specified diagonal matrix of positive bias time constants. These models can be used to describe slowly varying environmental forces and moments due to:

- second-order wave drift forces
- ocean currents
- wind forces (alternatively implemented as wind feedforward)

The estimate of \mathbf{b} will be nonphysical since it contains several components. Many DP operators call this *DP current* since it is experienced as a drift force due to second-order wave drift forces, ocean currents and unmodeled dynamics.

The model (11.81)–(11.85) is nonlinear since the kinematic transformation matrix $\mathbf{R}(\psi)$ depends on the state ψ . This suggests that the DP observer must be based on the extended KF formulation. For a DP system operating at constant heading or slow turning rates, the following assumption can be used:

Assumption (Constant Heading): The yaw rate is zero ($r = 0$) such that $\dot{\mathbf{R}}(\psi) = \mathbf{0}$.

Hence, the use of vessel parallel coordinates implies that (see Section 7.5.3)

$$\boldsymbol{\eta}_p = \mathbf{R}^\top(\psi)\boldsymbol{\eta} \quad (11.88)$$

$$\dot{\mathbf{b}}_p = \mathbf{R}^\top(\psi)\dot{\mathbf{b}} \quad (11.89)$$

Consequently, the kinematics (11.76) can be approximated by a linear model:

$$\begin{aligned} \dot{\boldsymbol{\eta}}_p &= \mathbf{R}^\top(\psi)\dot{\boldsymbol{\eta}} + \dot{\mathbf{R}}^\top(\psi)\boldsymbol{\eta} \\ &= \mathbf{R}^\top(\psi)\mathbf{R}(\psi)\boldsymbol{v} + \dot{\mathbf{R}}^\top(\psi)\boldsymbol{\eta} \\ &= \boldsymbol{v} + \dot{\mathbf{R}}^\top(\psi)\boldsymbol{\eta} \\ &\approx \boldsymbol{v} \end{aligned} \quad (11.90)$$

If the heading angle is constant, the bias model in vessel parallel coordinates can be formulated as

$$\dot{\mathbf{b}}_p = \mathbf{w}_2 \quad (11.91)$$

Remark: Notice that if the heading angle is changed, \mathbf{b}_p needs time to converge to its new value due to the dependency of the rotation matrix. In many commercial systems, the constant heading assumption is removed by designing an EKF for the nonlinear model (11.76)–(11.77) which includes the rotation matrix.

Linear DP Observer Model for Constant Heading

The resulting DP observer model in vessel parallel coordinates becomes

$$\dot{\boldsymbol{\xi}} = \mathbf{A}_w\boldsymbol{\xi} + \mathbf{E}_w\mathbf{w}_1 \quad (11.92)$$

$$\dot{\boldsymbol{\eta}}_p = \boldsymbol{v} \quad (11.93)$$

$$\dot{\mathbf{b}}_p = \mathbf{w}_2 \quad (\text{alternatively } \dot{\mathbf{b}}_p = -\mathbf{T}^{-1}\mathbf{b}_p + \mathbf{w}_2) \quad (11.94)$$

$$\mathbf{M}\dot{\boldsymbol{v}} = -\mathbf{D}\boldsymbol{v} + \mathbf{b}_p + \boldsymbol{\tau} + \boldsymbol{\tau}_{\text{wind}} + \mathbf{w}_3 \quad (11.95)$$

$$\mathbf{y} = \boldsymbol{\eta}_p + \mathbf{C}_w\boldsymbol{\xi} + \boldsymbol{v} \quad (11.96)$$

The control forces usually have two components:

$$\tau = -\hat{\tau}_{\text{wind}} + \mathbf{B}_u \mathbf{u} \quad (11.97)$$

where $\hat{\tau}_{\text{wind}}$ is an estimate of the wind forces implemented by using feedforward compensation and $\mathbf{B}_u \mathbf{u}$ represents actuator forces. The wind feedforward term, which is proportional to the square of the measured wind velocity, depends on the craft's projected area in the direction of the wind (see Section 8.1). The vector \mathbf{u} is the command to the actuators, which are assumed to have a much faster dynamic response than the craft; thus the coefficient \mathbf{B}_u represents the mapping from the actuator command to the force generated by the actuator. For example, if the command to a propeller is the rotation speed, then the corresponding coefficient in \mathbf{B}_u maps the speed to the generated thrust.

The resulting model for a DP observer design is the fifteenth-order state-space model:

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} + \mathbf{E}\mathbf{w} \quad (11.98)$$

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{v} \quad (11.99)$$

where $\mathbf{x} = [\xi^\top, \eta_p^\top, b_p^\top, v^\top]^\top \in \mathbb{R}^{15}$ is the state vector, $\mathbf{u} \in \mathbb{R}^p$ ($p \geq 3$) is the control vector, $\mathbf{w} = [\mathbf{w}_1^\top, \mathbf{w}_2^\top, \mathbf{w}_3^\top]^\top \in \mathbb{R}^9$ represents the process noise vector and $\mathbf{v} \in \mathbb{R}^3$ is a vector of measurement noise. The system matrices are

$$\mathbf{A} = \left[\begin{array}{c|ccc} \mathbf{A}_w & \mathbf{0}_{6 \times 3} & \mathbf{0}_{6 \times 3} & \mathbf{0}_{6 \times 3} \\ \hline \mathbf{0}_{3 \times 6} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{I}_{3 \times 3} \\ \mathbf{0}_{3 \times 6} & \mathbf{0}_{3 \times 3} & -\mathbf{T}^{-1} & \mathbf{0}_{3 \times 3} \\ \hline \mathbf{0}_{3 \times 6} & \mathbf{0}_{3 \times 3} & \mathbf{M}^{-1} & -\mathbf{M}^{-1}\mathbf{D} \end{array} \right], \quad \mathbf{B} = \left[\begin{array}{c} \mathbf{0}_{6 \times p} \\ \hline \mathbf{0}_{3 \times p} \\ \mathbf{0}_{3 \times p} \\ \hline \mathbf{M}^{-1}\mathbf{B}_u \end{array} \right] \quad (11.100)$$

$$\mathbf{E} = \left[\begin{array}{c|cc} \mathbf{E}_w & \mathbf{0}_{6 \times 3} & \mathbf{0}_{6 \times 3} \\ \hline \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \hline \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{M}^{-1} \end{array} \right], \quad \mathbf{H} = [\mathbf{C}_w \mid \mathbf{I}_{3 \times 3} \ \mathbf{0}_{3 \times 3} \ \mathbf{0}_{3 \times 3}] \quad (11.101)$$

Continuous-Time Kalman Filter

The continuous-time filter equations for (11.98) and (11.99) are (see Table 11.2 in Section 11.2.3)

$$\dot{\hat{\mathbf{x}}} = \mathbf{A}\hat{\mathbf{x}} + \mathbf{B}\mathbf{u} + \underbrace{\mathbf{P}\mathbf{H}^\top\mathbf{R}^{-1}}_K(\mathbf{y} - \mathbf{H}\hat{\mathbf{x}}) \quad (11.102)$$

$$\dot{\mathbf{P}} = \mathbf{A}\mathbf{P} + \mathbf{P}\mathbf{A}^\top + \mathbf{E}\mathbf{Q}\mathbf{E}^\top - \mathbf{P}\mathbf{H}^\top\mathbf{R}^{-1}\mathbf{H}\mathbf{P} \quad (11.103)$$

Notice that the covariance matrices $\mathbf{Q} = \mathbf{Q}^\top \in \mathbb{R}^{9 \times 9}$ and $\mathbf{R} = \mathbf{R}^\top \in \mathbb{R}^{3 \times 3}$ must be specified by the user. The measurement covariance matrix can be chosen as

$$\mathbf{R} = \text{diag} \{ \sigma_{v1}^2, \sigma_{v2}^2, \dots, \sigma_{vp}^2 \}$$

where the covariance σ_{vi}^2 of the measurement noise of the sensor i can be estimated by the sample covariance from a data record taken while the craft is at port with no motion. The matrix \mathbf{Q} can also be chosen to be diagonal with positive tunable parameters. These are usually found by trial and error. The estimation of the covariance \mathbf{Q} of the state noise \mathbf{w} in (11.98) is more complex since it depends

on the sea state, the heading of the craft relative to the environmental forces and how uncertain the model is. This covariance matrix is chosen to be block diagonal, that is

$$\mathbf{Q} = \text{diag}\{\mathbf{Q}_1, \mathbf{Q}_2, \mathbf{Q}_3\} \quad (11.104)$$

The matrix $\mathbf{Q}_1 \in \mathbb{R}^{3 \times 3}$ is the covariance of the noise \mathbf{w}_1 , which drives the noise filter representing linear wave-induced motion, $\mathbf{Q}_2 \in \mathbb{R}^{3 \times 3}$ is the covariance of the noise \mathbf{w}_2 , which represents the uncertainty in the equation of motion, and $\mathbf{Q}_3 \in \mathbb{R}^{3 \times 3}$ is the covariance of the noise \mathbf{w}_3 , which represents the uncertainty in the bias term that models the rest of the environmental forces. The matrices \mathbf{Q}_2 and \mathbf{Q}_3 are usually chosen to be diagonal. The entries of the matrix \mathbf{Q}_2 are taken as a fraction of the variance of the position measurement noises. The entries of \mathbf{Q}_3 are high values. These choices provide a filter with an appropriate balance of the uncertainty in various parts of the model. The covariance \mathbf{Q}_1 is estimated together with the parameters of the WF motion model from data measured before and during the operation of the craft.

Discrete-Time Kalman Filter

Since the GNSS measurement frequency can be as low as 1–10 Hz it is advantageous to implement the discrete-time version of the KF. The discrete-time system model is written as

$$\mathbf{x}(k+1) = \Phi \mathbf{x}(k) + \Delta \mathbf{u}(k) + \Gamma \mathbf{w}(k) \quad (11.105)$$

$$\mathbf{y}(k) = \mathbf{H} \mathbf{x}(k) + \mathbf{v}(k) \quad (11.106)$$

where

$$\Phi = \exp(Ah) \quad (11.107)$$

$$\Delta = A^{-1}(\Phi - I)\mathbf{B} \quad (11.108)$$

$$\Gamma = A^{-1}(\Phi - I)\mathbf{E} \quad (11.109)$$

Here h is the sampling time, and the equivalent discrete-time noises $\mathbf{w}(k)$ and $\mathbf{v}(k)$ are Gaussian and white with zero mean. The discrete-time Kalman filter uses the *corrector–predictor* representation (see Table 11.1 in Section 11.3.1):

Kalman Gain:

$$\mathbf{K}(k) = \bar{\mathbf{P}}(k) \mathbf{H}^\top(k) [\mathbf{H}(k) \bar{\mathbf{P}}(k) \mathbf{H}^\top(k) + \mathbf{R}(k)]^{-1} \quad (11.110)$$

Corrector:

$$\hat{\mathbf{P}}(k) = [\mathbf{I} - \mathbf{K}(k) \mathbf{H}(k)] \bar{\mathbf{P}}(k) [\mathbf{I} - \mathbf{K}(k) \mathbf{H}(k)]^\top + \mathbf{K}(k) \mathbf{R}(k) \mathbf{K}^\top(k) \quad (11.111)$$

$$\hat{\mathbf{x}}(k) = \bar{\mathbf{x}}(k) + \mathbf{K}(k) [\mathbf{y}(k) - \mathbf{H}(k) \bar{\mathbf{x}}(k)] \quad (11.112)$$

Predictor:

$$\bar{\mathbf{x}}(k+1) = \Phi(k) \hat{\mathbf{x}}(k) + \Delta(k) \mathbf{u}(k) \quad (11.113)$$

$$\bar{\mathbf{P}}(k+1) = \Phi(k) \hat{\mathbf{P}}(k) \Phi^\top(k) + \Gamma(k) \mathbf{Q}(k) \Gamma^\top(k) \quad (11.114)$$

In order to implement a Kalman filter, the parameters of the model as well as the covariance of the state measurement noises in the model are necessary. The mass and damping parameters of the model can be initially estimated from hydrodynamic computations. Then, an update of the parameter estimates can be obtained from data of tests performed in calm water (Fossen *et al.*, 1996).

The parameters are re-estimated after significant changes in heading or at regular intervals of 20 minutes, which is the time period for which the sea state can be considered to be stationary. Since the craft is in a positioning control mode, the total motion measured can be recorded and detrended to obtain an estimate of the wave-induced motion vector $\hat{\eta}_w(k)$, or, equivalently, a first-order high-pass filter can be used (Holzhüter and Strauch, 1987). These data can then be used to estimate the parameters of the wave-induced motion model, for which it is convenient to consider the directly parameterized innovations form (Ljung, 1999)

$$\hat{\xi}(k+1) = \mathbf{A}_w(\boldsymbol{\theta})\hat{\xi}(k) + \mathbf{K}_w(\boldsymbol{\theta})\epsilon(k) \quad (11.115)$$

$$\hat{\eta}_w(k) = \mathbf{C}_w(\boldsymbol{\theta})\hat{\xi}(k) + \epsilon(k) \quad (11.116)$$

where $\boldsymbol{\theta}$ is the vector of parameters to be estimated and $\epsilon(k)$ is the vector of innovations. The parameter estimation problem can then be formulated as (Fossen and Perez, 2009)

$$\hat{\boldsymbol{\theta}} = \arg \min_{\boldsymbol{\theta}} \det \sum_{k=1}^N \boldsymbol{\epsilon}(k, \boldsymbol{\theta}) \boldsymbol{\epsilon}(k, \boldsymbol{\theta})^\top \quad (11.117)$$

with

$$\boldsymbol{\epsilon}(k, \boldsymbol{\theta}) = \hat{\eta}_w(k) - \mathbf{C}_w(\boldsymbol{\theta})\hat{\xi}(k) \quad (11.118)$$

$$\hat{\xi}(k+1) = \mathbf{A}_w(\boldsymbol{\theta})\hat{\xi}(k) + \mathbf{K}_w(\boldsymbol{\theta})\epsilon(k, \boldsymbol{\theta}) \quad (11.119)$$

where $\eta_w(k)$ is replaced by the estimate $\hat{\eta}_w(k)$ obtained from detrending the measured data. Equations (11.117)–(11.119) comprise a standard prediction error estimation problem whose solution is related to the maximum likelihood estimate of the parameter vector $\boldsymbol{\theta}$ (Harvey, 1989; Ljung, 1999).

Once the parameters of the mode are estimated, the covariance $\hat{\mathbf{Q}}_\epsilon$ of the innovations can also be estimated from the sample covariance of the predictions errors. Then, the Kalman filter can be implemented with the innovation WF model and thus we can chose $\mathbf{Q}_1 = \hat{\mathbf{Q}}_\epsilon$. This choice entails no loss of information.

An alternative to the procedure described above consists of fixing the damping ζ of the WF model to a value in the range 0.01 to 0.1 as suggested in Holzhüter (1992) and estimate only the natural frequency ω_0 and noise covariance (Holzhüter and Strauch, 1987; Holzhüter, 1992). This estimation approach is summarized in Fossen (1994), where recursive least squares is used for parameter estimation. A related approach, also based on recursive least squares, is given in Perez (2005).

11.4 Nonlinear Passive Observer Designs

The drawback of the Kalman filter is that it is difficult and time-consuming to tune the state estimator, which is a stochastic system with 15 states and 120 covariance equations. The main reason for this is that the numerous covariance tuning parameters may be difficult to relate to physical quantities. This results in an *ad hoc* tuning procedure for the process covariance matrix \mathbf{Q} while the measurement covariance matrix \mathbf{R} usually is well defined in terms of sensor specifications.

In the 1990s, vectorial observer backstepping was presented as an alternative design methodology for DP state estimation (Fossen and Grøvlen, 1998). The motivation for this was to avoid vessel parallel coordinates or linearization of the yaw kinematics in order to obtain a global stability result. Another motivating factor was to reduce the relatively time-consuming process of tuning the Kalman filter covariance matrices online. In fact, vectorial observer backstepping resulted in a uniformly globally exponentially stable (UGES) output feedback control system, which could be directly applied to stationkeeping of ships and rigs. The work of Fossen and Grøvlen (1998) is, however, based on a simplified model of the environmental forces, since it is assumed that the WF motion and bias states can be neglected in the design. Aarset *et al.* (1998) have shown that these results can be extended to the general case by including a dynamic model for wave filtering and bias state estimation. It is also possible to extend this result to ships that are course-unstable (open-loop unstable in sway and yaw) thanks to the results by Robertson and Johansson (1998) and Lindegaard and Fossen (2001b).

A drawback with observer backstepping and also Kalman filter-based design techniques is that a relatively large number of parameters must be determined through experimental testing of the craft. This motivated the research of a nonlinear passivity-based observer, since passivity arguments simplify the tuning procedure significantly (Fossen and Strand, 1999b). Hence, the time needed for sea trials and tuning can be drastically reduced. The nonlinear passive observer, as opposed to a linearized or extended Kalman filter, guarantees global convergence of all estimation errors (including the bias terms) to zero. Hence, only one set of observer gains is needed to cover the whole state space. In addition, the number of observer tuning parameters is significantly reduced and the wave filter parameters are directly coupled to the dominating wave frequency. Passivity implies that the phase of the error dynamics is limited by 90 degrees, which gives excellent stability properties. Passivity theory also proved to be a new tool with respect to accurate tuning of the observer. The proposed nonlinear observer opens the way for new controller designs that are more in line with the actual structure of the physical system, for instance by using a nonlinear separation principle (Loria *et al.*, 2000).

For extensions to adaptive wave filtering, see Strand and Fossen (1999), while extensions to position mooring systems are found in Strand (1999).

11.4.1 Case Study: Passive Observer for Dynamic Positioning using GNSS and Compass Measurements

The passive observer is based on Fossen and Strand (1999b) in which the Kalman filter zero yaw rate assumption is removed. The following assumptions are, however, necessary to prove passivity:

Assumption P1: $\mathbf{w} = \mathbf{0}$ and $\mathbf{v} = \mathbf{0}$. The zero-mean Gaussian white noise terms are omitted in the analysis of the observer. If they are included in the Lyapunov function analysis the error dynamics will be uniformly ultimated bounded (UUB) instead of uniform global asymptotical/exponential stable (UGAS/UGES).

Assumption P2: $\mathbf{R}(y_3) = \mathbf{R}(\psi)$, implying that $y_3 = \psi + \psi_w \approx \psi$. This is a good assumption since the magnitude of the wave-induced yaw disturbance ψ_w will normally be less than 5 degrees in extreme weather situations (sea state codes 5–9) and less than 1 degree during normal operation of the ship (sea state codes 0–4).

The following model properties of the inertia and damping matrices will be exploited in the passivation design:

$$\mathbf{M} = \mathbf{M}^\top > 0, \quad \dot{\mathbf{M}} = \mathbf{0}, \quad \mathbf{D} > 0$$

System Model for Nonlinear Passive Observer

The application of Assumptions P1–P2 to (11.81)–(11.85) gives the following DP observer model:

$$\dot{\xi} = A_w \xi \quad (11.120)$$

$$\dot{\eta} = R(y_3)\nu \quad (11.121)$$

$$\dot{b} = -T^{-1}b \quad (\text{alternatively } \dot{b} = 0) \quad (11.122)$$

$$M\dot{\nu} = -D\nu + R^\top(y_3)b + \tau + \tau_{\text{wind}} \quad (11.123)$$

$$y = \eta + C_w \xi \quad (11.124)$$

For notational simplicity (11.120), (11.121) and (11.124) are written in state-space form:

$$\dot{\eta}_0 = A_0 \eta_0 + B_0 R(y_3)\nu \quad (11.125)$$

$$y = C_0 \eta_0 \quad (11.126)$$

where $\eta_0 = [\xi^\top, \eta^\top]^\top$ and

$$A_0 = \begin{bmatrix} A_w & \mathbf{0}_{6 \times 3} \\ \mathbf{0}_{3 \times 6} & \mathbf{0}_{3 \times 3} \end{bmatrix}, \quad B_0 = \begin{bmatrix} \mathbf{0}_{6 \times 3} \\ I_{3 \times 3} \end{bmatrix}, \quad C_0 = [C_w \quad I_{3 \times 3}] \quad (11.127)$$

Observer Equations

The observer equations can be chosen to copy the dynamics (11.120)–(11.124) resulting in 15 ODEs with no covariance updates, as shown in Figure 11.8. Moreover,

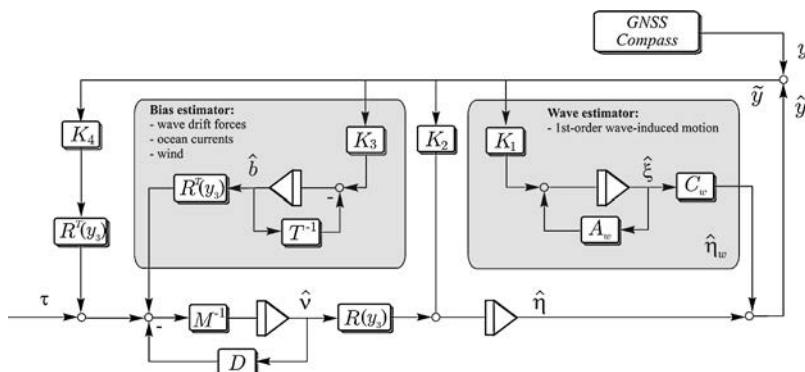


Figure 11.8 Block diagram showing the nonlinear passive DP observer.

$$\dot{\hat{\xi}} = A_w \hat{\xi} + K_1(\omega_o) \tilde{y} \quad (11.128)$$

$$\dot{\hat{\eta}} = R(y_3) \hat{v} + K_2 \tilde{y} \quad (11.129)$$

$$\dot{\hat{b}} = -T^{-1} \hat{b} + K_3 \tilde{y} \quad (\text{alternatively } \dot{\hat{b}} = K_3 \tilde{y}) \quad (11.130)$$

$$M \hat{v} = -D \hat{v} + R^\top(y_3) \hat{b} + \tau + \tau_{\text{wind}} + R^\top(y_3) K_4 \tilde{y} \quad (11.131)$$

$$\hat{y} = \hat{\eta} + C_w \hat{\xi} \quad (11.132)$$

where $\tilde{y} = y - \hat{y}$ is the estimation error and $K_1(\omega_o) \in \mathbb{R}^{6 \times 3}$ and $K_{2,3,4} \in \mathbb{R}^{3 \times 3}$ are observer gain matrices to be interpreted later. Notice that $K_1(\omega_o)$ is a function of the wave spectra peak frequencies $\omega_o = [\omega_{o1}, \omega_{o2}, \omega_{o3}]^\top$ in surge, sway and yaw.

The main difference in performance of the two bias state estimators (11.130) is that the first model includes low-pass filtering ($T > 0$) instead of pure integration of the white noise term $K_3 \tilde{y}$. This results in *exponential stability* while application of the model $\dot{\hat{b}} = K_3 \tilde{y}$ only results in *asymptotic stability*.

Observer Estimation Errors

As for (11.125) and (11.126), the system (11.128), (11.129) and (11.132) is written in state-space form:

$$\dot{\hat{\eta}}_0 = A_0 \hat{\eta}_0 + B_0 R(y_3) \hat{v} + K_0(\omega_o) \tilde{y} \quad (11.133)$$

$$\hat{y} = C_0 \hat{\eta}_0 \quad (11.134)$$

where $\hat{\eta}_0 = [\hat{\xi}^\top, \hat{\eta}^\top]^\top$ and

$$K_0(\omega_o) = \begin{bmatrix} K_1(\omega_o) \\ K_2 \end{bmatrix} \quad (11.135)$$

The estimation errors are defined as $\tilde{v} = v - \hat{v}$, $\tilde{b} = b - \hat{b}$ and $\tilde{\eta}_0 = \eta_0 - \hat{\eta}_0$. Hence, the error dynamics can be written

$$\dot{\tilde{\eta}}_0 = [A_0 - K_0(\omega_o) C_0] \tilde{\eta}_0 + B_0 R(y_3) \tilde{v} \quad (11.136)$$

$$\dot{\tilde{b}} = -T^{-1} \tilde{b} - K_3 \tilde{y} \quad (\text{alternatively } \dot{\tilde{b}} = -K_3 \tilde{y}) \quad (11.137)$$

$$M \dot{\tilde{v}} = -D \tilde{v} + R^\top(y_3) \tilde{b} - R^\top(y_3) K_4 \tilde{y} \quad (11.138)$$

In the Lyapunov analysis of the error dynamics (11.136)–(11.138), it is possible to prove UGES for $T > 0$ (Fossen and Strand, 1999b) since $\dot{V}(x, t) < 0$ (negative definite). If the bias model $\dot{b} = \mathbf{0}$ is applied, that is $T \rightarrow \infty$, the Lyapunov analysis results in $\dot{V}(x, t) \leq 0$ (negative semi-definite). Since the error dynamics is nonautonomous and recall that $y_3 = y_3(t)$ is time varying, Krasovskii–LaSalle’s theorem cannot be applied to prove UGAS. However, it is possible to prove UGAS by using Matrosov’s theorem. Technicalities with respect to the limiting case $T \rightarrow \infty$ are omitted in this section, but the analysis for $T > 0$ is given below.

The dynamics of the velocity estimation error (11.138) is rewritten as

$$M \tilde{v} = -D \tilde{v} - R^\top(y_3) \tilde{z} \quad (11.139)$$

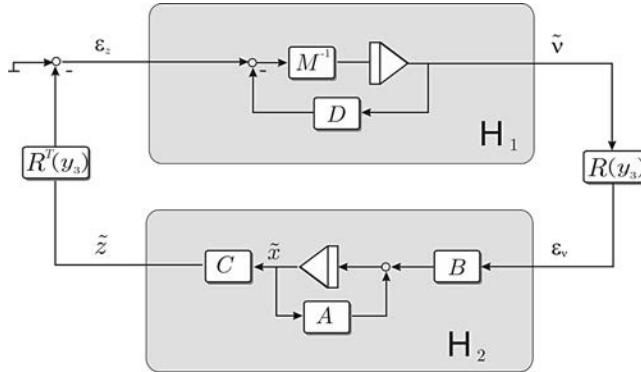


Figure 11.9 Block diagram showing the dynamics of the position/bias and velocity estimation errors.

where

$$\tilde{z} = \mathbf{K}_4 \tilde{y} - \tilde{b} \quad (11.140)$$

By defining a new state vector

$$\tilde{x} = \begin{bmatrix} \tilde{\eta}_0 \\ \tilde{b} \end{bmatrix} \quad (11.141)$$

Equations (11.136), (11.137) and (11.140) can be written in compact form as

$$\dot{\tilde{x}} = \mathbf{A} \tilde{x} + \mathbf{B} \mathbf{R}(y_3) \tilde{v} \quad (11.142)$$

$$\tilde{z} = \mathbf{C} \tilde{x} \quad (11.143)$$

where

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_0 - \mathbf{K}_0(\omega_o) \mathbf{C}_0 & \mathbf{0}_{9 \times 9} \\ -\mathbf{K}_3 \mathbf{C}_0 & -\mathbf{T}^{-1} \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} \mathbf{B}_0 \\ \mathbf{0}_{3 \times 3} \end{bmatrix}, \quad \mathbf{C} = \begin{bmatrix} \mathbf{K}_4 \mathbf{C}_0 - \mathbf{I}_{3 \times 3} \end{bmatrix} \quad (11.144)$$

In Figure 11.9 the error signals ϵ_z and ϵ_v are defined according to

$$\epsilon_z := -\mathbf{R}^\top(y_3) \tilde{z} \quad (11.145)$$

$$\epsilon_v := \mathbf{R}(y_3) \tilde{v} \quad (11.146)$$

Thus, the observer error system can be viewed as two linear blocks H_1 and H_2 , interconnected through the bounded transformation matrix $\mathbf{R}(y_3)$; that is

$$H_1 : \left\{ \begin{array}{l} \mathbf{M} \dot{\tilde{v}} = -\mathbf{D} \tilde{v} + \epsilon_z \\ \epsilon_v := \mathbf{R}(y_3) \tilde{v} \end{array} \right. \quad (11.147)$$

$$H_2 : \left\{ \begin{array}{l} \dot{\tilde{x}} = \mathbf{A} \tilde{x} + \mathbf{B} \epsilon_v \\ \tilde{z} = \mathbf{C} \tilde{x} \end{array} \right. \quad (11.148)$$

Stability Analysis for the Passive Observer

Based on the physical properties of the ship dynamics, the following statement can be made:

Proposition 11.1 (Strictly Passive Velocity Error Dynamics)

The mapping \mathcal{H}_1 is state strictly passive.

Proof. Let,

$$S_1 = \frac{1}{2} \tilde{\mathbf{v}}^\top \mathbf{M} \tilde{\mathbf{v}} \quad (11.149)$$

be a positive definite storage function. Time differentiation of S_1 along the trajectories of $\tilde{\mathbf{v}}$ yields

$$\dot{S}_1 = -\frac{1}{2} \tilde{\mathbf{v}}^\top (\mathbf{D} + \mathbf{D}^\top) \tilde{\mathbf{v}} - \tilde{\mathbf{z}}^\top \mathbf{R}(y_3) \tilde{\mathbf{v}} \quad (11.150)$$

Using the fact that $\boldsymbol{\epsilon}_z = -\mathbf{R}^\top(y_3)\tilde{\mathbf{z}}$, yields

$$\boldsymbol{\epsilon}_z^\top \tilde{\mathbf{v}} = \dot{S}_1 + \frac{1}{2} \tilde{\mathbf{v}}^\top (\mathbf{D} + \mathbf{D}^\top) \tilde{\mathbf{v}} \quad (11.151)$$

Hence,

$$\int_{t_0}^t \boldsymbol{\epsilon}_z^\top(\tau) \tilde{\mathbf{v}}(\tau) d\tau \geq \alpha \tilde{\mathbf{v}}^\top \tilde{\mathbf{v}} + \beta \quad (11.152)$$

where $\alpha = \frac{1}{2}\lambda_{\min}(\mathbf{M})$ is a positive constant and

$$\beta = \frac{1}{2} \int_{t_0}^t \tilde{\mathbf{v}}^\top (\mathbf{D} + \mathbf{D}^\top) \tilde{\mathbf{v}} d\tau \geq 0 \quad (11.153)$$

is the dissipated energy due to hydrodynamic damping. Thus, (11.152) proves that $\boldsymbol{\epsilon}_z \mapsto \tilde{\mathbf{v}}$ or the block H_1 is state strictly passive.

For definitions on passivity see, for instance, Sepulchre *et al.* (1997), Ortega *et al.* (1998) or Lozano *et al.* (2000).

In order to show that the interconnected system in Figure 11.9 is passive, one of the blocks must be passive while the other block must be strictly passive (Lozano *et al.*, 2000). Since the mapping $\boldsymbol{\epsilon}_z \mapsto \tilde{\mathbf{v}}$ is strictly passive (block H_1), post-multiplication with the bounded transformation matrix $\mathbf{R}(y_3)$ and pre-multiplication by its transpose will not affect the passivity properties. Hence, it only remains to show that the the mapping $\boldsymbol{\epsilon}_v \mapsto \tilde{\mathbf{z}}$ (block H_2) is *passive*. This can be done by applying the *Kalman–Yakubovich–Popov (KYP) lemma*.

Lemma 11.1 (Kalman–Yakubovich–Popov)

Let $\mathbf{Z}(s) = \mathbf{C}(s\mathbf{I} - \mathbf{A})^{-1}\mathbf{B}$ be an $m \times m$ transfer function matrix, where \mathbf{A} is Hurwitz, (\mathbf{A}, \mathbf{B}) is controllable and (\mathbf{A}, \mathbf{C}) is observable. Then $\mathbf{Z}(s)$ is strictly positive real (SPR) if and only if there exist positive definite matrices $\mathbf{P} = \mathbf{P}^\top$ and $\mathbf{Q} = \mathbf{Q}^\top$ such that

$$\mathbf{PA} + \mathbf{A}^\top \mathbf{P} = -\mathbf{Q} \quad (11.154)$$

$$\mathbf{B}^\top \mathbf{P} = \mathbf{C} \quad (11.155)$$

Proof. See Yakubovich (1973) or Khalil (2002).

Theorem 11.1 (Passive Observer Error Dynamics)

The interconnected system (11.147) and (11.148) is passive if the observer gain matrices \mathbf{K}_i ($i = 1, \dots, 4$) are chosen such that (11.148) satisfies the KYP lemma.

Proof. Since it is established that H_1 is strictly passive and H_2 , which is given by the matrices (A, B, C) , can be made SPR by choosing the gain matrices K_i ($i = 1, \dots, 4$) according to the KYP lemma, the interconnected system (11.147) and (11.148) is passive (Fossen and Strand, 1999b).

Determination of the Observer Gains

In practice it is easy to find a set of gain matrices \mathbf{K}_i ($i = 1, \dots, 4$) satisfying the KYP lemma. Notice that the mapping $\boldsymbol{\epsilon}_v \mapsto \tilde{\mathbf{z}}$ (block H_2) describes three decoupled systems in surge, sway and yaw. This suggests that the observer gain matrices should have a diagonal structure:

$$\mathbf{K}_1(\omega_o) = \begin{bmatrix} \text{diag}\{K_{11}(\omega_{o1}), K_{12}(\omega_{o2}), K_{13}(\omega_{o3})\} \\ \text{diag}\{K_{14}(\omega_{o1}), K_{15}(\omega_{o2}), K_{16}(\omega_{o3})\} \end{bmatrix} \quad (11.156)$$

$$\mathbf{K}_2 = \text{diag}\{K_{21}, K_{22}, K_{23}\} \quad (11.157)$$

$$\mathbf{K}_3 = \text{diag}\{K_{31}, K_{32}, K_{33}\} \quad (11.158)$$

$$\mathbf{K}_4 = \text{diag}\{K_{41}, K_{42}, K_{43}\} \quad (11.159)$$

Consequently, three decoupled transfer functions can be found:

$$\mathbf{H}(s) = \text{diag}\{h_1(s), h_2(s), h_3(s)\} \quad (11.160)$$

such that

$$\begin{aligned} \tilde{\mathbf{z}}(s) &= \mathbf{H}(s)\boldsymbol{\epsilon}_v(s) \\ &= \mathbf{H}_0(s)\mathbf{H}_B(s)\boldsymbol{\epsilon}_v(s) \end{aligned} \quad (11.161)$$

where

$$\begin{aligned} \mathbf{H}_0(s) &= \mathbf{C}_0[s\mathbf{I} + \mathbf{A}_0 - \mathbf{K}_0(\omega_0)\mathbf{C}_0]^{-1}\mathbf{B}_0 \\ \mathbf{H}_B(s) &= \mathbf{K}_4 + (s\mathbf{I} + \mathbf{T}^{-1})^{-1}\mathbf{K}_3 \end{aligned}$$

The diagonal structure of $\mathbf{H}(s)$ is illustrated in Figure 11.10. The transfer functions $h_{oi}(s)$ ($i = 1, \dots, 3$) and $h_{Bi}(s)$ ($i = 1, \dots, 3$) corresponding to $\mathbf{H}_0(s)$ and $\mathbf{H}_B(s)$, respectively, become

$$h_{oi}(s) = \frac{s^2 + 2\lambda_i\omega_{oi}s + \omega_{oi}^2}{s^3 + (K_{1(i+3)} + K_{2i} + 2\lambda_i\omega_{oi})s^2 + (\omega_{oi}^2 + 2\lambda_i\omega_{oi}K_{2i} - K_{1i}\omega_{oi}^2)s + \omega_{oi}^2K_{2i}} \quad (11.162)$$

$$\begin{aligned} h_{Bi}(s) &= K_{4i} \frac{s + \left(\frac{1}{T_i} + \frac{K_{3i}}{K_{4i}}\right)}{s + \frac{1}{T_i}} \\ &\stackrel{T_i \gg 1}{\approx} K_{4i} \frac{s + \frac{K_{3i}}{K_{4i}}}{s + \frac{1}{T_i}} \end{aligned} \quad (11.163)$$

where ω_{oi} is the wave spectrum peak frequency, T_i is defined in (11.87) and λ_i is the relative damping ratio of the wave spectrum. In order to obtain the desired notch effect (wave filtering) of the observer, the

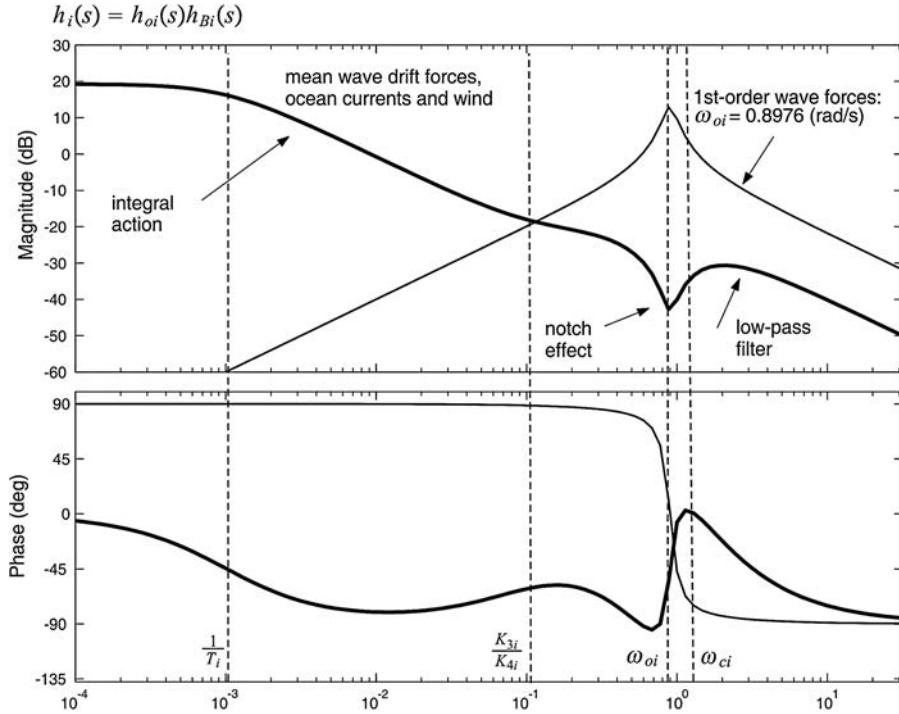


Figure 11.10 Bode plot showing the transfer function $h_i(s)$ in surge ($i = 1$) when $1/T_i \ll K_{3i}/K_{4i} < \omega_{oi} < \omega_{ci}$; see ExPassiveObs.m in the MSS toolbox.

desired shape of $h_{oi}(s)$ is specified as

$$h_{di}(s) = \frac{s^2 + 2\lambda_i\omega_{oi}s + \omega_{oi}^2}{(s^2 + 2\zeta_{ni}\omega_{oi}s + \omega_{oi}^2)(s + \omega_{ci})} \quad (11.164)$$

where $\zeta_{ni} > \lambda_i$ determines the notch and $\omega_{ci} > \omega_{oi}$ is the filter cutoff frequency. Typically $\zeta_{ni} = 1.0$ and $\lambda_i = 0.1$. Equating (11.162) and (11.164) yields the following formulae for the filter gains in $\mathbf{K}_1(\omega_o)$ and \mathbf{K}_2 :

$$K_{1i}(\omega_{oi}) = -2(\zeta_{ni} - \lambda_i) \frac{\omega_{ci}}{\omega_{oi}} \quad (11.165)$$

$$K_{1(i+3)}(\omega_{oi}) = 2\omega_{oi}(\zeta_{ni} - \lambda_i) \quad (11.166)$$

$$K_{2i} = \omega_{ci} \quad (11.167)$$

Notice that the filter gains can be gain-scheduled with respect to the dominating wave frequencies ω_{oi} if desired. In Figure 11.10 the transfer function $h_i(s) = h_{Bi}(s)h_{oi}(s)$ is illustrated when all filter gains are properly selected. It is important that the three decoupled transfer functions $h_i(s)$ all have a phase greater

than -90° in order to meet the SPR requirement. It turns out that the KYP lemma and therefore the SPR requirement can easily be satisfied if the following tuning rules for T_i , K_{3i} and K_{4i} are applied:

$$1/T_i \ll K_{3i}/K_{4i} < \omega_{oi} < \omega_{ci} \quad (i = 1, \dots, 3) \quad (11.168)$$

Here ω_{oi} ($i = 1, \dots, 3$) are the dominating wave frequencies and $T_i \gg 1$ ($i = 1, \dots, 3$) are the bias time constants used to specify the limited integral effect in the bias estimator.

Uniform Global Exponential Stability

The passivity analysis mainly serves as a tool to determine the observer gains. In order to ensure that all estimation errors converge exponentially to zero the following theorem is applied.

Theorem 11.2 (Uniformly Globally Exponentially Stable Observer Error Dynamics)

Under Assumptions P1–P2 the nonlinear observer given by (11.128)–(11.132) is uniformly globally exponentially stable.

Proof. Consider the following Lyapunov function candidate:

$$V = \tilde{\mathbf{v}}^\top \mathbf{M} \tilde{\mathbf{v}} + \tilde{\mathbf{x}}^\top \mathbf{P} \tilde{\mathbf{x}} \quad (11.169)$$

Differentiation of V along the trajectories of $\tilde{\mathbf{v}}$ and $\tilde{\mathbf{x}}$ and application of Assumptions P1–P2 yields

$$\dot{V} = -\tilde{\mathbf{v}}^\top (\mathbf{D} + \mathbf{D}^\top) \tilde{\mathbf{v}} + \tilde{\mathbf{x}}^\top (\mathbf{P}\mathbf{A} + \mathbf{A}^\top \mathbf{P}) \tilde{\mathbf{x}} + 2\tilde{\mathbf{v}}^\top \mathbf{R}^\top(y_3) \mathbf{B}^\top \mathbf{P} \tilde{\mathbf{x}} - 2\tilde{\mathbf{v}}^\top \mathbf{R}^\top(y_3) \tilde{\mathbf{z}} \quad (11.170)$$

Application of the KYP lemma, that is $\mathbf{B}^\top \mathbf{P} \tilde{\mathbf{x}} = \mathbf{C} \tilde{\mathbf{x}} = \tilde{\mathbf{z}}$, to (11.170) yields

$$\dot{V} = -\tilde{\mathbf{v}}^\top (\mathbf{D} + \mathbf{D}^\top) \tilde{\mathbf{v}} - \tilde{\mathbf{x}}^\top \mathbf{Q} \tilde{\mathbf{x}} < 0, \quad \forall \tilde{\mathbf{x}} \neq 0, \tilde{\mathbf{v}} \neq 0 \quad (11.171)$$

Hence, $\tilde{\mathbf{v}}$ and $\tilde{\mathbf{x}} = [\tilde{\xi}^\top, \tilde{\eta}^\top, \tilde{\mathbf{b}}^\top]^\top$ converge exponentially to zero.

Computer Simulations and Experimental Results

A combination of computer simulations and full-scale experiments have been used to evaluate the performance and robustness of the nonlinear passive observer.

Example 11.6 (Passive Nonlinear DP Observer)

The case studies are based on the following models of the ship–bias–wave system (Fossen and Strand, 1999b):

$$\mathbf{M} = \begin{bmatrix} 5.3122 \times 10^6 & 0 & 0 \\ 0 & 8.2831 \times 10^6 & 0 \\ 0 & 0 & 3.7454 \times 10^9 \end{bmatrix} \quad (11.172)$$

$$\mathbf{D} = \begin{bmatrix} 5.0242 \times 10^4 & 0 & 0 \\ 0 & 2.7229 \times 10^5 & -4.3933 \times 10^6 \\ 0 & -4.3933 \times 10^6 & 4.1894 \times 10^8 \end{bmatrix} \quad (11.173)$$

with the coordinate system located in the CG. In the experiments the bias time constants were chosen as

$$\mathbf{T} = \text{diag}\{1000, 1000, 1000\} \quad (11.174)$$

The wave model parameters were chosen as $\lambda_i = 0.1$ and $\omega_{oi} = 0.8976$ rad/s, corresponding to a wave period of 7.0 s in surge, sway and yaw. The notch filter parameters were chosen as $\zeta_{ni} = 1.0$ and $\omega_{ci} = 1.2255\omega_{oi} = 1.1$ rad/s. From (11.165)–(11.167) we get (see the MSS toolbox script ExPassiveObs.m)

$$\mathbf{K}_1 = \begin{bmatrix} -\text{diag}\{2.2059, 2.2059, 2.2059\} \\ \text{diag}\{1.6157, 1.6157, 1.6157\} \end{bmatrix} \quad (11.175)$$

$$\mathbf{K}_2 = \text{diag}\{1.1, 1.1, 1.1\} \quad (11.176)$$

The loop transfer function $h_{oi}(s) = h_{Bi}(s)h_{oi}(s)$ for

$$\mathbf{K}_3 = 0.1\mathbf{K}_4 \quad (11.177)$$

$$\mathbf{K}_4 = \text{diag}\{0.1, 0.1, 0.01\} \quad (11.178)$$

is plotted in Figure 11.10.

Both the simulation study and the full-scale experiment were performed with a measurement frequency of 1 Hz. The simulation study was performed with nonzero noise terms \mathbf{v} and \mathbf{w} even though these terms were assumed to be zero in the Lyapunov analysis. This was done to demonstrate the excellent performance of the observer in the presence of stochastic noise.

The results of the computer simulations are shown in Figures 11.11–11.12. The plots illustrate that all state estimates converge to their true values. In Figures 11.13–11.14 full-scale experimental results with the same observer are reported. Again, excellent convergence and performance in surge, sway and yaw are observed. In the full-scale experiment it was not possible to verify that the velocity estimates converged to their true values; see the lower plots in Figure 11.14. The main reason for this was that only GPS position measurements were available. However, simulation studies indicate that the velocity estimates converge to their true values as well.

11.4.2 Case Study: Passive Observer for Heading Autopilots using only Compass Measurements

The DP observer in Section 11.4.1 can be reduced to 1 DOF and used in autopilot designs. For this purpose, the autopilot model in Section 11.3.5 is considered. In the 1 DOF case, the compass measurement is taken as the sum of the LF and WF signals:

$$y = \psi + \psi_w \quad (11.179)$$

The corresponding system model is

$$\dot{\xi}_w = \psi_w \quad (11.180)$$

$$\dot{\psi}_w = -\omega_0^2 \xi_w - 2\lambda\omega_0 \psi_w \quad (11.181)$$

$$\dot{\psi} = r \quad (11.182)$$

$$\dot{r} = -\frac{1}{T}r + \frac{1}{m}(\tau_{\text{wind}} + \tau_N) + b \quad (11.183)$$

$$b = -\frac{1}{T_b}b \quad (11.184)$$

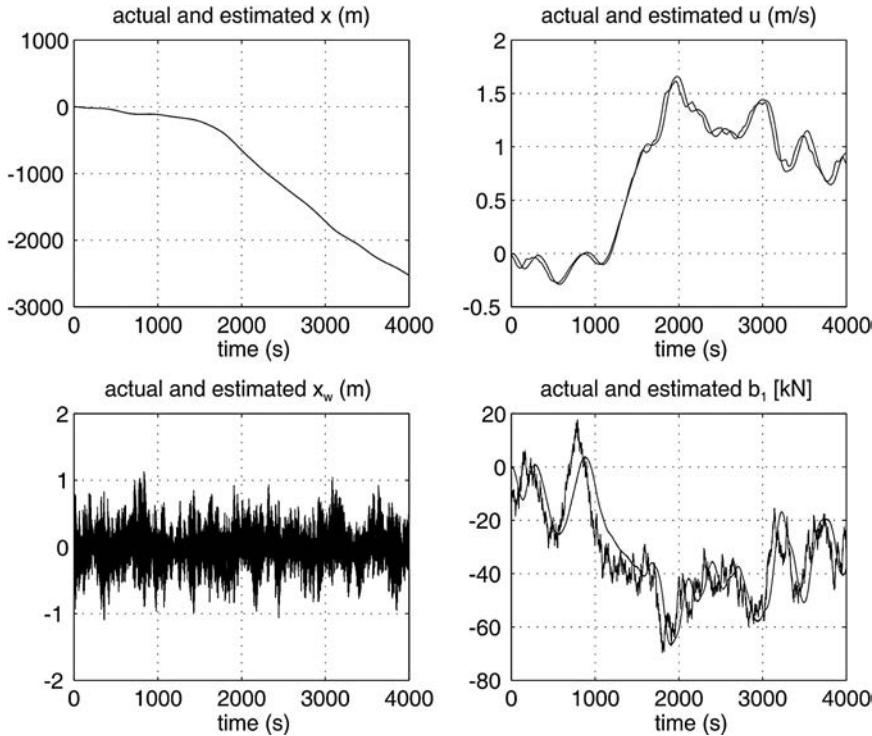


Figure 11.11 Simulation study: LF and WF position, velocity, bias and their estimates in surge.

where λ and ω_0 are the relative damping ratio and peak frequency of the wave spectrum, respectively. The constant $m = I_z - N_r$ is introduced for convenience such that the rudder angle δ generates a yaw moment τ_N given by

$$\begin{aligned} \tau_N &= m \frac{K}{T} \delta \\ &= N_\delta \delta \end{aligned} \quad (11.185)$$

while τ_{wind} represents an optional term for wind feedforward. Notice that neither the yaw rate r nor the wave states ξ_w and ψ_w are measured. The resulting state-space model is

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{b}u \quad (11.186)$$

$$\mathbf{y} = \mathbf{h}^\top \mathbf{x} \quad (11.187)$$

where $\mathbf{x} = [\xi_w, \psi_w, \psi, r, b]^\top$, $u = \tau_{\text{wind}} + \tau_N$ and

$$\mathbf{A} = \left[\begin{array}{cc|ccc} 0 & 1 & 0 & 0 & 0 \\ -\omega_0^2 & -2\lambda\omega_0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1/T & 1 \\ 0 & 0 & 0 & 0 & -1/T_b \end{array} \right], \quad \mathbf{b} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1/m \\ 0 \end{bmatrix} \quad (11.188)$$

$$\mathbf{h}^\top = [0, 1, 1, 0, 0] \quad (11.189)$$

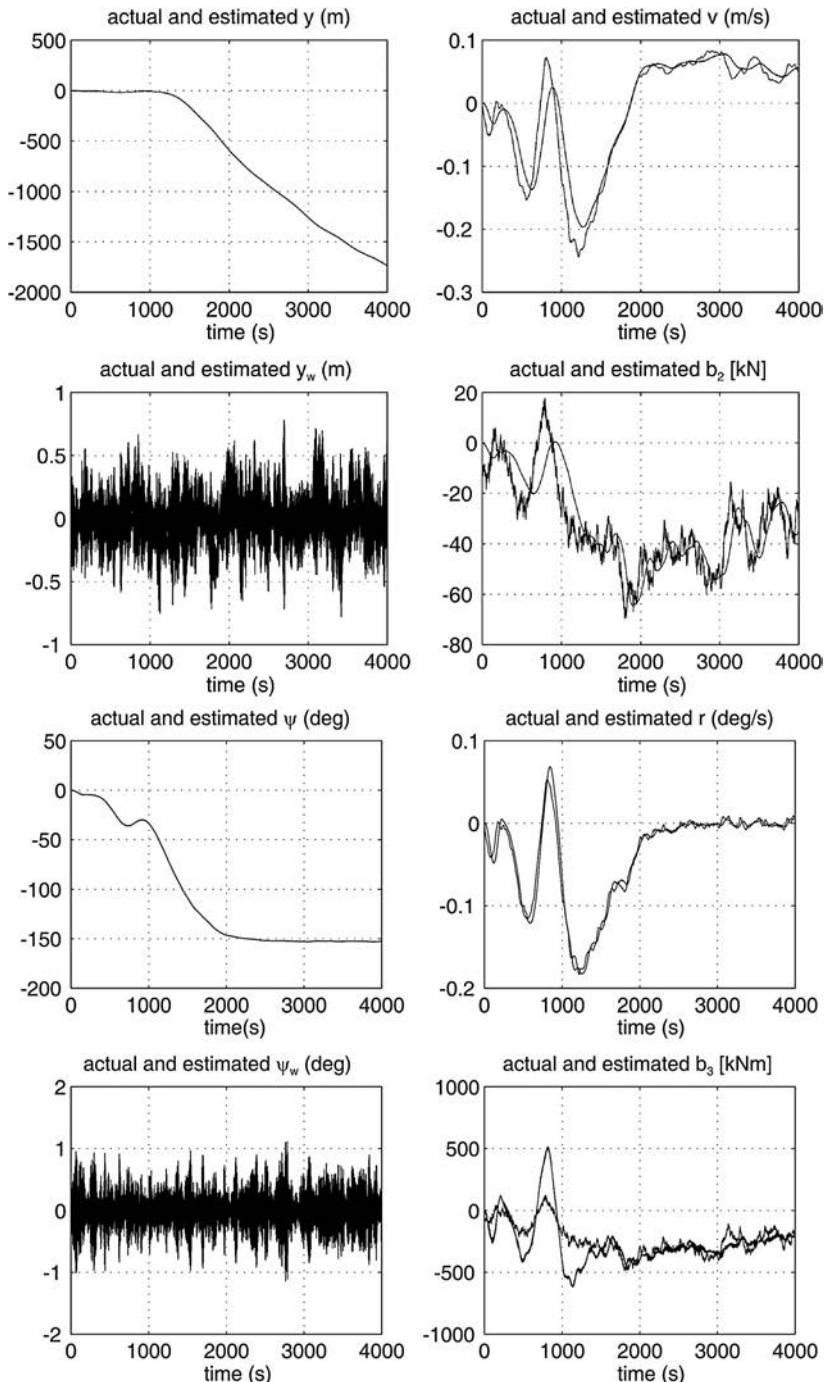


Figure 11.12 Simulation study: LF and WF position, velocity, bias and their estimates in sway and yaw.

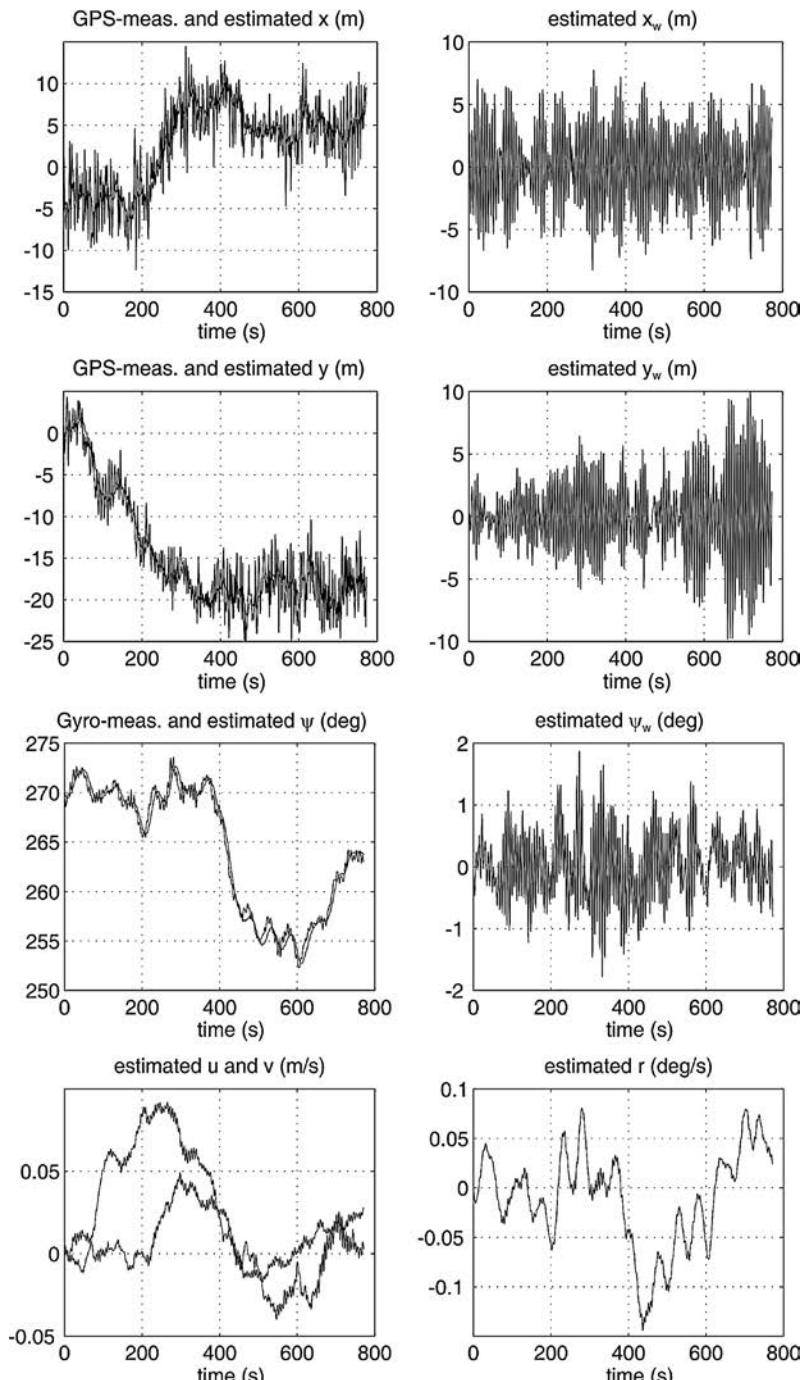


Figure 11.13 Experimental data. Three upper plots: actual position (LF+WF) with estimates of the LF and WF positions in surge, sway and yaw. Lower plots: estimates of the LF velocities.

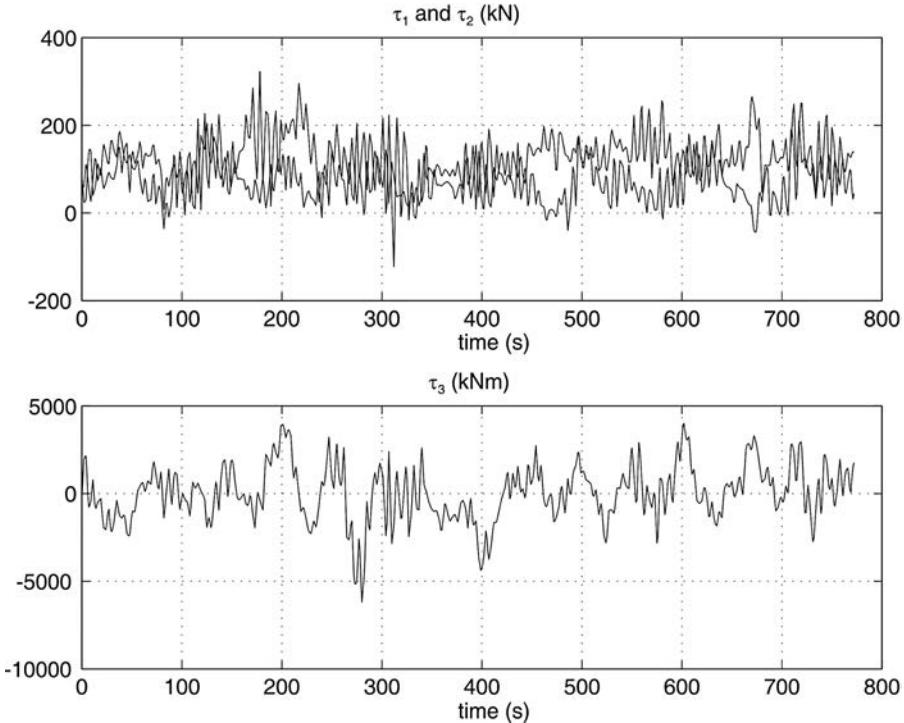


Figure 11.14 Experimental data: control inputs in surge, sway and yaw.

The passive observer copying the dynamics (11.186) and (11.187) is

$$\dot{\hat{x}} = A\hat{x} + bu + k(y - h^\top \hat{x}) \quad (11.190)$$

$$y = h^\top x \quad (11.191)$$

Expanding this expression gives

$$\dot{\hat{\xi}}_w = \hat{\psi}_w + K_1 \varepsilon \quad (11.192)$$

$$\dot{\hat{\psi}}_w = -\omega_0^2 \hat{\xi}_w - 2\lambda\omega_0 \hat{\psi}_w + K_2 \varepsilon \quad (11.193)$$

$$\dot{\hat{r}} = \hat{r} + K_3 \varepsilon \quad (11.194)$$

$$\dot{\hat{b}} = -\frac{1}{T} \hat{r} + \frac{1}{m} (\tau_{\text{wind}} + \tau_N) + \hat{b} + K_4 \varepsilon \quad (11.195)$$

$$\dot{\hat{b}} = -\frac{1}{T_b} \hat{b} + K_5 \varepsilon \quad (11.196)$$

where $\varepsilon = y - \hat{y}$ is the estimation error. The observer gains K_1, K_2, K_3, K_4 and K_5 can be computed by noticing that the observer error dynamics can be reformulated as two subsystems for yaw angle/rudder bias and yaw rate. These systems form a *passive interconnection* if the observer gains are chosen according to

$$\mathbf{k} = \begin{bmatrix} -2\omega_0(1-\lambda)/\omega_c \\ 2\omega_0(1-\lambda) \\ \omega_c \\ K_4 \\ K_5 \end{bmatrix} \quad (11.197)$$

where $\omega_c > \omega_0$ is the filter cutoff frequency and the remaining gains must satisfy

$$0 < 1/T_b < K_5/K_4 < \omega_0 < \omega_c \quad (11.198)$$

The design problem is now reduced to choosing K_4 and K_5 such that the ratio K_5/K_4 satisfies the passive gain constraint (11.198).

Matlab

The passive wave filter can be simulated using the Simulink block:

passive autopilot wave filter 2

in the MSS toolbox (see Example 11.7).

A more detailed analysis of the passive observer is done in Section 11.4.1, which discusses applications to ship positioning in 3 DOF.

Example 11.7 (Passive Wave Filtering)

Consider the Mariner class cargo ship with $K = 0.185 \text{ s}^{-1}$, $T = T_1 + T_2 - T_3 = 107.3 \text{ s}$ and input $\tau_N/m = (K/T)\delta$, where δ is the rudder angle (Chislett and Strøm-Tøjsen, 1965a). The bias time constant is chosen to be rather large, for instance $T_b = 100 \text{ s}$. The wave response model is modeled by a linear approximation to the JONSWAP spectrum with $\lambda = 0.1$ and $\omega_0 = 1.2 \text{ rad/s}$ (see Section 8.2.6). Hence, (11.34)–(11.35) become

$$\mathbf{A} = \left[\begin{array}{cc|ccc} 0 & 1 & 0 & 0 & 0 \\ -1.44 & -0.24 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -0.0093 & 1 \\ 0 & 0 & 0 & 0 & -0.01 \end{array} \right], \quad \mathbf{b} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0.0017 \\ 0 \end{bmatrix} \quad (11.199)$$

$$\mathbf{E} = \left[\begin{array}{c|cc} 0 & 0 & 0 \\ 0.24\sigma & 0 & 0 \\ \hline 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{array} \right], \quad \mathbf{h}^\top = [0, 1, 1, 0, 0] \quad (11.200)$$

where $\sigma > 0$ reflects the sea state. Using passivity as a tool for filter design with cutoff frequency $\omega_c = 1.1 \omega_0$ yields

$$\mathbf{k} = \begin{bmatrix} K_1 \\ K_2 \\ \hline K_3 \\ K_4 \\ K_5 \end{bmatrix} = \begin{bmatrix} -2\omega_0(1-\lambda)/\omega_c \\ 2\omega_0(1-\lambda) \\ \hline \omega_c \\ K_4 \\ K_5 \end{bmatrix} = \begin{bmatrix} -1.64 \\ 1.80 \omega_0 \\ \hline 1.10 \omega_0 \\ K_4 \\ K_5 \end{bmatrix} \quad (11.201)$$

This clearly shows that the gains should be adjusted with varying ω_0 . Choosing $K_4 = 0.1$ and $K_5 = 0.01$ such that $K_5/K_4 = 0.1$ yields the transfer functions shown later in Figure 11.16. Notice that the notch effect at ω_0 for $h_3(s)$ and $h_4(s)$ represents the state estimates $\hat{\psi}$ and \hat{r} . We also see that high-frequency motion components above ω_c are low-pass filtered. Finally, the transfer function $h_2(s)$ representing reconstruction of the WF motion $\hat{\psi}_w$ filters out signals on the outside of the wave response spectrum, while signals close to ω_0 pass through the filter with unity gain, that is 0 dB. The poles of the error dynamics are

$$\begin{aligned} p_1 &= -0.7248 + 0.4388i \\ p_2 &= -0.7248 - 0.4388i \\ p_3 &= -2.1762 \\ p_4 &= -0.1037 \\ p_5 &= -0.0098 \end{aligned}$$

The time series for $\sigma = 6.25$ are shown in Figure 11.15.

Wave Filter Frequency Analysis

Consider the state estimator

$$\dot{\hat{x}} = \mathbf{A}\hat{x} + \mathbf{b}u + \mathbf{k}(y - \mathbf{h}^\top \hat{x}) \quad (11.202)$$

It is then straightforward to show that

$$\hat{x}(s) = (s\mathbf{I} - \mathbf{A} + \mathbf{k}\mathbf{h}^\top)^{-1}(\mathbf{k}y(s) + \mathbf{b}u(s)) \quad (11.203)$$

Assume that $u(s) = 0$ (no feedback) such that

$$\mathbf{h}(s) = [h_1, h_2, h_3, h_4, h_5]^\top = (s\mathbf{I} - \mathbf{A} + \mathbf{k}\mathbf{h}^\top)^{-1}\mathbf{k} \quad (11.204)$$

The states of interest are

$$\hat{\psi}_w(s) = h_2(s)y(s) \quad (11.205)$$

$$\hat{\psi}(s) = h_3(s)y(s) \quad (11.206)$$

$$\hat{r}(s) = h_4(s)y(s) \quad (11.207)$$

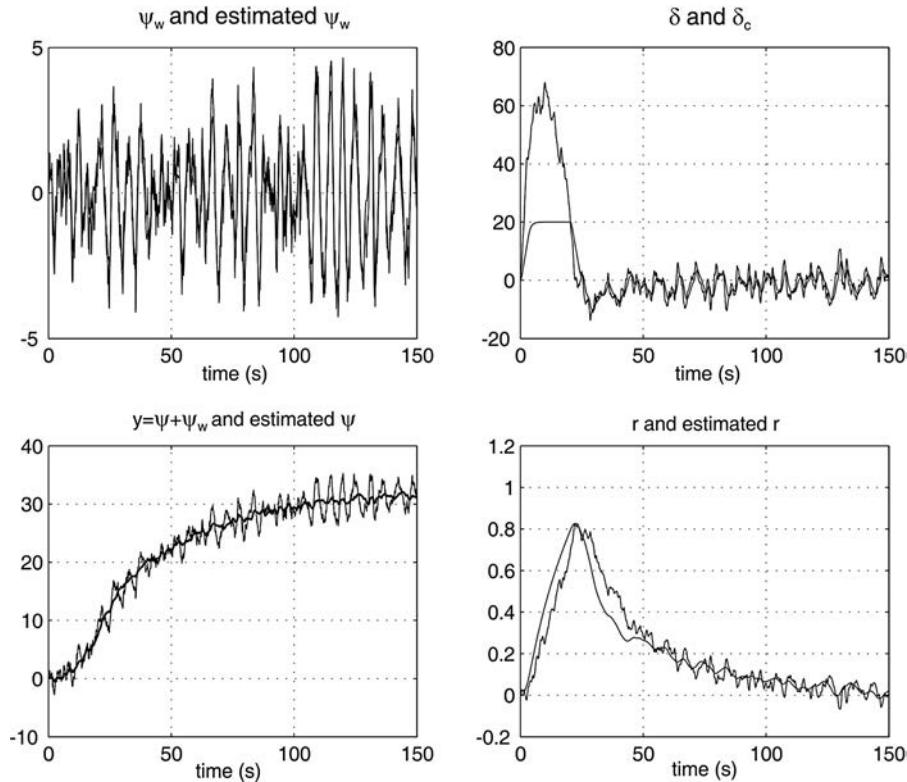


Figure 11.15 Time series showing the performance of the passive wave filter.

where $h_3(s)$ represents a notch filter with a low-pass filter in cascade:

$$h_3(s) = h_{\text{notch}}(s) \cdot h_{\text{low pass}}(s) \quad (11.208)$$

The filter $h_4(s)$ also possesses notch filtering in cascade with a second filter representing a limited differentiator for generation of $\hat{r}(s)$ from $y(s)$. Notice that $h_2(s)$ is close to 1 (0 dB) in a band around the wave spectrum, while lower and higher frequencies are suppressed in order to reconstruct $\psi_w(s)$ from $y(s)$. This can be seen from the Bode plot in Figure 11.16. These results have also been theoretically verified by Grimble (1978). In this work Grimble showed that the *stationary Kalman filter* for the ship positioning problem will be approximately equivalent to a notch filter in cascade with a second filter, typically a low-pass filter.

When including the feedback term $u(s)$ in the analysis, it is well known that application of an observer is superior to notch and low-pass filtering in cascade, since the observer uses the input $u(s)$ for prediction in addition to filtering the measured output $y(s)$. In fact, this input signal reduces the problems associated with additional phase lag in the filtered signal, which is the main problem with most standard filters (low-pass, high-pass and notch). Simulation results verifying these observations have been documented by Grimble (1978).

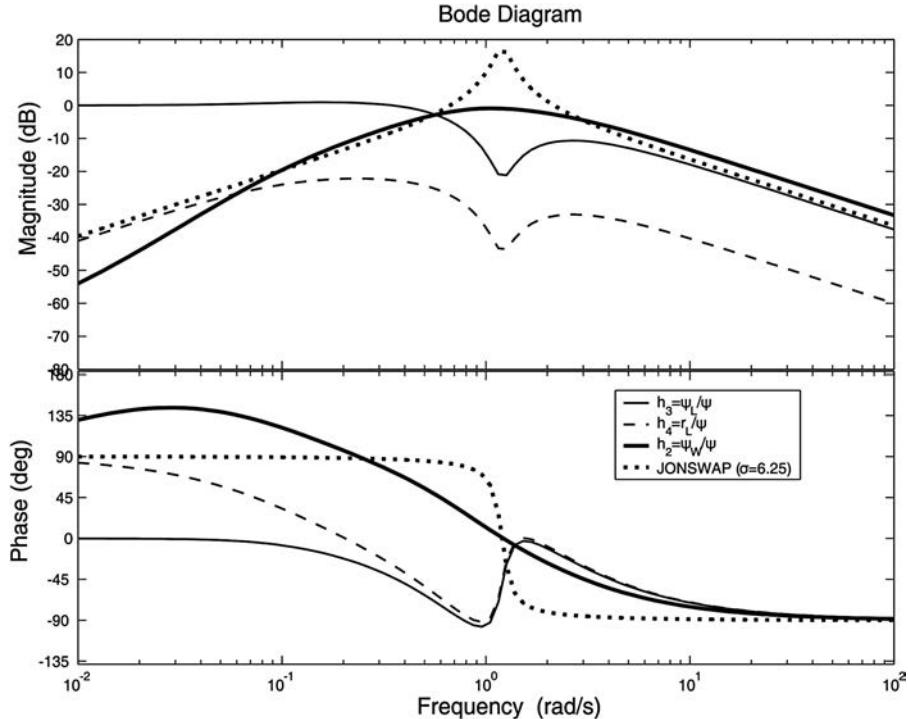


Figure 11.16 Bode plot showing the wave filter transfer functions and the JONSWAP spectrum.

11.4.3 Case Study: Passive Observer for Heading Autopilots using both Compass and Rate Measurements

In this section the design of the previous section is modified to include a rate gyro in addition to the compass. This is advantageous since the gyro can be integrated with the compass in an optimal manner, resulting in less variance and better accuracy of the state estimates. One simple way to do this is to treat the gyro measurements as an input to the system model by writing the yaw dynamics according to

$$\dot{\psi} = u_{\text{gyro}} + b \quad (11.209)$$

where b denotes the gyro bias and u_{gyro} is the *rate gyro measurement*. The WF model is similar to (11.180)–(11.181). This model will give proper wave filtering of the state ψ . However, the estimate of r is not wave filtered, since this signal is taken directly from the gyro measurement u_{gyro} . This can be improved by filtering u_{gyro} with a notch filter $h_{\text{notch}}(s)$ and a low-pass filter $h_{\text{lp}}(s)$ to the cost of some phase lag:

$$u_f = h_{\text{notch}}(s) h_{\text{lp}}(s) u_{\text{gyro}} \quad (11.210)$$

The observer equations become

$$\dot{\hat{\xi}}_w = \hat{\psi}_w + K_1 \varepsilon \quad (11.211)$$

$$\dot{\hat{\psi}}_w = -\omega_0^2 \hat{\xi}_w - 2\lambda \omega_0 \hat{\psi}_w + K_2 \varepsilon \quad (11.212)$$

$$\dot{\hat{\psi}} = u_f + \hat{b} + K_3 \varepsilon \quad (11.213)$$

$$\dot{\hat{b}} = -\frac{1}{T_b} \hat{b} + K_4 \varepsilon \quad (11.214)$$

where $\varepsilon = y - \hat{\psi} - \hat{\psi}_w$ and $T_b \gg 0$. Notice that the gyro bias must be estimated online since it will vary with temperature and possible scale factor/misalignment errors when mounted onboard the ship. This is a slowly varying process so the gain K_4 can be chosen quite small, reflecting a large bias time constant. If passivity-based pole placement (11.197) is used, K_1 , K_2 and K_3 become

$$K_1 = -2 \frac{\omega_0}{\omega_c} (1 - \lambda), \quad K_2 = 2\omega_0(1 - \lambda), \quad K_3 = \omega_c \quad (11.215)$$

Alternatively, the KF algorithm can be used to compute the gains.

Matlab

The observer with compass and rate measurements can be simulated using the Simulink block:

passive autopilot wave filter 1

in the MSS toolbox.

Other techniques for the integration of compass and rate measurements are described in Lindegaard (2003).

11.5 Integration Filters for IMU and Global Navigation Satellite Systems

An inertial measurement unit (IMU) can be integrated with a satellite navigation system in a state observer to obtain estimates of generalized position and velocity in 6 DOFs. The measurements available from a typical IMU are three-axes rate gyros, accelerometers and magnetometers. A stand alone IMU solution, where acceleration measurements are integrated twice and gyro outputs are integrated once to obtain positions and attitude, respectively, will drift due to sensor biases, misalignments and temperature variations (see Figure 11.17). Hence, an estimator providing feedback and compensation of bias drift terms is needed. The kinematic equations (strapdown equations) which are integrated numerically in conjunction with an IMU constitutes an inertial navigation system (INS). The INS drift can be removed by GNSS/INS integration in a state observer. The 6 DOF solution for drift compensation requires that the coupled observers for linear and angular velocity estimation are constructed while the special case where only the 3 DOF rotation dynamics is considered is referred to as an attitude and heading reference system (AHRS).

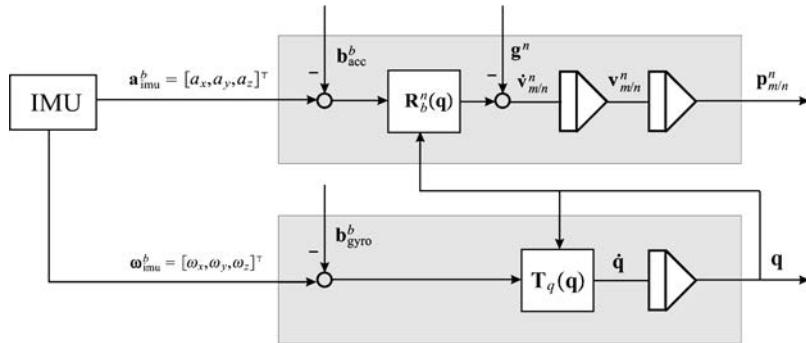


Figure 11.17 The principle for integration of IMU sensor data. The position and quaternion outputs will drift due to the bias terms.

The position and velocity accuracies will mainly depend on the GNSS quality while acceleration and attitude depend on the quality of the accelerometers, gyros and magnetometers. If a low-cost IMU is used, the position and attitude estimates will drift rapidly during GNSS shortages while a more expensive unit will have better stand alone capabilities. Construction of integrated GNSS/INS navigation systems, their performance and stand alone capabilities are described more closely in Farrell and Barth (1998), Titterton and Weston (1997) and Grewal *et al.* (2001), to mention only some. Strapdown inertial navigation systems are usually designed using the EKF. However, a nonlinear observer avoiding the Riccati equations is presented by Vik and Fossen (2001).

The goal of this section is to present low-cost IMU/GNSS integration techniques for marine craft navigation by neglecting the Earth rotation and assuming that the GNSS signals are available all the time. Consequently, the *North-East-Down* reference frame $\{n\}$ is assumed to be the inertial reference frame even though the Earth is moving relatively to a star fixed reference frame. This is, indeed, a good approximation for a marine craft navigating on the surface of the Earth. The solutions presented here are not intended for INS stand alone applications or cases with GNSS failure.

IMU Measurements

Today inertial measurement technology is available for commercial users thanks to a significant reduction in price during the last decades. As a consequence of this, low-cost inertial sensors can be integrated with a satellite navigation system using a conventional Kalman filter or a nonlinear state observer; see Farrell and Barth (1998), Titterton and Weston (1997), Grewal *et al.* (2001) and Vik and Fossen (2001). The key components of the IMU are:

- **Gyrosopes:** The classic gyro is a spinning wheel that utilizes conservation of momentum to detect rotation, and belongs naturally in a gimballed system. For strapdown applications, optical gyros such as ring laser gyros (RLG) and fiber optic gyros (FOG) have been used for some time, and are also expected to be the standard for high accuracy strapdown inertial systems for the foreseeable future. For low and medium cost applications, gyros based on micro-electric-mechanical systems (MEMSs) are expected to be dominant (Barbour and Schmidt, 1998).
- **Accelerometers:** There are several different types of accelerometer. Two of these are *mechanical* and *vibratory accelerometers*. The mechanical accelerometer can be a *pendulum*, which in its simplest form is based on Newton's second law of motion:

$$F = ma$$

A force F acting on a body of mass m causes the body to accelerate with respect to inertial space. When the case of the instrument is subjected to an acceleration along its sensitive axis, the proof mass tends to resist the change in movement due to its own inertia. As a result, the mass is displaced with respect to the case. Under steady-state conditions the force acting on the mass will be balanced by the tension of the spring. The extension of the spring then provides a measure of the force, which is proportional to the acceleration.

The *vibratory accelerometers* are usually based on measurement of frequency shifts due to increased or decreased tension in a string. The operation is similar to that of a violin. When a violin string is tightened, the frequency goes up. Similarly, when the accelerometer proof mass attached to a quartz beam is loaded, the frequency of the quartz beam increases. The difference in frequency is measured, and is proportional to the applied acceleration. In addition to quartz technology, vibrating beam accelerometers using silicon are also being developed.

The inertial sensors are mounted onboard the craft in a body-fixed coordinate system $\{m\}$ located at $\mathbf{r}_m^b := \mathbf{r}_{m/b}^b = [x_m, y_m, z_m]^\top$ with respect to the $\{b\}$ -frame coordinate origin CO. This is referred to as a strapdown system because the sensors are strapped to the craft and a lightweight digital computer is used to perform computations. Thus the need for a mechanical gimbal system is eliminated. Instead of transforming the IMU measurements to the coordinate origin of $\{b\}$, the state estimator is formulated in $\{m\}$ and the estimated states are transformed to $\{b\}$ using the lever arm given by \mathbf{r}_m^b . The tool for this is the transformation matrix in Section 7.5.4. The estimated velocity vector $\hat{\mathbf{v}}_m = [(\hat{\mathbf{v}}_{m/n}^b)^\top, (\hat{\omega}_{m/n}^b)^\top]^\top$ can be transformed to $\{b\}$ to obtain $\hat{\mathbf{v}}$ by using the following transformation:

$$\begin{bmatrix} \hat{\mathbf{v}}_{b/n}^b \\ \hat{\omega}_{b/n}^b \end{bmatrix} = \mathbf{H}^{-1}(\mathbf{r}_m^b) \begin{bmatrix} \hat{\mathbf{v}}_{m/n}^b \\ \hat{\omega}_{m/n}^b \end{bmatrix} \quad (11.216)$$

‡

$$\hat{\mathbf{v}} = \mathbf{H}^{-1}(\mathbf{r}_m^b) \hat{\mathbf{v}}_m \quad (11.217)$$

where

$$\mathbf{H}^{-1}(\mathbf{r}_m^b) = \begin{bmatrix} \mathbf{I}_{3 \times 3} & \mathbf{S}(\mathbf{r}_m^b) \\ \mathbf{0}_{3 \times 3} & \mathbf{I}_{3 \times 3} \end{bmatrix} \quad (11.218)$$

If the IMU is mounted close to the coordinate origin CO this transformation is not needed.

The measurements from the three-axis rate gyros, accelerometers and magnetometers are conveniently expressed as (Mahony *et al.*, 2008)

$$\mathbf{a}_{\text{imu}}^b = \mathbf{R}_n^b(\Theta)(\dot{\mathbf{v}}_{m/n}^n + \mathbf{g}^n) + \mathbf{b}_{\text{acc}}^b + \mathbf{w}_{\text{acc}}^b \quad (11.219)$$

$$\boldsymbol{\omega}_{\text{imu}}^b = \boldsymbol{\omega}_{m/n}^b + \mathbf{b}_{\text{gyro}}^b + \mathbf{w}_{\text{gyro}}^b \quad (11.220)$$

$$\mathbf{m}_{\text{imu}}^b = \mathbf{R}_n^b(\Theta)\mathbf{m}^n + \mathbf{b}_{\text{mag}}^b + \mathbf{w}_{\text{mag}}^b \quad (11.221)$$

where $\Theta = [\phi, \theta, \psi]^\top$ is a vector of Euler angles and $\mathbf{R}_n^b(\Theta)$ is the rotation matrix between $\{n\}$ and $\{b\}$ (see Section 2.2). Alternatively, the quaternion rotation matrix $\mathbf{R}_n^b(\mathbf{q})$ can be used. The accelerometer and gyro biases are denoted as $\mathbf{b}_{\text{acc}}^b$ and $\mathbf{b}_{\text{gyro}}^b$ while $\mathbf{b}_{\text{mag}}^b$ is the local magnetic disturbance. Additive zero-mean sensor measurement noises are modeled by $\mathbf{w}_{\text{acc}}^b$, $\mathbf{w}_{\text{gyro}}^b$ and $\mathbf{w}_{\text{mag}}^b$.

The IMU measurement model is only valid for low-speed applications such as a marine craft moving on the surface of the Earth since it assumes that $\{n\}$ is nonrotating. For terrestrial navigation the Earth rotation will affect the results and it is necessary to express the velocities and accelerations in $\{i\}$. Inertial navigation systems are also sensitive to scale factor and misalignments angles due to inaccurate mounting of the IMU (Titterton and Weston, 1997). These effects can, however, be neglected for local area navigation and low-speed applications. The following sections discuss effective methods for GNSS/INS integration.

Gravity

The gravity of Earth in $\{n\}$ is modeled as a constant vector:

$$\mathbf{g}^n = \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} \quad (11.222)$$

Gravity increases from 9.789 m/s^2 at the equator to 9.832 m/s^2 at the poles. The nominal “average” value at the surface of the Earth, known as standard gravity, is, by definition, $g = 9.80665 \text{ m/s}^2$.

Compass Heading and Roll-Pitch Angles from Magnetometers

The magnetic field of the Earth is similar to a simple bar magnet. The magnetic field is a magnetic dipole that has its field lines originating at a point near the South Pole and terminating at a point near the North Pole. The field lines vary in both strength and direction about the face of the Earth. At each location on the Earth, the field lines intersect the Earth’s surface at a specific angle of inclination. Near the equator, the field lines are approximately parallel to the Earth’s surface and thus the inclination angle in this region is 0° . As one travels North from the equator the field lines become progressively steeper. At the magnetic pole, the field lines are directed almost straight down into the Earth and the inclination is 90° . Consequently, the inclination angle varies with latitude.

It is necessary to perform filtering and calibration of the magnetometer to remove the bias $\mathbf{b}_{\text{mag}}^b$ and noise $\mathbf{w}_{\text{mag}}^b$. The calibrated measurements are denoted

$$\begin{bmatrix} m_x \\ m_y \\ m_z \end{bmatrix} = \mathbf{m}_{\text{imu}}^b - \mathbf{b}_{\text{mag}}^b \quad (11.223)$$

The magnetic compass heading can be determined from the three magnetometer measurements if the tilt angles of the device are known. If the magnetometer is sitting in a local horizontal plane leveled to the surface of the Earth such that $\phi = \theta = 0$, the *magnetic heading angle* ψ_m is recognized as the direction planar with the surface of the Earth satisfying

$$\tan(\psi_m) = \frac{m_y}{m_x} \quad (11.224)$$

One method to determine the roll and pitch angles is to use a tilt sensor. Alternatively, a gyroscope can be used to maintain a known inertial reference frame at all times. If the roll and pitch

angles are known the magnetic readings m_x , m_y and m_z can be transformed to the horizontal plane according to

$$\begin{bmatrix} h_x \\ h_y \\ h_z \end{bmatrix} = \mathbf{R}_{y,\theta} \mathbf{R}_{x,\phi} \begin{bmatrix} m_x \\ m_y \\ m_z \end{bmatrix} \quad (11.225)$$

or

$$\begin{bmatrix} h_x \\ h_y \\ h_z \end{bmatrix} = \begin{bmatrix} \cos(\theta) & 0 & \sin(\theta) \\ 0 & 1 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & \sin(\phi) & \cos(\phi) \end{bmatrix} \begin{bmatrix} m_x \\ m_y \\ m_z \end{bmatrix} \quad (11.226)$$

The horizontal components are

$$h_x = m_x \cos(\theta) + m_y \sin(\theta) \sin(\phi) + m_z \cos(\phi) \sin(\theta) \quad (11.227)$$

$$h_y = m_y \cos(\phi) - m_z \sin(\phi) \quad (11.228)$$

The sign of the arguments h_x and h_y must be taken into account when computing the magnetic heading. This is can be done by using the following mapping:

$$\psi_m = \begin{cases} 180^\circ - \frac{180^\circ}{\pi} \arctan\left(\frac{h_y}{h_x}\right) & \text{if } h_x < 0 \\ -\frac{180^\circ}{\pi} \arctan\left(\frac{h_y}{h_x}\right) & \text{if } h_x > 0, h_y < 0 \\ 360^\circ - \frac{180^\circ}{\pi} \arctan\left(\frac{h_y}{h_x}\right) & \text{if } h_x > 0, h_y > 0 \\ 90^\circ & \text{if } h_x = 0, h_y < 0 \\ 270^\circ & \text{if } h_x = 0, h_y > 0 \end{cases} \quad (11.229)$$

To determine true North heading ψ the appropriate declination angle which depends on the latitude must be added or subtracted.

11.5.1 Integration Filter for Position and Linear Velocity

The expression for the linear acceleration in $\{n\}$ is derived from the acceleration measurement equation (11.219). Moreover,

$$\dot{\mathbf{v}}_{m/n}^n = \mathbf{R}_n^b(\Theta)^\top [\mathbf{a}_{\text{imu}}^b - \mathbf{b}_{\text{acc}}^b - \mathbf{w}_{\text{acc}}^b] - \mathbf{g}^n \quad (11.230)$$

Since the measurement noise $E(\mathbf{w}_{\text{acc}}^b) = \mathbf{0}$, the velocity observer will be designed under the assumption that the term $\mathbf{w}_{\text{acc}}^b$ can be neglected when analyzing the stability properties of the error dynamics.

Integration of IMU and GNSS Position Measurements

The state estimator will be formulated in $\{m\}$. The position of the IMU coordinate system $\{m\}$ with respect to the NED reference frame $\{n\}$ expressed in $\{n\}$ is denoted by $\mathbf{p}_{m/n}^n$. If the GNSS receiver is located at $\mathbf{r}_{\text{gnss}}^b = [x_{\text{gnss}}, y_{\text{gnss}}, z_{\text{gnss}}]^\top$ with respect to $\{m\}$, the GNSS position measurements $\mathbf{p}_{\text{gnss}}^n = [N_{\text{gnss}}, E_{\text{gnss}}, D_{\text{gnss}}]^\top$ must be corrected for rotations Θ and lever arm $\mathbf{r}_{\text{gnss}}^b$ according to

$$\mathbf{p}_{m/n}^n = \mathbf{p}_{\text{gnss}}^n - \mathbf{R}_b^n(\Theta) \mathbf{r}_{\text{gnss}}^b \quad (11.231)$$

If the GNSS receiver is located next to the IMU, $\mathbf{r}_{\text{gnss}}^b = \mathbf{0}$, the position measurements satisfies $\mathbf{p}_{m/n}^n = \mathbf{p}_{\text{gnss}}^n$.

The translational dynamics including acceleration bias is (see Figure 11.17)

$$\dot{\mathbf{p}}_{m/n}^n = \mathbf{v}_{m/n}^n \quad (11.232)$$

$$\dot{\mathbf{v}}_{m/n}^n = \mathbf{R}_b^n(\Theta)[\mathbf{a}_{\text{imu}}^b - \mathbf{b}_{\text{acc}}^b] - \mathbf{g}^n \quad (11.233)$$

$$\dot{\mathbf{b}}_{\text{acc}}^b = \mathbf{0} \quad (11.234)$$

$$\mathbf{y}_1 = \mathbf{p}_{m/n}^n \quad (11.235)$$

where \mathbf{y}_1 is the GNSS measurement. A nonlinear design method for simultaneously linear and angular velocity estimation has been proposed by Hua (2010) using GNSS/INS measurements. This method discusses stability of accelerated vehicles where the linear and angular dynamics are coupled. The EKF method discussed in Section 11.3.3 can also be used for this purpose. However, Kalman filtering requires linearization of the rotation matrix and implementation of time-varying Riccati equations that suffer from singularities. Hence, care must be taken when implementing the EKF. An alternative approach is to use a nonlinear decoupled fixed-gain observer where it is assumed that the attitude signal Θ is available when estimating the linear velocity $\dot{\mathbf{v}}_{m/n}^n$. Algorithms for computation of Θ are presented in Section 11.5.2.

Consider the following nonlinear observer for linear velocity (see Figure 11.17):

$$\dot{\hat{\mathbf{p}}}_{m/n}^n = \hat{\mathbf{v}}_{m/n}^n + \mathbf{K}_1 \tilde{\mathbf{y}}_1 \quad (11.236)$$

$$\dot{\hat{\mathbf{v}}}_{m/n}^n = \mathbf{R}_b^n(\Theta)[\mathbf{a}_{\text{imu}}^b - \hat{\mathbf{b}}_{\text{acc}}^b] - \mathbf{g}^n + \mathbf{K}_2 \tilde{\mathbf{y}}_1 \quad (11.237)$$

$$\dot{\hat{\mathbf{b}}}_{\text{acc}}^b = \mathbf{K}_3 \mathbf{R}_b^n(\Theta)^T \tilde{\mathbf{y}}_1 \quad (11.238)$$

$$\tilde{\mathbf{y}}_1 = \hat{\mathbf{p}}_{m/n}^n \quad (11.239)$$

where $\tilde{\mathbf{y}}_1 = \mathbf{y}_1 - \hat{\mathbf{y}}_1 = \mathbf{p}_{m/n}^n - \hat{\mathbf{p}}_{m/n}^n$ is the injection term. The linear velocity estimate $\hat{\mathbf{v}}_{m/n}^n$ expressed in $\{n\}$ is transformed to $\{b\}$ using the rotation matrix

$$\hat{\mathbf{v}}_{m/n}^b = \mathbf{R}_b^n(\Theta) \hat{\mathbf{v}}_{m/n}^n \quad (11.240)$$

The observer error dynamics becomes

$$\begin{bmatrix} \dot{\tilde{\mathbf{p}}}_{m/n}^n \\ \dot{\tilde{\mathbf{v}}}_{m/n}^n \\ \dot{\tilde{\mathbf{b}}}_{\text{acc}}^b \end{bmatrix} = \begin{bmatrix} -\mathbf{K}_1 & \mathbf{I} & \mathbf{0} \\ -\mathbf{K}_2 & \mathbf{0} & -\mathbf{R}_b^n(\Theta) \\ -\mathbf{K}_3 \mathbf{R}_b^n(\Theta)^T & \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{p}}_{m/n}^n \\ \tilde{\mathbf{v}}_{m/n}^n \\ \tilde{\mathbf{b}}_{\text{acc}}^b \end{bmatrix} \quad (11.241)$$

‡

$$\dot{\mathbf{x}} = \mathbf{A}(\Theta) \mathbf{x} \quad (11.242)$$

The gains \mathbf{K}_1 , \mathbf{K}_2 and \mathbf{K}_3 can be chosen such that \mathbf{x} converges exponentially to zero. This is not straightforward since the matrix $\mathbf{A}(\Theta)$ depends on the attitude vector Θ and thus becomes time varying.

For marine craft a practical solution to this problem can be found by noticing that the angular rates $\omega_{m/n}^b = \omega_{b/n}^b = [p, q, r]^\top$ are quite small. This is the key assumption in order to apply the result of Lindegaard and Fossen (2001a). Consider the transformation

$$\mathbf{x} = \mathbf{T}(\Theta)\mathbf{z} \quad (11.243)$$

where $\mathbf{T}(\Theta)$ is a transformation matrix

$$\mathbf{T}(\Theta) = \text{diag}\{\mathbf{R}_b^n(\Theta), \mathbf{R}_b^n(\Theta), \mathbf{I}\} \quad (11.244)$$

satisfying $\mathbf{T}(\Theta)^{-1} = \mathbf{T}(\Theta)^\top$. Hence,

$$\begin{aligned} \dot{\mathbf{z}} &= \mathbf{T}(\Theta)^\top \dot{\mathbf{x}} + \dot{\mathbf{T}}(\Theta)^\top \mathbf{x} \\ &= \mathbf{T}(\Theta)^\top \mathbf{A}(\Theta)\mathbf{x} + \dot{\mathbf{T}}(\Theta)^\top \mathbf{x} \\ &= \mathbf{T}(\Theta)^\top \mathbf{A}(\Theta)\mathbf{T}(\Theta)\mathbf{z} + \dot{\mathbf{T}}(\Theta)^\top \mathbf{T}(\Theta)\mathbf{z} \end{aligned} \quad (11.245)$$

Stability of this system can be guaranteed under the assumption that $\dot{\mathbf{T}}(\Theta)$ is sufficiently small. This is indeed satisfied if the angular rate vector $\omega_{m/n}^b$ of the craft is small. Moreover,

$$\begin{aligned} \dot{\mathbf{T}}(\Theta) &= \text{diag}\{\mathbf{R}_b^n(\Theta)\mathbf{S}(\omega_{m/n}^b), \mathbf{R}_b^n(\Theta)\mathbf{S}(\omega_{m/n}^b), \mathbf{0}\} \\ &\approx \mathbf{0} \end{aligned} \quad (11.246)$$

if $\mathbf{S}(\omega_{m/n}^b) \approx \mathbf{0}$. This is a good assumption for a marine craft since $\omega_{m/n}^b$ is quite small during rolling, pitching and yawing. Hence, from a practical point of view it is sufficient to check stability of the system:

$$\dot{\mathbf{z}} = \mathbf{T}(\Theta)^\top \mathbf{A}(\Theta)\mathbf{T}(\Theta)\mathbf{z} \quad (11.247)$$

A pole-placement algorithm can be derived by using the following property:

Property 11.1 (Commuting Matrices)

A matrix $\mathbf{K} \in \mathbb{R}^{3 \times 3}$ is said to commute with the rotation matrix $\mathbf{R}_b^n(\Theta)$ if

$$\mathbf{K}\mathbf{R}_b^n(\Theta) = \mathbf{R}_b^n(\Theta)\mathbf{K} \quad (11.248)$$

Examples of \mathbf{K} matrices satisfying Property 11.1 are linear combinations:

$$\mathbf{K} = a_1 \mathbf{R}_b^n(\Theta) + a_2 \mathbf{I} + a_3 \mathbf{k}\mathbf{k}^\top \quad (11.249)$$

where

$$\mathbf{k} = [0, 0, 1]^\top \quad (11.250)$$

is the axis of rotation and a_i ($i = 1, \dots, 3$) are scalars.

If the observer gain matrices \mathbf{K}_i ($i = 1, \dots, 3$) are chosen to commute with the rotation matrix $\mathbf{R}_b^n(\Theta)$, Property 11.1 implies that the error dynamics (11.247) can be written

$$\dot{\mathbf{z}} = \mathbf{A}\mathbf{z} \quad (11.251)$$

where A is a constant system matrix:

$$A = \begin{bmatrix} -K_1 & I & \mathbf{0} \\ -K_2 & \mathbf{0} & -I \\ -K_3 & \mathbf{0} & \mathbf{0} \end{bmatrix} \quad (11.252)$$

Then it follows that

$$A = T(\Theta)^\top A(\Theta) T(\Theta) \quad (11.253)$$

One way to satisfy this is to choose the matrices K_i with the following diagonal structure:

$$K_i = \text{diag}\{k_i, k_i, l_i\}, \quad i = 1, 2, 3 \quad (11.254)$$

where surge and sway have the same gains $k_i > 0$ and heave can be tuned independently by $l_i > 0$. This clearly satisfies (11.249) since $a_1 = 0$, $a_2 > 0$ and $a_3 > 0$. Hence, stability can be checked by computing the eigenvalues of A since the eigenvalues of A and $A(\Theta)$ are equal for all Θ . A necessary condition for exponential stability is that the eigenvalues of A lie in the left half-plane, that is A must be *Hurwitz*.

Integration of IMU and GNSS Position and Velocity Measurements

It is straightforward to modify the observer (11.236)–(11.239) to include GNSS velocity measurements, $y_2 = v_{m/n}$. Moreover,

$$\dot{\tilde{p}}_{m/n}^n = \hat{v}_{m/n}^n + K_{11}\tilde{y}_1 + K_{21}\tilde{y}_2 \quad (11.255)$$

$$\dot{\tilde{v}}_{m/n}^n = R_b^n(\Theta)[a_{\text{imu}}^b - \hat{b}_{\text{acc}}^b] - g^n + K_{12}\tilde{y}_1 + K_{22}\tilde{y}_2 \quad (11.256)$$

$$\dot{\tilde{b}}_{\text{acc}}^b = K_{13}R_b^n(\Theta)^\top \tilde{y}_1 + K_{23}R_b^n(\Theta)^\top \tilde{y}_2 \quad (11.257)$$

$$\hat{y}_1 = \hat{p}_{m/n}^n \quad (11.258)$$

$$\hat{y}_2 = \hat{v}_{m/n}^n \quad (11.259)$$

where $\tilde{y}_i = y_i - \hat{y}_i$ ($i = 1, 2$) results in the error dynamics

$$\begin{bmatrix} \dot{\tilde{p}}_{m/n}^n \\ \dot{\tilde{v}}_{m/n}^n \\ \dot{\tilde{b}}_{\text{acc}}^b \end{bmatrix} = \begin{bmatrix} -K_{11} & I - K_{21} & \mathbf{0} \\ -K_{12} & -K_{22} & -R_b^n(\Theta) \\ -K_{13}R_b^n(\Theta)^\top & -K_{23}R_b^n(\Theta)^\top & \mathbf{0} \end{bmatrix} \begin{bmatrix} \tilde{p}_{m/n}^n \\ \tilde{v}_{m/n}^n \\ \tilde{b}_{\text{acc}}^b \end{bmatrix} \Downarrow \dot{x} = A(\Theta)x \quad (11.260)$$

Choosing the gains \mathbf{K}_{ij} ($i = 1, 2, 3, j = 1, 2$) according to Property 11.1 such that they commute with $\mathbf{R}_b^n(\Theta)$ and assuming that the angular rate vector $\omega_{m/n}^b$ is small, gives the following error dynamics:

$$\begin{aligned}\dot{\mathbf{z}} &= \mathbf{T}^\top(\Theta) \mathbf{A} \mathbf{T}(\Theta) \mathbf{z} \\ &= \mathbf{A} \mathbf{z}\end{aligned}\quad (11.261)$$

where

$$\mathbf{A} = \begin{bmatrix} -\mathbf{K}_{11} & \mathbf{I} - \mathbf{K}_{21} & \mathbf{0} \\ -\mathbf{K}_{12} & -\mathbf{K}_{22} & -\mathbf{I} \\ -\mathbf{K}_{13} & -\mathbf{K}_{23} & \mathbf{0} \end{bmatrix} \quad (11.262)$$

Hence, exponential convergence of \mathbf{z} to zero is guaranteed if the gains \mathbf{K}_{ij} are chosen such that \mathbf{A} is *Hurwitz*.

11.5.2 Accelerometer and Compass Aided Attitude Observer

A nonlinear attitude observer can be designed by integrating the gyro measurements ω_{imu} to obtain an estimate of the quaternions $\hat{\mathbf{q}}$. The quaternion estimate is corrected by approximating \mathbf{q} using accelerometer and compass measurements (see Figure 11.18). The attitude observer in this section can be viewed as a special case of Vik and Fossen (2001) where the Earth rotation is neglected. An attitude observer can also be derived using the EKF algorithm. However, the nonlinear representation is highly advantageous from an implementation point of view since it avoids numerical integration of a large number of the Kalman filter Riccati equations.

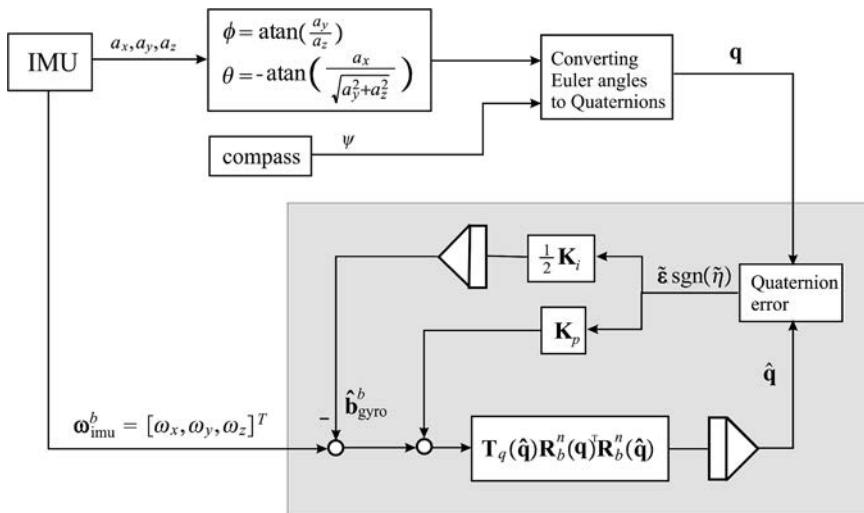


Figure 11.18 Block diagram showing the nonlinear attitude observer with the IMU acceleration mapping.

Mapping from Linear Accelerations to Roll and Pitch Angles

Before designing the attitude observer, it is necessary to map the three-axis linear IMU accelerations to roll and pitch angles. The principle for this is that the angle between the acceleration and gravity vectors can be computed using trigonometry. This is a static mapping that suffers from inaccuracies when performing high-acceleration maneuvers. For ships this works quite well but aircraft and other highly maneuverable vehicles should use other methods. The static acceleration mapping, together with a magnetometer, is used to construct a Euler angle measurement vector Θ , which again is used to compute q as illustrated in Figure 11.18.

The IMU acceleration measurements

$$\mathbf{a}_{\text{imu}}^b = \mathbf{R}_n^b(\Theta)(\dot{\mathbf{v}}_{m/n}^n + \mathbf{g}^n) + \mathbf{b}_{\text{acc}}^b + \mathbf{w}_{\text{acc}}^b \quad (11.263)$$

can be transformed to roll and pitch angles by noticing that for three orthogonal accelerometers onboard a craft at rest, $\dot{\mathbf{v}}_{m/n}^n = \mathbf{0}$, the measurement equation is

$$\mathbf{a}_{\text{imu}}^b = \mathbf{R}_n^b(\Theta)\mathbf{g}^n + \mathbf{b}_{\text{acc}}^b + \mathbf{w}_{\text{acc}}^b \quad (11.264)$$

The initial accelerometer biases $\mathbf{b}_{\text{acc}}^b$ are usually removed by calibrating the accelerometer in a laboratory for varying temperatures. This can be implemented as a look-up table in combination with a temperature sensor. It is also necessary to remove the dynamic drift and measurement noise $\mathbf{w}_{\text{acc}}^b$ by low-pass filtering the accelerometer measurements properly before the roll and pitch angles are computed. The key assumption is to assume that the average acceleration with respect to the environment during some period of time is zero, for instance 10–20 seconds. For aircraft this assumption does not hold since they can generate significant accelerations lasting longer than the maximum time.

After calibration and filtering, this suggests that

$$\mathbf{a}_{\text{imu}}^b \approx \mathbf{R}_n^b(\Theta)\mathbf{g}^n \quad (11.265)$$

‡

$$\begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix} \approx \mathbf{R}_n^b(\Theta) \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} = \begin{bmatrix} -g \sin(\theta) \\ g \cos(\theta) \sin(\phi) \\ g \cos(\theta) \cos(\phi) \end{bmatrix} \quad (11.266)$$

Taking the ratios

$$\frac{a_y}{a_z} \approx \tan(\phi), \quad \frac{a_x}{g} \approx -\sin(\theta), \quad \frac{a_y^2 + a_z^2}{g^2} \approx \cos^2(\theta) \quad (11.267)$$

this gives

$$\phi \approx \text{atan}\left(\frac{a_y}{a_z}\right) \quad (11.268)$$

$$\theta \approx -\text{atan}\left(\frac{a_x}{\sqrt{a_y^2 + a_z^2}}\right) \quad (11.269)$$

Notice that the transformation is singular for $\phi = \pm 90$ degrees. When combined with a compass measuring the heading ψ the attitude vector $\Theta = [\phi, \theta, \psi]^\top$ is completely determined. The Euler angles Θ can

easily be transformed to unit quaternions $\mathbf{q} = [\eta, \varepsilon_1, \varepsilon_2, \varepsilon_3]^\top$ by using Algorithm 2.2 in Section 2.2.3. The quaternion representation is advantageous when implementing the attitude observer in a computer.

Attitude Observer

In Section 2.2.2 the unit quaternion differential equation was written

$$\dot{\mathbf{q}} = \mathbf{T}_q(\mathbf{q})\boldsymbol{\omega}_{m/n}^b \quad (11.270)$$

with

$$\mathbf{T}_q(\mathbf{q}) = \frac{1}{2} \begin{bmatrix} -\boldsymbol{\varepsilon}^\top \\ \eta \mathbf{I} + \mathbf{S}(\boldsymbol{\varepsilon}) \end{bmatrix} \quad (11.271)$$

From the gyro measurement equation (11.220) it follows that

$$\boldsymbol{\omega}_{m/n}^b = \boldsymbol{\omega}_{\text{imu}}^b - \mathbf{b}_{\text{gyro}}^b - \mathbf{w}_{\text{gyro}}^b \quad (11.272)$$

Consequently,

$$\dot{\mathbf{q}} = \mathbf{T}_q(\mathbf{q}) [\boldsymbol{\omega}_{\text{imu}}^b - \mathbf{b}_{\text{gyro}}^b - \mathbf{w}_{\text{gyro}}^b] \quad (11.273)$$

$$\dot{\mathbf{b}}_{\text{gyro}}^b = \mathbf{0} \quad (11.274)$$

where $\mathbf{b}_{\text{gyro}}^b$ is the gyro bias. The nonlinear attitude observer of Salcudean (1991) has been extended to include gyro bias estimation by Vik *et al.* (1999), Vik (2000) and Vik and Fossen (2001):

$$\dot{\hat{\mathbf{q}}} = \mathbf{T}_q(\hat{\mathbf{q}}) \underbrace{\mathbf{R}_b^n(\mathbf{q})^\top \mathbf{R}_b^n(\hat{\mathbf{q}})}_{\mathbf{R}^\top(\tilde{\mathbf{q}})} [\boldsymbol{\omega}_{\text{imu}}^b - \hat{\mathbf{b}}_{\text{gyro}}^b + \mathbf{K}_p \tilde{\varepsilon} \text{sgn}(\tilde{\eta})] \quad (11.275)$$

$$\dot{\hat{\mathbf{b}}}_{\text{gyro}}^b = -\frac{1}{2} \mathbf{K}_i \tilde{\varepsilon} \text{sgn}(\tilde{\eta}) \quad (11.276)$$

where $\mathbf{K}_p = \mathbf{K}_p^\top > 0$ and $\mathbf{K}_i = \mathbf{K}_i^\top > 0$ are tunable gain matrices. The observer structure is shown in Figure 11.18.

The quaternion estimation error is defined as

$$\tilde{\mathbf{q}} := \hat{\mathbf{q}}^* \otimes \mathbf{q} \quad (11.277)$$

where $\mathbf{q} = [\eta, \varepsilon_1, \varepsilon_2, \varepsilon_3]^\top$ and $\hat{\mathbf{q}}^* = [\hat{\eta}, -\hat{\varepsilon}_1, -\hat{\varepsilon}_2, -\hat{\varepsilon}_3]^\top$ is the conjugate of $\hat{\mathbf{q}}$ corresponding to multiplying the vector $\hat{\boldsymbol{\varepsilon}} = [\hat{\varepsilon}_1, \hat{\varepsilon}_2, \hat{\varepsilon}_3]^\top$ with -1 . The symbol \otimes denotes the *quaternion product*, which is defined as (Chou, 1992)

$$\begin{aligned} \mathbf{q}_1 \otimes \mathbf{q}_2 &:= \begin{bmatrix} \eta_1 \eta_2 - \boldsymbol{\varepsilon}_1^\top \boldsymbol{\varepsilon}_2 \\ \eta_2 \boldsymbol{\varepsilon}_1 + \eta_1 \boldsymbol{\varepsilon}_2 + \boldsymbol{\varepsilon}_1 \times \boldsymbol{\varepsilon}_2 \end{bmatrix} \\ &= \begin{bmatrix} \eta_1 & -\boldsymbol{\varepsilon}_1^\top \\ \boldsymbol{\varepsilon}_1 & \eta_1 \mathbf{I} + \mathbf{S}(\boldsymbol{\varepsilon}_1) \end{bmatrix} \mathbf{q}_2 \end{aligned} \quad (11.278)$$

This yields

$$\tilde{\mathbf{q}} = \begin{bmatrix} \hat{\eta}\eta + \hat{\boldsymbol{\epsilon}}^\top \boldsymbol{\epsilon} \\ -\eta\hat{\boldsymbol{\epsilon}} + \hat{\eta}\boldsymbol{\epsilon} - S(\hat{\boldsymbol{\epsilon}})\boldsymbol{\epsilon} \end{bmatrix} \quad (11.279)$$

Notice that $\tilde{\mathbf{q}} \neq \mathbf{q} - \hat{\mathbf{q}}$. After some tedious calculations, it can be shown that the observer error dynamics becomes

$$\dot{\tilde{\mathbf{q}}} = -\mathbf{T}(\tilde{\mathbf{q}}) [\tilde{\mathbf{b}}_{\text{gyro}}^b + \mathbf{K}_p \tilde{\boldsymbol{\epsilon}} \text{sgn}(\tilde{\eta})] \quad (11.280)$$

$$\dot{\tilde{\mathbf{b}}}_{\text{gyro}}^b = \frac{1}{2} \mathbf{K}_i \tilde{\boldsymbol{\epsilon}} \text{sgn}(\tilde{\eta}) \quad (11.281)$$

From (11.271) it is seen that the expression for $\dot{\tilde{\mathbf{q}}}$ can be written as

$$\dot{\tilde{\mathbf{q}}} = -\frac{1}{2} \begin{bmatrix} -\tilde{\boldsymbol{\epsilon}}^\top \\ \tilde{\eta}\mathbf{I} + S(\tilde{\boldsymbol{\epsilon}}) \end{bmatrix} [\tilde{\mathbf{b}}_{\text{gyro}}^b + \mathbf{K}_p \tilde{\boldsymbol{\epsilon}} \text{sgn}(\tilde{\eta})] \quad (11.282)$$

or

$$\dot{\tilde{\eta}} = \frac{1}{2} \tilde{\boldsymbol{\epsilon}}^\top [\tilde{\mathbf{b}}_{\text{gyro}}^b + \mathbf{K}_p \tilde{\boldsymbol{\epsilon}} \text{sgn}(\tilde{\eta})] \quad (11.283)$$

$$\dot{\tilde{\boldsymbol{\epsilon}}} = -\frac{1}{2} [\tilde{\eta}\mathbf{I} + S(\tilde{\boldsymbol{\epsilon}})] [\tilde{\mathbf{b}}_{\text{gyro}}^b + \mathbf{K}_p \tilde{\boldsymbol{\epsilon}} \text{sgn}(\tilde{\eta})] \quad (11.284)$$

Since the measurement noise $E(\mathbf{w}_{\text{gyro}}^b) = \mathbf{0}$, the term $\mathbf{w}_{\text{gyro}}^b$ is neglected in the Lyapunov analysis. Consider

$$V = \frac{1}{2} (\tilde{\mathbf{b}}_{\text{gyro}}^b)^\top \mathbf{K}_i^{-1} \tilde{\mathbf{b}}_{\text{gyro}}^b + H(\tilde{\eta}) \quad (11.285)$$

where different candidates for $H(\tilde{\eta})$ are found in Table 11.4 (Fjellstad and Fossen, 1994). Notice that the function $H(\tilde{\eta})$ is Lipschitz. Hence, time differentiation of V along the trajectories of $\dot{\tilde{\mathbf{b}}}_{\text{gyro}}^b$ and $\dot{\tilde{\eta}}$ yields

$$\dot{V} = (\tilde{\mathbf{b}}_{\text{gyro}}^b)^\top \mathbf{K}_i^{-1} \dot{\tilde{\mathbf{b}}}_{\text{gyro}}^b + \frac{\partial H(\tilde{\eta})}{\partial \tilde{\eta}} \dot{\tilde{\eta}} \quad (11.286)$$

Choosing $H(\tilde{\eta}) = 1 - |\tilde{\eta}|$ such that $\partial H(\tilde{\eta})/\partial \tilde{\eta} = -\text{sgn}(\tilde{\eta})$ (see the first row in Table 11.4) yields

$$\begin{aligned} \dot{V} &= \frac{1}{2} (\tilde{\mathbf{b}}_{\text{gyro}}^b)^\top \tilde{\boldsymbol{\epsilon}} \text{sgn}(\tilde{\eta}) - \text{sgn}(\tilde{\eta}) \frac{1}{2} \tilde{\boldsymbol{\epsilon}}^\top [\tilde{\mathbf{b}}_{\text{gyro}}^b + \mathbf{K}_p \tilde{\boldsymbol{\epsilon}} \text{sgn}(\tilde{\eta})] \\ &= -\frac{1}{2} \tilde{\boldsymbol{\epsilon}}^\top \mathbf{K}_p \tilde{\boldsymbol{\epsilon}} \leq 0 \end{aligned} \quad (11.287)$$

Thienel and Sanner (2003) have proven that the equilibrium points $\tilde{\mathbf{q}} = [\pm 1, 0, 0, 0]^\top$ of the attitude observer error dynamics is asymptotically stable if the pair $(\mathbf{A}(t), \mathbf{C})$ is uniformly completely observable

Table 11.4 Alternative choices of attitude update laws

$H(\tilde{\eta})$	Update law	Stable equation	Unstable equation
$1 - \tilde{\eta} $	$-\mathbf{K}_p \tilde{\boldsymbol{\epsilon}} \text{sgn}(\tilde{\eta})$	$\tilde{\eta} = \pm 1$	
$1 - \tilde{\eta}$	$-\mathbf{K}_p \tilde{\boldsymbol{\epsilon}}$	$\tilde{\eta} = 1$	$\tilde{\eta} = -1$
$1 + \tilde{\eta}$	$\mathbf{K}_p \tilde{\boldsymbol{\epsilon}}$	$\tilde{\eta} = -1$	$\tilde{\eta} = 1$

for the system

$$\dot{\mathbf{x}} = \mathbf{A}(t)\mathbf{x} \quad (11.288)$$

where $\mathbf{x} = [\tilde{\boldsymbol{\epsilon}}^\top, (\tilde{\mathbf{b}}_{\text{gyro}}^b)^\top]^\top$ and

$$\mathbf{A}(t) = \begin{bmatrix} -\frac{1}{2}\mathbf{K}_p \text{sgn}(\tilde{\eta}) [\tilde{\eta}\mathbf{I} + \mathbf{S}(\tilde{\boldsymbol{\epsilon}})] & -\frac{1}{2} [\tilde{\eta}\mathbf{I} + \mathbf{S}(\tilde{\boldsymbol{\epsilon}})] \\ \frac{1}{2}\text{sgn}(\tilde{\eta})\mathbf{I} & \mathbf{0} \end{bmatrix} \quad (11.289)$$

Hence,

$$\begin{aligned} \dot{V} &= -\mathbf{x}^\top \mathbf{C}^\top \mathbf{C} \mathbf{x} \\ &\leq 0 \end{aligned} \quad (11.290)$$

where

$$\mathbf{C} = \left[\sqrt{\frac{1}{2}\mathbf{K}_i}, \mathbf{0} \right] \quad (11.291)$$

This corresponds to a persistency of excitation (PE) argument ensuring exponential convergence of the estimation error $\tilde{\mathbf{b}}_{\text{gyro}}^b$ to zero. Global results cannot be obtained since this system has two equilibrium points. This is a well-known topological limitation as described by Bhat and Bernstein (2000).

Vertical Reference Unit (VRU)

The special solution of the observer when only ϕ and θ are estimated (no compass measurement) is referred to as a *vertical reference unit* (VRU). The performance of state-of-the-art VRUs has been evaluated by Ingram *et al.* (1996).

A VRU is particularly useful if you want to transform the GNSS position and velocity measurements

$$\mathbf{p}_{\text{gnss}}^n = [N_{\text{gnss}}, E_{\text{gnss}}, D_{\text{gnss}}]^\top \quad (11.292)$$

$$\mathbf{v}_{\text{gnss}}^n = [\dot{N}_{\text{gnss}}, \dot{E}_{\text{gnss}}, \dot{D}_{\text{gnss}}]^\top \quad (11.293)$$

for a GNSS receiver located at the position $\mathbf{r}_{\text{gnss}}^b = [x_{\text{gnss}}, y_{\text{gnss}}, z_{\text{gnss}}]^\top$ to the IMU coordinate system $\{m\}$. The NED position $\mathbf{p}_{m/n}^n$ and linear velocity $\mathbf{v}_{m/n}^n$ of the craft are found as

$$\mathbf{p}_{m/n}^n = \mathbf{p}_{\text{gnss}}^n - \mathbf{R}_b^n(\Theta)\mathbf{r}_{\text{gnss}}^b \quad (11.294)$$

$$\mathbf{v}_{m/n}^n = \mathbf{v}_{\text{gnss}}^n - \mathbf{R}_b^n(\Theta)\mathbf{S}(\omega_{b/n}^b)\mathbf{r}_{\text{gnss}}^b \quad (11.295)$$

since $\mathbf{S}(\omega_{b/n}^b) = \mathbf{S}(\omega_{m/n}^b)$. The expression for the NED velocity (11.295) makes use of $\dot{\mathbf{r}}_{\text{gnss}}^b = \mathbf{0}$; that is the position of the GNSS receiver is constant when mounted onboard a rigid craft. Consequently,

$$\mathbf{v}_{m/n}^n = \dot{\mathbf{p}}_{m/n}^n = \dot{\mathbf{p}}_{\text{gnss}}^n - \dot{\mathbf{R}}_b^n(\Theta)\mathbf{r}_{\text{gnss}}^b \quad (11.296)$$

where $\dot{\mathbf{p}}_{\text{gnss}}^n = \mathbf{v}_{\text{gnss}}^n$ and $\dot{\mathbf{R}}_b^n(\Theta) = \mathbf{R}_b^n(\Theta)\mathbf{S}(\omega_{b/n}^b)$; see Theorem 2.2 in Section 2.2.1.

11.5.3 Attitude Observer using Gravitational and Magnetic Field Directions

The attitude observer developed in Section 11.5.2 suffers from the assumption that $\dot{\mathbf{v}}_{m/n}^n = \mathbf{0}$ when computing the roll and pitch angles from accelerometer measurements. Hence, inaccuracies propagate

to \mathbf{q} and thus the observer injection term. An alternative approach could be to use accelerometer and magnetometer measurements directly to update the estimator. Mahony *et al.* (2008) have derived a nonlinear attitude observer with an injection term that compares the directions of the measurement vectors when transformed from the body-fixed frame to the inertial frame using true and estimated attitude.

Let \mathbf{v}_{0i}^n ($i = 1, \dots, n$) denote a set of n known inertial directions. This vector can be expressed in $\{b\}$ by using the quaternion rotation matrix

$$\mathbf{v}_i^b = \mathbf{R}_n^b(\mathbf{q})\mathbf{v}_{0i}^n + \mathbf{w}_i \quad (11.297)$$

where \mathbf{w}_i ($i = 1, \dots, n$) are zero-mean noise processes. Since only the direction of the measurement is relevant to the observer, it is assumed that all measurements are normalized such that $\|\mathbf{v}_{0i}^n\| = 1$. The associated estimate of \mathbf{v}_i^b is computed as

$$\hat{\mathbf{v}}_i^b = \mathbf{R}_n^b(\hat{\mathbf{q}})\mathbf{v}_{0i}^n \quad (11.298)$$

where $\hat{\mathbf{q}}$ is the estimate of the quaternion vector. Hence, the difference between the two signals (11.297) and (11.298) is zero if $\hat{\mathbf{q}} = \mathbf{q}$.

The attitude observer can be implemented using normalized acceleration and magnetometer measurements according to

$$\mathbf{v}_1^b = \frac{\mathbf{a}_{\text{imu}}^b}{\|\mathbf{a}_{\text{imu}}^b\|}, \quad \mathbf{v}_2^b = \frac{\mathbf{m}_{\text{mag}}^b}{\|\mathbf{m}_{\text{mag}}^b\|} \quad (11.299)$$

where the acceleration $\mathbf{a}_{\text{imu}}^b$ and magnetometer $\mathbf{m}_{\text{mag}}^b$ measurements are given by (11.219) and (11.221), respectively.

Explicit Complementary Filter with Bias Correction

The quaternion representation of the Mahony *et al.* (2008) attitude observer can be expressed as (see Figure 11.19)

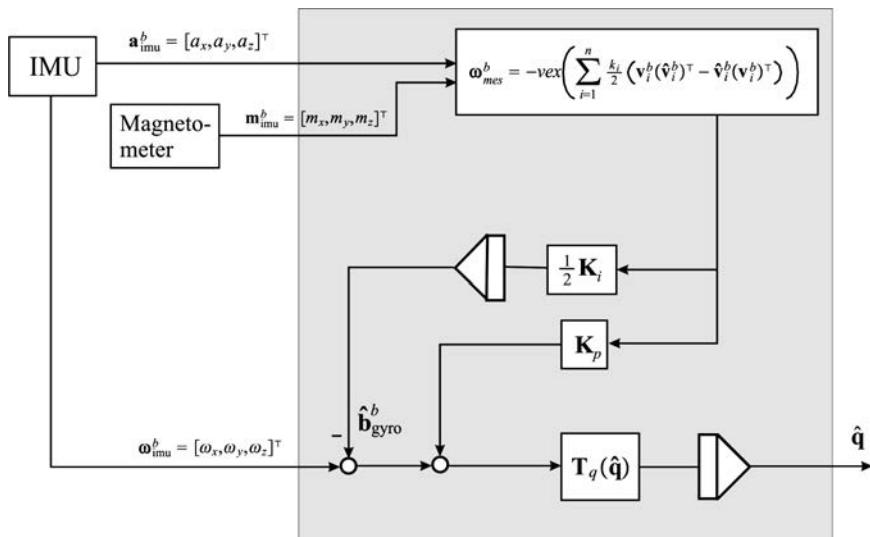


Figure 11.19 Nonlinear attitude observer-based directional measurements.

$$\boldsymbol{\omega}_{\text{mes}}^b = -\text{vex} \left(\sum_{i=1}^n \frac{k_i}{2} (\mathbf{v}_i^b (\hat{\mathbf{v}}_i^b)^\top - \hat{\mathbf{v}}_i^b (\mathbf{v}_i^b)^\top) \right) \quad (11.300)$$

$$\dot{\hat{\mathbf{q}}} = \mathbf{T}_q(\hat{\mathbf{q}}) [\boldsymbol{\omega}_{\text{imu}}^b - \hat{\mathbf{b}}_{\text{gyro}}^b + \mathbf{K}_p \boldsymbol{\omega}_{\text{mes}}^b] \quad (11.301)$$

$$\dot{\hat{\mathbf{b}}}_{\text{gyro}}^b = -\frac{1}{2} \mathbf{K}_i \boldsymbol{\omega}_{\text{mes}}^b \quad (11.302)$$

where k_i is a tunable gain and the operator $\text{vex}: \text{SO}(3) \rightarrow \mathbb{R}^3$ denotes the inverse of the cross-product operator $\mathbf{S}(\mathbf{a})$. Moreover,

$$\mathbf{a} \times \mathbf{b} = \mathbf{S}(\mathbf{a})\mathbf{b} \quad (11.303)$$

$$\text{vex}(\mathbf{S}(\mathbf{a})) = \mathbf{a} \quad (11.304)$$

Hence, the following expression corresponding to (11.300) can be derived:

$$\text{vex}(\mathbf{ab}^T - \mathbf{ba}^\top) = \begin{bmatrix} a_3 b_2 - a_2 b_3 \\ a_1 b_3 - a_3 b_1 \\ a_2 b_1 - a_1 b_2 \end{bmatrix} \quad (11.305)$$

The estimate $\hat{\mathbf{v}}_1^b$ corresponding to the accelerometer measurement is computed according to

$$\begin{aligned} \hat{\mathbf{v}}_1^b &= \mathbf{R}_n^b(\hat{\mathbf{q}})\mathbf{v}_{01}^n \\ &= \mathbf{R}_n^b(\hat{\mathbf{q}}) \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \end{aligned} \quad (11.306)$$

where the known inertial direction of gravity $\mathbf{v}_{01}^n = \mathbf{g}^n / \|\mathbf{g}^n\|$ is exploited.

The magnetometer measures three inertial components:

$$\mathbf{m}_{\text{mag}}^b = [m_x, m_y, m_z]^\top \quad (11.307)$$

when the sensor unit is mounted in a craft at rest to sense the components of the Earth's magnetic field. Consequently, the normalized magnetometer estimate in $\{b\}$ becomes

$$\begin{aligned} \hat{\mathbf{v}}_2^b &= \mathbf{R}_n^b(\hat{\mathbf{q}})\mathbf{v}_{02}^n \\ &= \mathbf{R}_n^b(\hat{\mathbf{q}}) \frac{1}{\sqrt{m_x^2 + m_y^2 + m_z^2}} \begin{bmatrix} m_x \\ m_y \\ m_z \end{bmatrix} \end{aligned} \quad (11.308)$$

Stability Analysis

The stability proof of Mahony *et al.* (2008) assumes that the reference vectors \mathbf{v}_{0i}^n are constant. This has been relaxed by Grip *et al.* (2011) using the projection algorithm. From this work it is concluded that the equilibrium point of the quaternion error dynamics is semi-global exponential stable for $n \geq 2$ independent inertial directions \mathbf{v}_{0i}^n .

12

Motion Control Systems

Motion control systems for marine craft have been an active field of research since the first mechanical autopilot was constructed by *Elmer Sperry* in 1911. Modern control systems are based on a variety of design techniques such as PID control, linear quadratic optimal and stochastic control, \mathcal{H}_∞ control methods, fuzzy systems, neural networks and nonlinear control theory, to mention only some. In the first part of the book, models for simulation of marine craft were presented. In this chapter, dynamic models are used to design model-based control systems. The dynamic properties and limitation of the craft are incorporated into the design process to obtain robust performance. Many of the presented design methods have been successfully implemented and tested onboard ships, underwater vehicles and floating vessels.

Chapter 12 covers state-of-the-art PID control methods for setpoint regulation, trajectory-tracking control and path-following control of marine craft. This includes autopilot design, stationkeeping, position mooring systems, cross-tracking control systems and LOS control systems. In addition to this, control allocation methods are discussed. Advance methods such as linear quadratic optimal control, sliding mode control, state feedback linearization and integrator backstepping are discussed in Chapter 13.

Preview of the Chapter

This chapter starts with open-loop analysis and maneuverability (Section 12.1) followed by state-of-the-art linear PID design methods (Section 12.2). Conventional PID control systems have their origin from SISO linear systems theory. However, it is possible to generalize this to nonlinear MIMO systems by using results from robotics (Fossen, 1991). This requires that the marine craft equations of motion are expressed in a vectorial setting:

$$\dot{\eta} = J_\Theta(\eta)v \quad (12.1)$$

$$M\ddot{v} + C(v)v + D(v)v + g(\eta) = \tau + w \quad (12.2)$$

For this model class, MIMO nonlinear PID control systems can be designed by exploiting the fact that the mass matrix is positive definite and constant ($M = M^\top > 0$, $\dot{M} = 0$), the Coriolis and centripetal matrix $C(v) = -C^\top(v)$ is skew-symmetrical and the damping matrix $D(v) > 0$ is strictly positive.

12.1 Open-Loop Stability and Maneuverability

When designing a motion control system a compromise between stability and maneuverability must be made. More specifically:



Figure 12.1 Maneuverability versus stability. Illustration by Bjarne Stenberg.

- *Stability* of an uncontrolled marine craft can be defined as the ability to return to an equilibrium point after a disturbance, without any corrective action of the actuators.
- *Maneuverability*, on the other hand, is defined as the capability of the craft to carry out specific maneuvers.

It is well known that a craft that is easy to maneuver, for instance a fighter aircraft or a high-speed watercraft, can be marginally stable or even unstable in open loop. On the other hand, excessive stability implies that the control effort will be excessive in a maneuvering situation whereas a marginally stable ship is easy to maneuver. Consequently, a compromise between stability and maneuverability must be made (see Figure 12.1).

12.1.1 Straight-Line, Directional and Positional Motion Stability

For marine craft it is common to distinguish between three types of stability, namely:

- *Straight-line stability*
- *Directional or course stability*
- *Positional motion stability*

This can be explained using open-loop and closed-loop stability analyzes. In order to understand the different types of stability one can consider the following test system:

$$\begin{aligned}\dot{x} &= u \cos(\psi) - v \sin(\psi) \\ &\approx u_0 \cos(\psi)\end{aligned}\tag{12.3}$$

$$\begin{aligned}\dot{y} &= u \sin(\psi) + v \cos(\psi) \\ &\approx u_0 \sin(\psi)\end{aligned}\quad (12.4)$$

$$\dot{\psi} = r \quad (12.5)$$

$$Tr + r = K\delta + w \quad (12.6)$$

where w is the external disturbances and $u_0 = \text{constant}$ is the cruise speed. The first two equations represent the (x, y) position of the ship while the last two equations describe the yaw dynamics modeled by Nomoto's first-order model. For simplicity, it is assumed that the yaw motion of the craft is stabilized by a PD-controlled rudder servo:

$$\delta = -K_p(\psi - \psi_d) - K_d r \quad (12.7)$$

where $\psi_d = \text{constant}$ denotes the desired heading angle and K_p and K_d are two positive regulator gains. Substituting the control law (12.7) into Nomoto's first-order model (12.6) yields the closed-loop system

$$\underbrace{T_m}_{m} \ddot{\psi} + \underbrace{(1 + KK_d)}_{d} \dot{\psi} + \underbrace{KK_p}_{k} \psi = \underbrace{KK_p \psi_d + w}_{f(t)} \quad (12.8)$$

The closed-loop system represents a second-order mass–damper–spring system

$$m\ddot{\psi} + d\dot{\psi} + k\psi = f(t) \quad (12.9)$$

with driving input

$$f(t) = k\psi_d + w \quad (12.10)$$

The eigenvalues $\lambda_{1,2}$, the natural frequency ω_n and the relative damping ratio ζ for the mass–damper–spring system are

$$\lambda_{1,2} = \frac{-d \pm \sqrt{d^2 - 4km}}{2m} \quad (12.11)$$

$$\omega_n = \sqrt{\frac{k}{m}} \quad (12.12)$$

$$\zeta = \frac{d}{2} \frac{1}{\sqrt{km}} \quad (12.13)$$

Matlab

The test system (12.8) is simulated in Matlab for varying model parameters using the MSS toolbox script

`StabDemo`

The simulation results and the stability analysis are presented on the next pages. This includes the following cases:

- Instability
- Straight-line stability
- Directional stability
- Positional motion stability

Instability: For uncontrolled marine craft ($K_p = K_d = 0$) instability occurs when

$$\lambda_1 = -\frac{d}{m} = -\frac{1}{T} > 0$$

$$\lambda_2 = 0$$

which simply states that $T < 0$. This is common for large tankers.

Straight-Line Stability: Consider an uncontrolled marine craft ($K_p = K_d = 0$) moving in a straight path. If the new path is straight after a disturbance w in yaw the craft is said to have straight-line stability. The direction of the new path will usually differ from the initial path because no restoring forces are present ($k = 0$). This corresponds to

$$\lambda_1 = -\frac{d}{m} = -\frac{1}{T} < 0$$

$$\lambda_2 = 0$$

Consequently, the requirement $T > 0$ implies straight-line stability for the uncontrolled craft ($\delta = 0$).

Directional Stability (Stability on Course): Directional stability is a much stronger requirement than straight-line stability (see Figure 12.2). Directional stability requires the final path to be parallel to the initial path that is obtained for $K_p > 0 \Rightarrow k > 0$. Additional damping is added through $K_d > 0$. This corresponds to PD control. A marine craft is said to be directionally stable if both eigenvalues have negative real parts, that is

$$\operatorname{Re}\{\lambda_{1,2}\} < 0$$

The following two types of directional stability are observed:

No oscillations ($d^2 - 4km \geq 0$): This implies that both eigenvalues are negative and real, that is $\zeta \geq 1$ such that

$$\lambda_{1,2} = \frac{-d \pm \sqrt{d^2 - 4km}}{2m} = \left(-\zeta \pm \sqrt{\zeta^2 - 1} \right) \omega_n < 0$$

For a critically damped system $\zeta = 1.0$, such that $\lambda_{1,2} = -1/2(d/m) = -\omega_n$.

Damped oscillator ($d^2 - 4km < 0$): This corresponds to two imaginary eigenvalues $\lambda_{1,2}$ with negative real parts ($\zeta < 1$), that is

$$\lambda_{1,2} = \frac{-d \pm j\sqrt{4km - dm}}{2m} = \left(-\zeta \pm j\sqrt{1 - \zeta^2} \right) \omega_n$$

Directional stability for a critically damped ($\zeta = 1.0$) and underdamped craft ($\zeta = 0.1$) is shown in Figures 12.3–12.4. Notice the oscillations in both positions and yaw angle in Figure 12.4. Directional stability requires feedback control since there are no restoring forces in yaw. However, in heave, roll

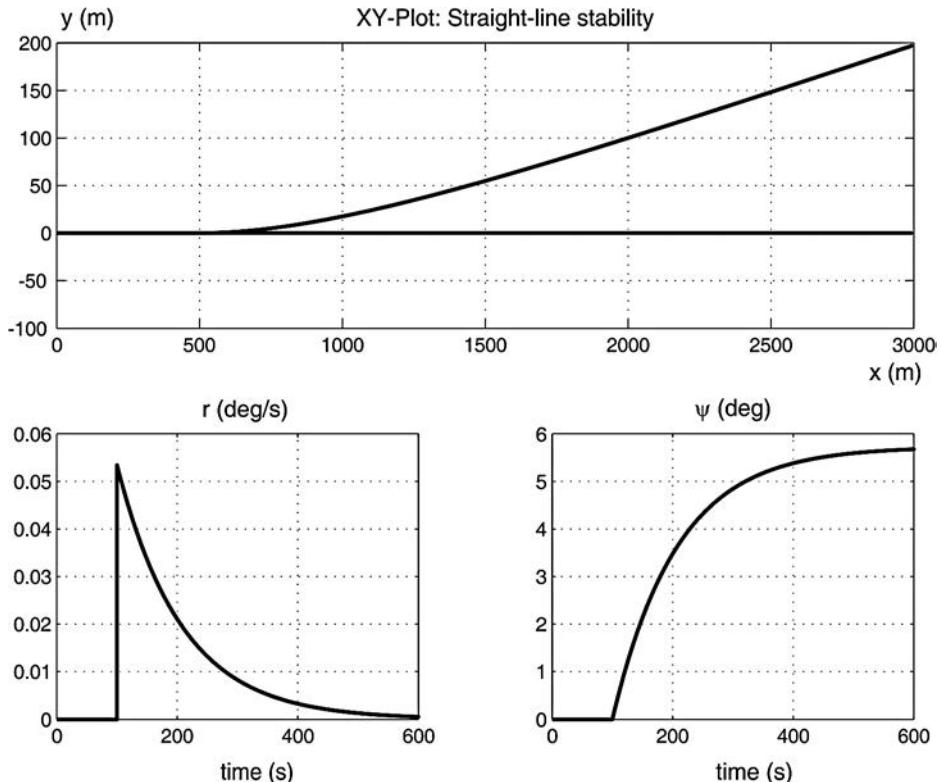


Figure 12.2 Straight-line stability for a ship when an impulse $w(t)$ is injected at $t = 100$ s.

and pitch where metacentric restoring forces are present ($k > 0$) no feedback is required to damp out the oscillations.

Positional Motion Stability: Positional motion stability implies that the ship should return to its original path after a disturbance (see Figure 12.5). This can be achieved by including integral action in the controller. Hence, a PID controller can be designed to compensate for the unknown disturbance term w while a PD controller will generally result in a steady-state offset.

Example 12.1 (Straight-Line Stability)

Consider the cargo ship and oil tanker of Example 7.1. Recall that the equivalent time constant in Nomoto's first-order model was defined as

$$T := T_1 + T_2 - T_3$$

Hence, the uncontrolled cargo ship has an equivalent time constant

$$T_{\text{cargo ship}} = 118.0 + 7.8 - 18.5 = 107.3 \text{ s} > 0$$

while the oil tanker has an equivalent time constant

$$T_{\text{oil tanker}} = -124.1 + 16.4 - 46.0 = -153.7 \text{ s} < 0$$

This implies that the cargo ship is straight-line stable while the oil tanker is unstable.

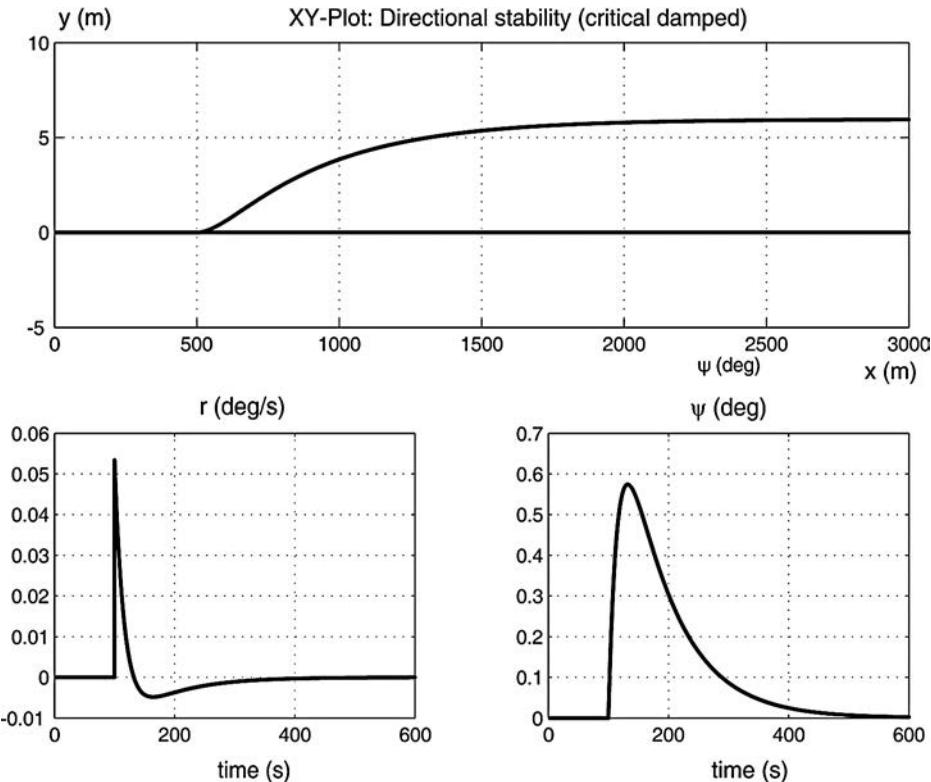


Figure 12.3 Directional stability for a critically damped ship ($\zeta = 1.0$) when an impulse $w(t)$ is injected at $t = 100$ s.

Criteria for Straight-Line Stability

Recall that a ship is said to be dynamically straight-line stable if it returns to a straight-line motion after a disturbance in yaw without any corrective action from the rudder. Consequently, instability refers to the case when the ship goes into a starboard or port turn without any rudder deflections. For Nomoto's first-order model straight-line motion was guaranteed for a positive time constant T . Similarly, it is possible to derive a criterion for straight-line stability for the state-space model (7.33):

$$\mathbf{M}\dot{\mathbf{v}} + \mathbf{N}(u_0)\mathbf{v} = \mathbf{b}\delta \quad (12.14)$$

where $\mathbf{v} = [v, r]^\top$. Applications of *Laplace's transformation* to the linear model (12.14) with $\mathbf{v}(0) = \mathbf{0}$ yields

$$[\mathbf{M}s + \mathbf{N}(u_0)]\mathbf{v}(s) = \mathbf{b}\delta(s) \quad (12.15)$$

Consequently,

$$\mathbf{v}(s) = [\mathbf{M}s + \mathbf{N}(u_0)]^{-1}\mathbf{b}\delta(s) = \frac{\text{adj}(\mathbf{M}s + \mathbf{N}(u_0))}{\det(\mathbf{M}s + \mathbf{N}(u_0))}\mathbf{b}\delta(s) \quad (12.16)$$

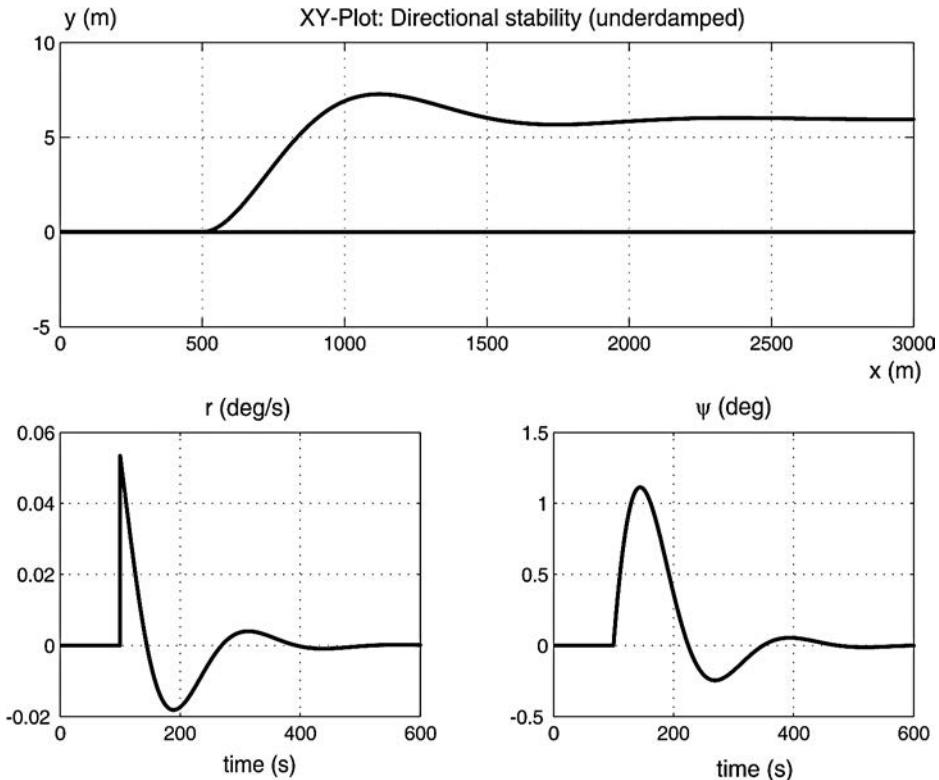


Figure 12.4 Directional stability for an underdamped ship ($\zeta = 0.1$) when an impulse $w(t)$ is injected at $t = 100$ s.

The characteristic equation is

$$\det(\mathbf{M}\sigma + \mathbf{N}(u_0)) = A\sigma^2 + B\sigma + C = 0 \quad (12.17)$$

where

$$\begin{aligned} A &= \det(\mathbf{M}) \\ B &= n_{11}m_{22} + n_{22}m_{11} - n_{12}m_{21} - n_{21}m_{12} \\ C &= \det(\mathbf{N}(u_0)) \end{aligned} \quad (12.18)$$

The two roots $\sigma_{1,2}$ of (12.17), both of which must have negative real parts for open-loop stability, are

$$\operatorname{Re}\{\sigma_{1,2}\} = \operatorname{Re} \left\{ \frac{-B \pm \sqrt{B^2 - 4AC}}{2A} \right\} < 0 \quad (12.19)$$

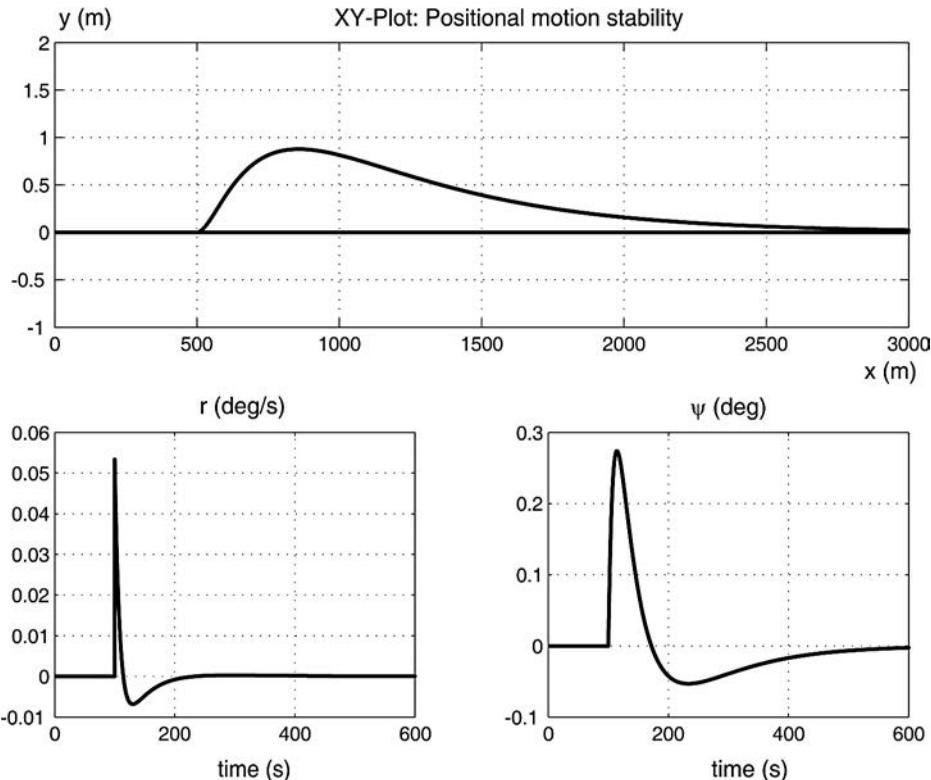


Figure 12.5 Positional motion stability for a PID-controlled ship when an impulse $w(t)$ is injected at $t = 100$ s.

The quantities $\sigma_{1,2}$ are often referred to as the control-fixed stability indices for straight-line stability. Alternatively, the Routh stability criterion can be applied.

Theorem 12.1 (The Routh Stability Criterion)

Consider the characteristic equation

$$a_n \lambda^n + a_{n-1} \lambda^{n-1} + a_{n-2} \lambda^{n-2} + \dots + a_0 = 0 \quad (12.20)$$

To apply the Routh criterion, the Routh array shown in Table 12.1 must be constructed. The coefficients a_i are the coefficients of the characteristic equation (12.20) and b_i, c_i, d_i, \dots are defined as

$$\begin{aligned} b_1 &= (a_{n-1}a_{n-2} - a_n a_{n-3})/a_{n-1} & b_2 &= (a_{n-1}a_{n-4} - a_n a_{n-5})/a_{n-1} & \dots \\ c_1 &= (b_1 a_{n-3} - a_{n-1} b_2)/b_1 & c_2 &= (b_1 a_{n-5} - a_{n-1} b_3)/b_1 & \dots \\ d_1 &= (c_1 b_2 - c_2 b_1)/c_1 & & & \dots \end{aligned}$$

Necessary and sufficient conditions for the system to be stable are:

1. All the coefficients of the characteristic equation must be nonzero and have the same sign.
2. All the coefficients of the first column of the Routh array must have the same sign.

Table 12.1 Routh array

λ^n	a_n	a_{n-2}	a_{n-4}	...
λ^{n-1}	a_{n-1}	a_{n-3}	a_{n-5}	...
λ^{n-2}	b_1	b_2	b_3	...
λ^{n-3}	c_1	c_2	c_3	...
λ^{n-4}	d_1	d_2	d_3	...
\vdots	...			

If Condition 2 is violated, the number of sign changes will indicate how many roots of the characteristic equation will have positive real parts. Hence, the system will be unstable.

Proof. See Routh (1877).

According to the Routh stability criterion, necessary and sufficient conditions for a ship given by (12.14) with characteristic equation (12.17) to be stable are

$$A, B, C > 0 \quad (12.21)$$

The first condition $A = \det(\mathbf{M}) > 0$ is automatically satisfied since the inertia matrix \mathbf{M} is always positive definite for a marine craft. Condition $B > 0$ implies that

$$n_{11}m_{22} + n_{22}m_{11} > n_{12}m_{21} + n_{21}m_{12} \quad (12.22)$$

Consequently, the products of the diagonal elements of \mathbf{M} and $\mathbf{N}(u_0)$ must be larger than the products of the off-diagonal elements. This is satisfied for most ships. Consequently, condition (12.21) reduces to

$$C = \det(\mathbf{N}(u_0)) > 0 \quad (12.23)$$

This condition has also been verified by Abkowitz (1964), who stated the following theorem.

Theorem 12.2 (Straight-Line Stability (Abkowitz, 1964))

A ship is dynamically stable in straight-line motion if the hydrodynamic derivatives satisfy

$$\begin{aligned} \det(\mathbf{N}(u_0)) &= \det \begin{bmatrix} -Y_v & mu_0 - Y_r \\ -N_v & mx_g u_0 - N_r \end{bmatrix} \\ &= Y_v(N_r - mx_g u_0) - N_v(Y_r - mu_0) > 0 \end{aligned} \quad (12.24)$$

Proof. This is seen as a consequence of (12.23) and (12.24).

It is interesting to notice that making C more positive will improve stability and thus reduce the ship's maneuverability, and the other way around. Straight-line stability implies that the new path of the ship will be a straight line after a disturbance in yaw. The direction of the new path will usually differ from the initial path. Contrary to this, unstable ships will go into a starboard or port turn without any rudder

deflection. It should be noted that most modern large tankers are slightly unstable. For such ships, the criterion (12.24) corresponds to one of the poles being in the right half-plane.

Straight-Line Stability in Terms of Time Constants

The criterion (12.21) can be related to Nomoto's second-order model (7.46) by noticing that

$$T_1 T_2 = \frac{A}{C} > 0, \quad T_1 + T_2 = \frac{B}{C} > 0 \quad (12.25)$$

Consequently, straight-line stability is guaranteed if $T_1 > 0$ and $T_2 > 0$. This can also be seen from

$$\sigma_{1,2} = -\frac{1}{T_{1,2}} = \operatorname{Re} \left\{ \frac{-B \pm \sqrt{B^2 - 4AC}}{2A} \right\} < 0 \quad (12.26)$$

Criteria for Directional Stability

Dynamic stability on course, or directional stability, cannot be obtained without activating the rudder. Usually a PID control system is used to generate the necessary rudder action to stabilize the ship. For simplicity, consider a PD controller:

$$\delta = -K_p(\psi - \psi_d) - K_d r \quad (12.27)$$

which after substitution into Nomoto's second-order model yields the closed-loop dynamics:

$$T_1 T_2 \psi^{(3)} + (T_1 + T_2 + T_3 K K_d) \ddot{\psi} + (1 + K K_d + T_3 K K_p) \dot{\psi} + K K_p \psi = K K_p \psi_d \quad (12.28)$$

From this expression, the cubic characteristic equation

$$A\sigma^3 + B\sigma^2 + C\sigma + D = 0 \quad (12.29)$$

is recognized, where

$$A = T_1 T_2 \quad (12.30)$$

$$B = T_1 + T_2 + T_3 K K_d \quad (12.31)$$

$$C = 1 + K K_d + T_3 K K_p \quad (12.32)$$

$$D = K K_p \quad (12.33)$$

The requirement for directional stability is

$$\operatorname{Re}\{\sigma_{1,2,3}\} < 0 \quad (12.34)$$

This can be checked by forming the Routh array:

$$\begin{array}{cc} A & C \\ B & D \\ \frac{BC-AD}{B} & 0 \\ D & \end{array} \quad (12.35)$$

Consequently, sufficient and necessary conditions for the ship to be dynamically stable on course are

$$A, B, C, D > 0 \quad (12.36)$$

$$BC - AD > 0 \quad (12.37)$$

This again implies that the controller gains K_p and K_d must be chosen such that the conditions (12.36) and (12.37) are satisfied.

12.1.2 Maneuverability

Several ship maneuvers can be used to evaluate the robustness, performance and limitations of a ship. This is usually done by defining a criterion in terms of a *maneuvering index* or by simply using a *maneuvering characteristic* to compare the maneuverability of the test ship with previously obtained results from other ships. A frequently used measure of maneuverability is the turning index of Norrbom (1965).

The Norrbom Measure of Maneuverability

Norrbom (1965) defines the *turning index* as

$$P := \frac{\psi'(t' = 1)}{\delta'(t' = 1)} \quad (12.38)$$

where $t' = t(U/L)$ is the nondimensional time. P is a measure of turning ability or maneuverability since it can be interpreted as the heading change per unit rudder angle in one ship length traveled at $U = 1$ m/s. An expression for P can be found by solving the ODE:

$$T' \ddot{\psi}' + \dot{\psi}' = K' \delta' \quad (12.39)$$

with $\delta' = \text{constant}$. This results in

$$\psi'(t') = K'[t' - T' + T' \exp(-(t'/T'))] \delta'(t') \quad (12.40)$$

A second-order Taylor expansion of $\exp(-t'/T')$ is

$$\exp(-t'/T') = 1 - \frac{t'}{T'} + \frac{(t')^2}{2(T')^2} + O(3) \quad (12.41)$$

such that

$$\frac{\psi'(t')}{\delta'(t')} \approx K' \left[t' - T' + T' \left(1 - \frac{t'}{T'} + \frac{(t')^2}{2(T')^2} \right) \right] = K \frac{(t')^2}{2T'} \quad (12.42)$$

$$\frac{\psi'(t' = 1)}{\delta'(t' = 1)} \approx K' \left[\frac{(t')^2}{2T'} \right]_{t'=1} = \frac{K'}{2T'} \quad (12.43)$$

Consequently,

$$P \approx \frac{1}{2} \frac{K'}{T'} \quad (12.44)$$

The P number is a good measure of maneuverability for course-stable ships. Norrbin concludes that $P > 0.3$ guarantees a reasonable standard of course-change quality for most ships while $P > 0.2$ seems to be sufficient for large oil tankers. For poorly stable ships it is recommended to use P together with another maneuverability index, for instance the slope $dr'/d\delta'$ or the width of the $r'-\delta'$ loop (see Figure 12.12 later).

Maneuvering Characteristics

A maneuvering characteristic can be obtained by changing or keeping a predefined course and speed of the ship in a systematic manner by means of active controls. For most surface vessels these controls are rudders, fins, propellers and thrusters. However, since ship maneuverability depends on the water depth, environmental forces, ship speed and hydrodynamic derivatives care must be taken when performing a full-scale maneuvering test. A guide for sea trials describing how these maneuvers should be performed is found in SNAME (1989). The following standard ship maneuvers have been proposed by the International Towing Tank Conference (ITTC):

- **Turning Circle:** This trial is mainly used to calculate the ship's steady turning radius and to check how well the steering machine performs under course-changing maneuvers.
- **Kempf's Zigzag Maneuver:** The zigzag test is a standard maneuver used to compare the maneuvering properties and control characteristic of a ship with those of other ships. Another feature is that the experimental results of the test can be used to calculate the K and T values of Nomoto's first-order model.
- **Pull-Out Maneuver:** The pull-out maneuver can be used to check whether the ship is straight-line stable or not. The maneuver can also be used to indicate the degree of stability.
- **Dieudonné's Spiral Maneuver:** The spiral maneuver is also used to check straight-line stability. The maneuver gives an indication of the range of validity of the linear theory.
- **Bech's Reverse Spiral Maneuver:** The reverse spiral maneuver can be used for unstable ships to produce a nonlinear maneuvering characteristic. The results from the test indicate which rudder corrections are required to stabilize an unstable ship.
- **Stopping Trials:** Crash stops and low-speed stopping trials can be used to determine the ship's head reach and maneuverability during emergency situations.

Turning Circle

This is probably the oldest maneuvering test. The test can be used as an indication on how well the steering machine and rudder control performs during course-changing maneuvers. It is also used to calculate standard measures of maneuverability such as *tactical diameter*, *advance* and *transfer* shown in Figure 12.6; see Gertler and Hagen (1960) for a detailed description.

Matlab

The turning circle for the Mariner class vessel is computed using the MSS toolbox script ExTurnCircle.m, where:

```
t_final = 700; % final simulation time (sec)
t_rudderexecute = 100; % time rudder is executed (sec)
h = 0.1; % sampling time (sec)

% Mariner class cargo ship, cruise speed U0 = 7.7 m/s
x = zeros(7,1); % x=[u v r x y psi delta]' (initial values)
u_i = -15*pi/180; % delta_c=-delta_R at time t = t_rudderexecute

[t,u,v,r,x,y,psi,U] =...
turncircle('mariner', x, ui, t_final, t_rudderexecute, h);
```

The results are plotted in Figure 12.6. Similar results are obtained by replacing mariner.m with the container ship, container.m; see ExTurnCircle.m.

The maneuvering characteristics for the Mariner class vessel were computed to be:

Rudder execute (x coordinate):	769 m
Steady turning radius:	711 m
Maximum transfer:	1 315 m
Maximum advance:	947 m
Transfer at 90 degrees heading:	534 m
Advance at 90 degrees heading:	943 m
Tactical diameter at 180 degrees heading:	1 311 m

The *steady turning radius R* is perhaps the most interesting quantity obtained from the turning trials. In the maneuvering trial code of the 14th ITTC (1975) it is proposed to turn the ship over at maximum speed and with a rudder angle of minimum 15° to obtain the turning circle. The rudder angle δ should be held constant such that a constant rate of turn is reached (in practice a turning circle of 540° may be necessary).

The output from a positioning system is used to calculate the tactical diameter, steady turning radius, maximum advance and maximum transfer. A typical turning circle corresponding to a negative rudder angle is shown in Figure 12.6.

For a constant rudder angle δ , the ship will move in a circle with constant turning radius R and speed U in the steady state, that is $\dot{\mathbf{v}} = \mathbf{0}$. Solving (7.33) for the steady-state solution of $\mathbf{v} = [v, r]^T$ yields

$$\mathbf{N}(u_0)\mathbf{v} = \mathbf{b}\delta \implies \mathbf{v} = \mathbf{N}^{-1}(u_0)\mathbf{b}\delta \quad (12.45)$$

The equation for r in this expression becomes

$$r = \frac{(Y_v N_\delta - N_v Y_\delta)}{Y_v(N_r - mx_g u_0) - N_v(Y_r - mu_0)} \delta \quad (12.46)$$

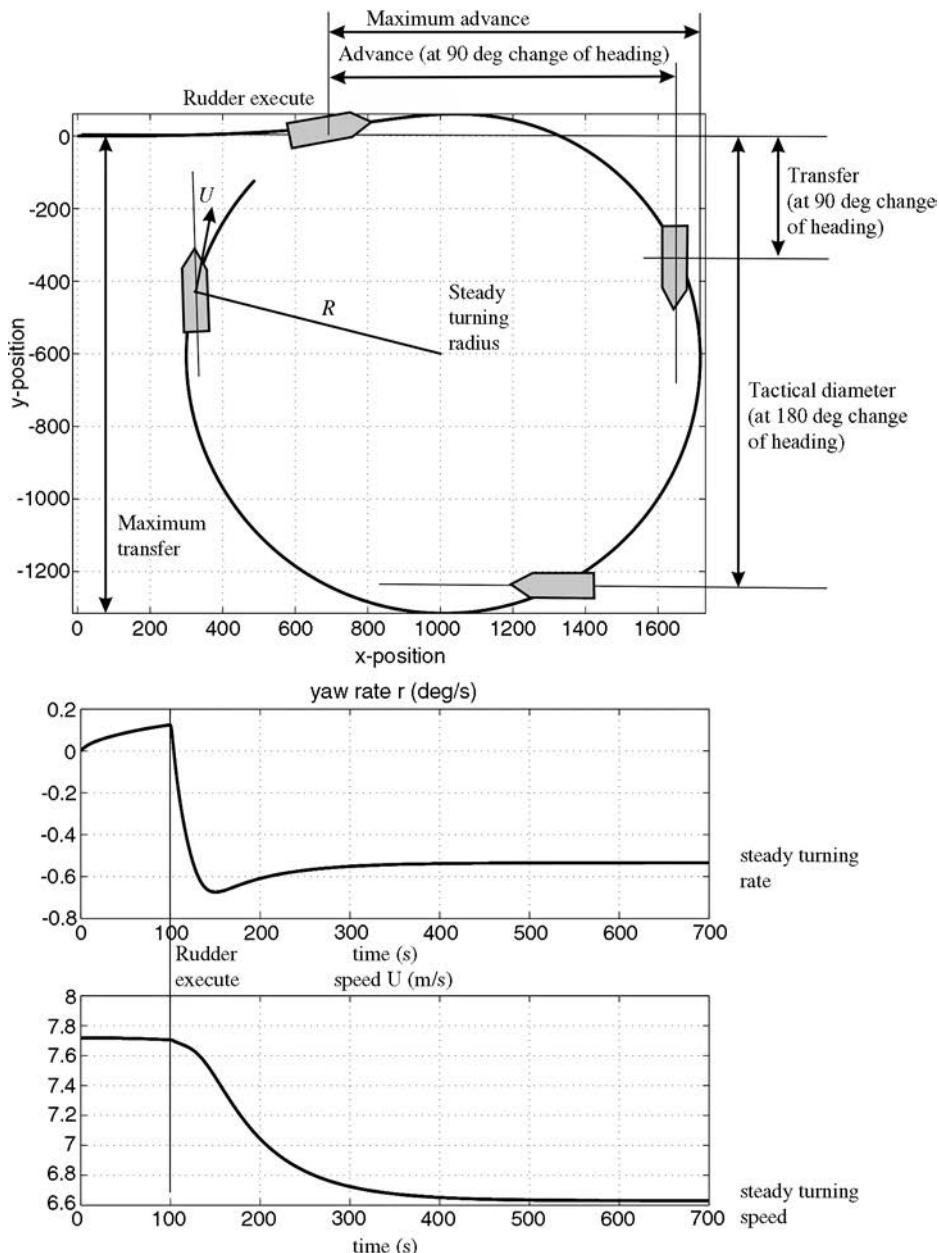


Figure 12.6 Turning circle, yaw rate and speed for the Mariner class vessel for a constant rudder angle $\delta_R = -15$ degrees applied at $t = 100$ s.

The ship's turning radius R is defined as

$$R := \frac{U}{r} \quad \text{where} \quad U = \sqrt{u^2 + v^2} \quad (12.47)$$

Introducing the length $L = L_{pp}$ of the ship, the following expression for the ratio (R/L) is obtained:

$$\left(\frac{R}{L}\right) = \left(\frac{U}{L}\right) \frac{C}{(Y_v N_\delta - N_v Y_\delta)} \frac{1}{\delta}, \quad \delta \neq 0 \quad (12.48)$$

where

$$C = \det(N(u_0)) = Y_v(N_r - mx_g u_0) - N_v(Y_r - mu_0) > 0 \quad (\text{stable ship})$$

is recognized as one of the stability derivatives in the straight-line stability criterion discussed in Section 12.1.1. From (12.48) it is seen that increased stability (large C) implies that the turning radius will increase. Consequently, a highly stable ship requires more maneuvering effort than a marginally stable one. The ratio (R/L) can also be written in terms of nondimensional quantities by

$$\left(\frac{R}{L}\right) = \frac{Y'_v(N'_r - m' x'_g) - N'_v(Y'_r - m')}{Y'_v N'_\delta - N'_v Y'_\delta} \frac{1}{\delta}, \quad \delta \neq 0 \quad (12.49)$$

This formula is independent of the ship speed. It should be noted that the formulae for the turning radius are based on linear theory, which assumes that δ is small and accordingly that R is large.

Another feature of the turning test is that the Nomoto gain and time constants can be determined. This is illustrated in the following example where a cargo ship is considered.

Example 12.2 (Determination of the Nomoto Gain and Time Constants)

The Nomoto gain and time constants can be computed from a turning test by using nonlinear least-squares curve fitting, for instance. Solving the ODE:

$$Tr + r = K\delta \quad (12.50)$$

for a step input $\delta = \delta_0 = \text{constant}$ yields

$$r(t) = \exp(-t/T)r(0) + [1 - \exp(-t/T)] K\delta_0 \quad (12.51)$$

where K and T are unknowns. The Matlab MSS toolbox script ExKT.m fits this model to a simulated step response of the model mariner.m, which is a nonlinear model of the Mariner class vessel.

The results for a step $\delta_0 = 5^\circ$ and $U = 7.7 \text{ m/s} = 15 \text{ knots}$, are (see Figure 12.7)

$$K = 0.09 \text{ s}^{-1} \quad (12.52)$$

$$T = 22.6 \text{ s} \quad (12.53)$$

The Norrbom measure of maneuverability becomes

$$P = \frac{1}{2} \frac{K'}{T'} = \frac{1}{2} \frac{K}{T} \left(\frac{L}{U}\right)^2 = \frac{1}{2} \left(\frac{0.09}{22.6}\right) \left(\frac{160.9}{7.7}\right)^2 = 0.87 \quad (12.54)$$

which guarantees good maneuverability since $P > 0.3$. The turning circle is shown in Figure 12.6, indicating that the steady-state turning radius is 711 m.

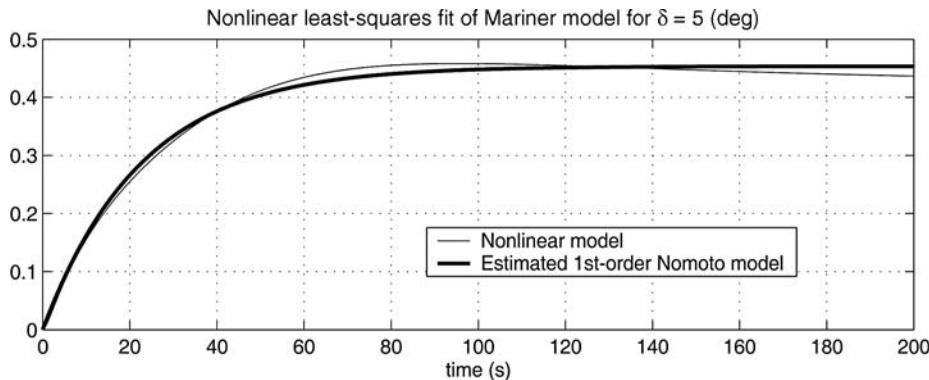


Figure 12.7 Plot showing the estimated linear model and the nonlinear Mariner model for a step $\delta = \delta_0 = 5$ degrees.

Matlab

```
% ExKT Script for computation of Nomoto gain and time constants
% using nonlinear least squares. The rudder input is 5 deg at t=0

N = 2000;           % number of samples
h = 0.1;            % sample time

xout = zeros(N,2);
x = zeros(7,1);
delta_R = 5*(pi/180);           % rudder angle step input

for i=1:N,
    xout(i,:) = [(i-1)*h ,x(3)];
    xdot = mariner(x,delta_R);      % nonlinear Mariner model
    x = euler2(xdot,x,h);         % Euler integration
end

% time-series
tdata = xout(:,1);
rdata = xout(:,2)*180/pi;

% nonlinear least-squares parametrization: x(1)=1/T and x(2)=K
x0 = [0.01 0.1]';
F = inline('exp(-tdata*x(1))*0 +...
            x(2)*(1-exp(-tdata*x(1)))*5','x','tdata')
x = lsqcurvefit(F,x0, tdata, rdata);

plot(tdata,rdata,'g',tdata,exp(-tdata*x(1))*0 +...
      x(2)*(1-exp(-tdata*x(1)))*5,'r'),grid
```

```

title('NLS fit of Mariner model for \delta = 5 (deg)')
xlabel('time (s)')
legend('Nonlinear model','Estimated 1st-order Nomoto model')

```

Kempf's Zigzag Maneuver

The zigzag test was first proposed by Kempf (1932). Comprehensive test results of 75 freighters are published in Kempf (1944). The zigzag time response (see Figures 12.8–12.9) is obtained by moving the rudder 20° to starboard from an initially straight course. The rudder setting is kept constant until the heading is changed 20° , and then the rudder is reversed 20° to port. Again, this rudder setting is maintained until the ship's heading has reached 20° in the opposite direction. This process continues until a total of five rudder step responses have been completed. This test is usually referred to as a 20° – 20° maneuver; the first angle refers to the actual rudder settings while the second angle denotes how much the heading angle should change before the rudder is reversed.

The zigzag maneuver was standardized by the International Towing Tank Conference (ITTC) in 1963. For larger ships, ITTC has recommended the use of a 10° – 10° or a 20° – 10° maneuver to reduce the time and waterspace required.

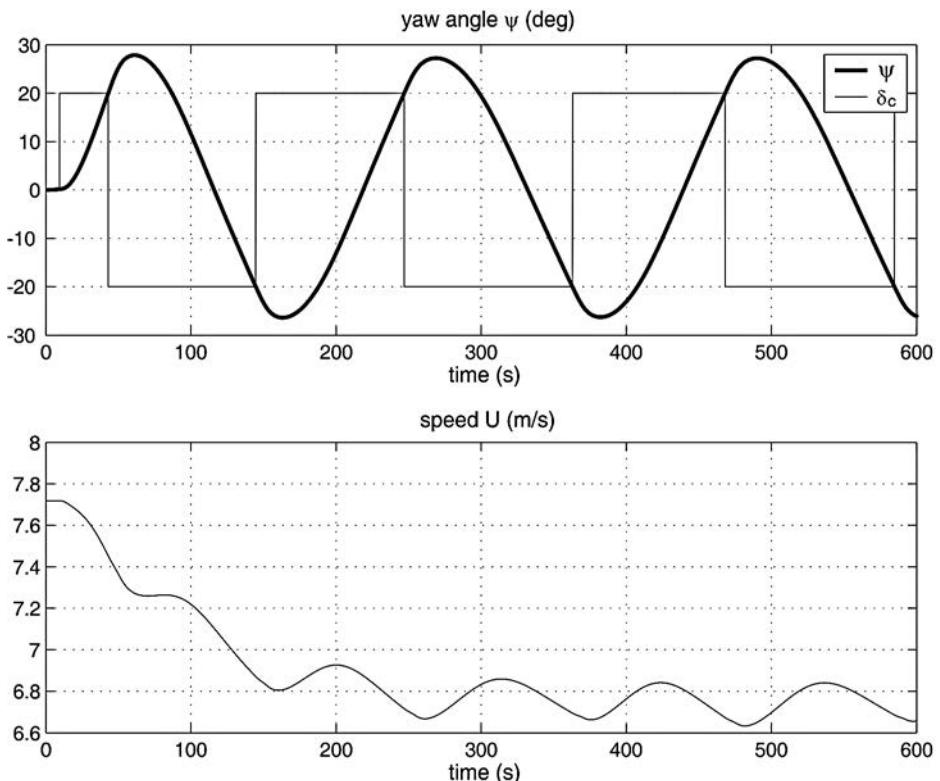


Figure 12.8 A 20° – 20° maneuver for the Mariner class vessel.

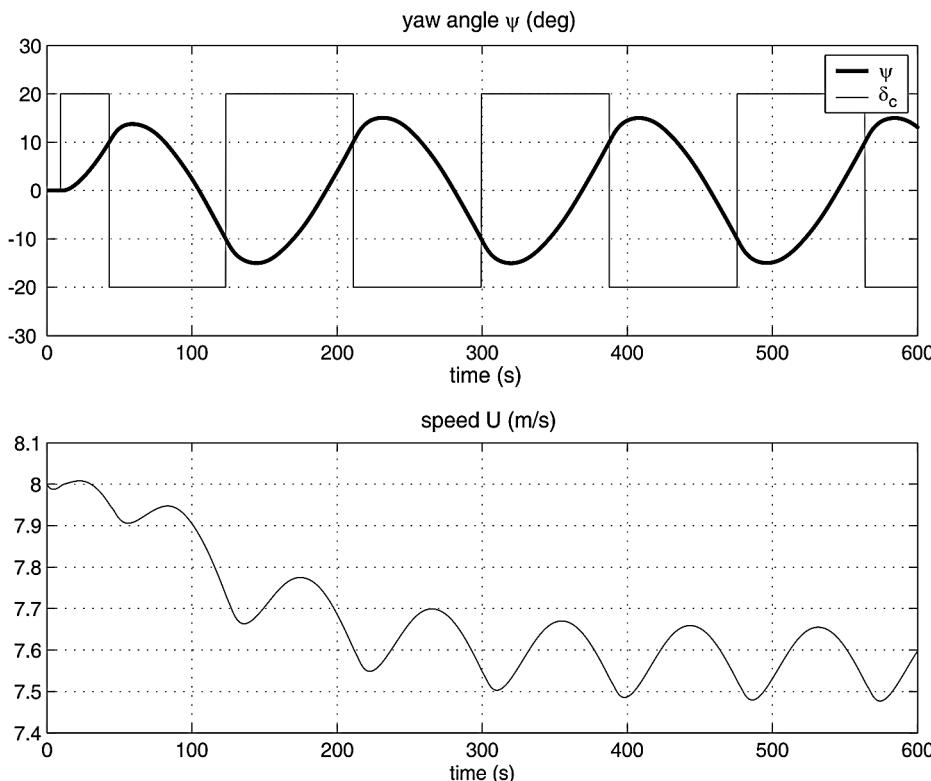


Figure 12.9 A 20° – 10° maneuver for the container ship.

The only apparatus required to perform the test is a compass and a stopwatch. Alternatively, a computer interfaced for real-time logging of compass data can be used. The results from the zigzag maneuver can be used to compare the maneuvering properties of different ships. Maneuvering trials are also used in the design process since it is possible to test scale models in towing tanks to see how well they perform. In addition, maneuvering characteristics can be computed using hull parameters and by performing computer simulations based on seakeeping and maneuvering models.

Example 12.3 (Zigzag Maneuvering Trials)

Both the Mariner class vessel (`mariner.m`) and the container ship (`container.m`) are simulated for a 20° – 20° and a 20° – 10° zigzag maneuver, respectively, by using the Matlab script `ExZigZag.m`.

The simulation results for the two vessels are shown in Figures 12.8–12.9.

Matlab

```
t_final = 600; % final simulation time (sec)
t_rudderexecute = 10; % time rudder is executed (sec)
h = 0.1; % sampling time (sec)
```

```
% 20-20 zigzag maneuver for the Mariner class cargo ship
% cruise speed U0 = 7.7 m/s (see mariner.m)
x = zeros(7,1); % x = [ u v r x y psi delta ]' (initial values)
ui = 0; % delta_c = 0 for time t < t_rudderexecute
[t,u,v,r,x,y,psi,U] =...
    zigzag('mariner',x,ui,t_final,t_rudderexecute,h,[20,20]);

% 20-10 zigzag maneuver for a container ship
% cruise speed 8.0 m/s see container.m
x = [8.0 0 0 0 0 0 0 0 70]'; % x = [ u v r x y psi delta n ]'
delta_c = 0; % delta_c = 0 for time t < t_rudderexecute
n_c = 80; % n_c = propeller revolution in rpm
ui = [delta_c, n_c];
[t,u,v,r,x,y,psi,U] =...
    zigzag('container',x,ui,t_final,t_rudderexecute,h,[20,10]);
```

Pull-Out Maneuver

In 1969 Roy Burcher proposed a simple test procedure to determine whether a ship is straight-line stable or not. This test is referred to as the pull-out maneuver (12th ITTC, 1969). The pull-out maneuver involves a pair of maneuvers in which a rudder angle of approximately 20° is applied and returned to zero after steady turning has been attained. Both a port and a starboard turn should be performed.

During the test the ship's rate of turn must be measured or at least calculated by numerical derivation of the measured compass heading. If the ship is straight-line stable the rate of turn will decay to the same value for both the starboard and port turns (see Figure 12.10). The ship is unstable if the steady rate of turn from the port and starboard turns differ (see Figure 12.11). The difference between these two steady rates of turn corresponds exactly to the height of Dieudonné's spiral loop.

Example 12.4 (Pullout Maneuver for a Stable and an Unstable Ship)

Both the Mariner class vessel (mariner.m) and the Esso Osaka tanker (tanker.m) are simulated under a pullout maneuver by using the Matlab script ExPullout.m.

Matlab:

```
delta_c = 20*pi/180; % rudder angle for maneuver (rad)
h = 0.1; % sampling time (sec)

% Mariner class cargo ship, speed U0 = 7.7 m/s (see mariner.m)
x = zeros(7,1); % x = [ u v r x y psi delta ]' (initial values)
```

```

ui = delta_c;      % ui = delta_c
[t,r1,r2] = pullout('mariner',x,ui,h);

% The Esso Osaka tanker (see tanker.m)
n = 80;
U = 8.23;
x = [ U 0 0 0 0 0 n ]';    % x = [ u v r x y psi delta n ]'
n_c = 80;                  % n_c = propeller revolution in rpm
depth = 200;                % water depth
ui = [delta_c, n_c, depth];
[t,r1,r2] = pullout('tanker',x,ui,h);

```

The results are shown in Figures 12.10–12.11 where the curves meet for the stable ship (Mariner class vessel) while there is an offset between the curves for the unstable model of the Esso Osaka tanker.

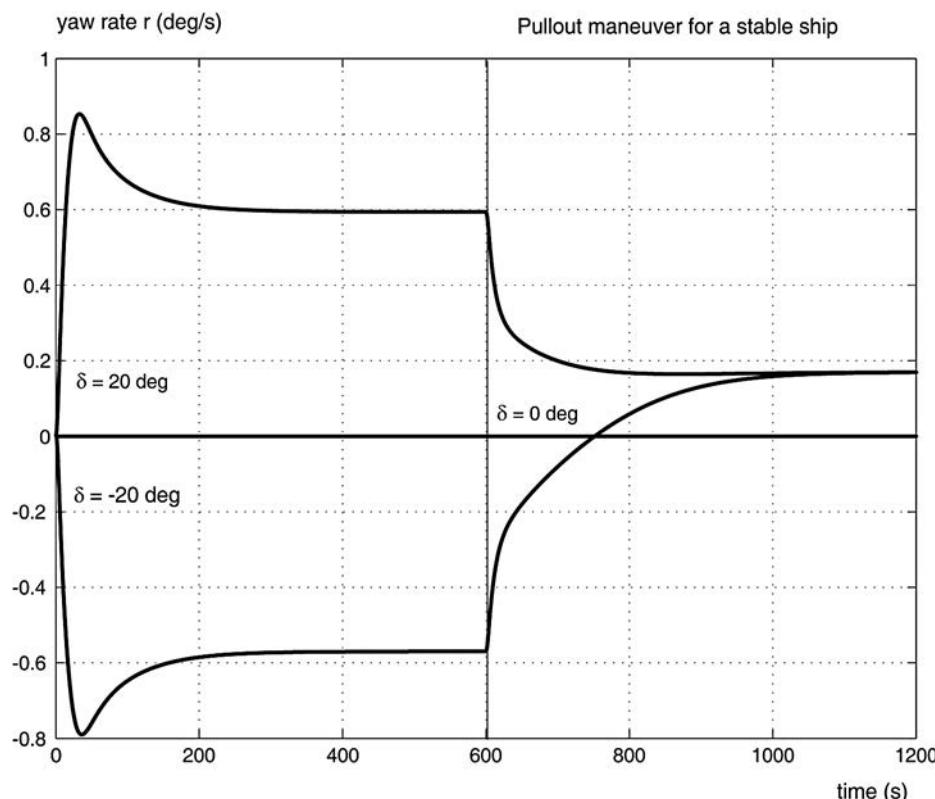


Figure 12.10 Pull-out maneuver for the Mariner class vessels. Notice that the positive and negative curves meet for the stable ship.

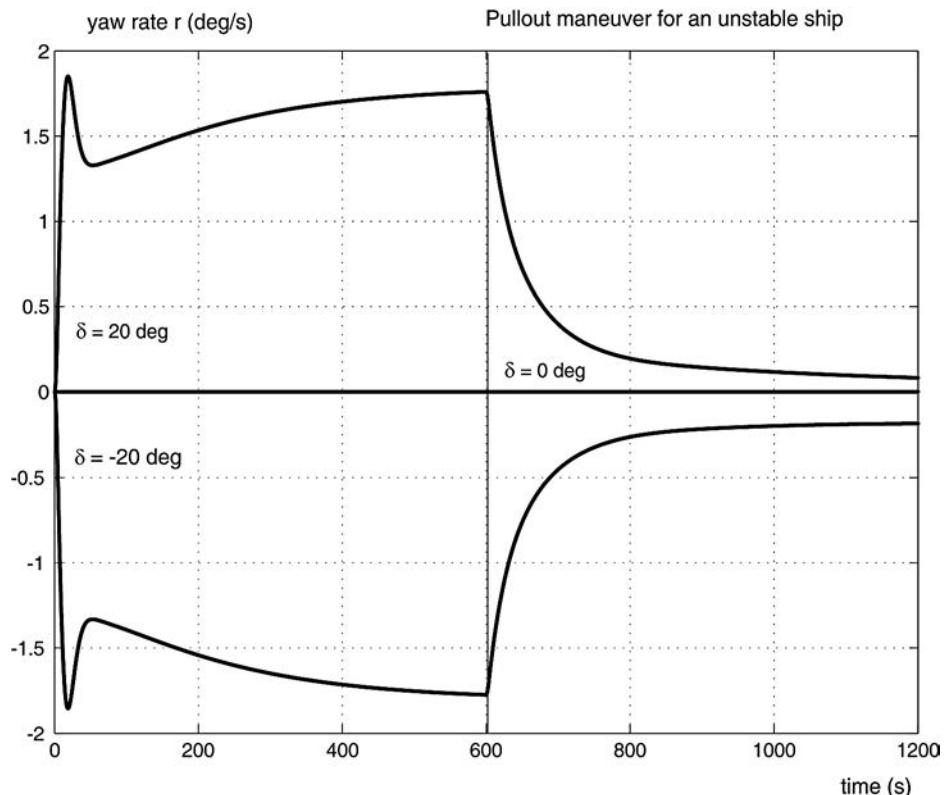


Figure 12.11 Pullout maneuver for the Esso Osaka tanker. Notice that the positive and negative curves do not meet.

Dieudonné's Spiral Maneuver

The direct spiral test was published first in 1949–1950 by the French scientist Jean Dieudonné. An English translation is found in Dieudonné (1953). The direct spiral maneuver is used to check straight-line stability. As seen from Figure 12.12, the maneuver also gives an indication of the degree of stability and the range of validity of the linear theory.

To perform the test the ship should initially be held on a straight course. The rudder angle is then put to 25° starboard and held until a steady yawing rate is obtained. After this the rudder angle is decreased in steps of 5° and again held until constant yawing rates are obtained for all the rudder angles. The procedure is performed for all rudder angles between 25° starboard and 25° port. In the range around zero rudder angle the step of 5° rudder should be reduced to obtain more precise values. The results are plotted in an r – δ diagram, as shown in Figure 12.12. It should be noted that the spiral maneuver should be performed in still air and calm water to obtain the best results.

For straight-line unstable ships it is recommended to use Bech's reverse spiral maneuver.

Bech's Reverse Spiral Maneuver

For stable ships both Dieudonné's direct and Bech's reverse spiral tests can be used. For unstable ships within the limits indicated by the pull-out maneuver Bech's reverse spiral should be applied. The reverse

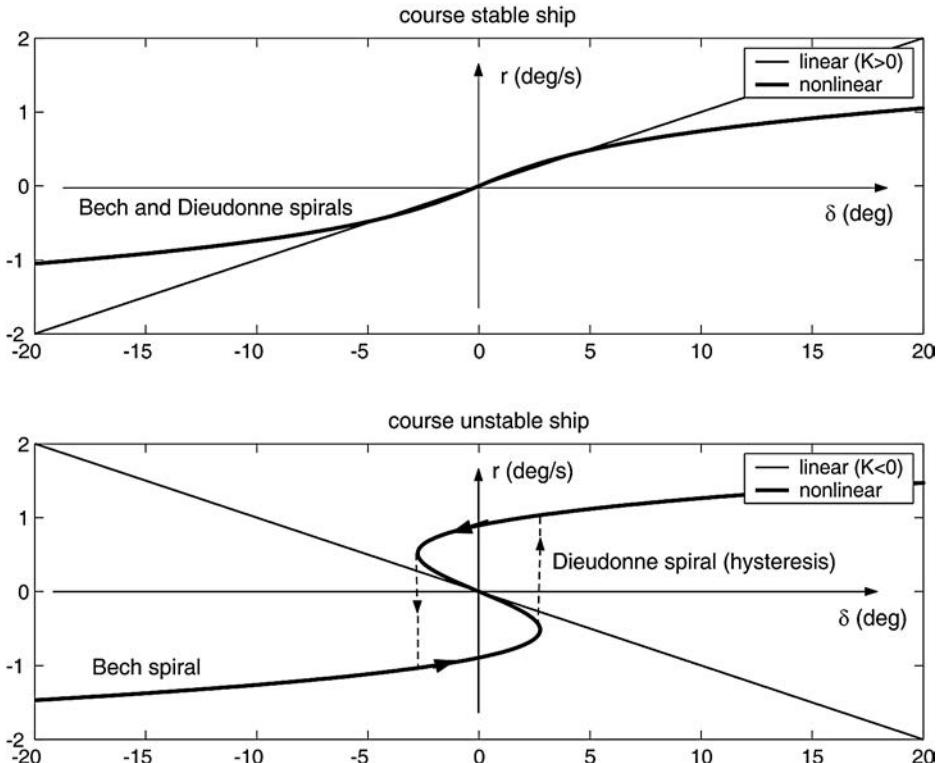


Figure 12.12 An r - δ diagram showing the Dieudonne and Bech spirals for both a stable and course-unstable ship. Notice the hysteresis loop in the Dieudonne spiral for the unstable ship.

spiral test was first published by Mogens Bech in 1966 and later in 1968 (Bech, 1968). Since then the reverse spiral test has been quite popular, because of the simplicity and reliability of the method. The reverse spiral test is also less time-consuming than Dieudonné's spiral test.

By observing that the ship steering characteristic is nonlinear outside a limited area, Bech (1968) suggested that one describes the *mean* value of the required rudder deflection δ_{ss} to steer the ship at a constant rate of turn r_{ss} as a nonlinear function:

$$\delta_{ss} = H_B(r_{ss}) \quad (12.55)$$

where $H_B(r_{ss})$ is a nonlinear function describing the maneuvering characteristic.

This can be understood by considering Nomoto's second-order model:

$$T_1 T_2 \ddot{r} + (T_1 + T_2) \dot{r} + K H_B(r) = K(\delta + T_3 \dot{\delta}) \quad (12.56)$$

where the linear term r has been replaced with a function $H_B(r)$. Assuming that $r = r_{ss}$ is constant in the steady state, that is $\ddot{r} = \dot{r} = \dot{\delta} = 0$, directly gives (12.55). This implies that the r - δ curve will be a single-valued (one-to-one) function of r for both the stable and unstable ship (see Figure 12.12). If the conventional spiral test is applied to an unstable ship a hysteresis loop will be observed.

The full-scale test is performed by measuring the necessary rudder action required to bring the ship into a desired rate of turn. For an unstable ship this implies that the rudder angle will oscillate about a mean rudder angle. The amplitude of the rudder oscillations should be kept to a minimum. After some time a *balance condition* is reached and both the mean rudder angle and rate of turn can be calculated. Care should be taken for large ships since they will require some more time to converge to their “balance condition”.

12.2 PID Control and Acceleration Feedback

This section discusses PID control design for SISO and MIMO motion control systems. The presented methods are used in many industrial systems. The PID control laws are also extended to include optional acceleration feedback. This topic is also covered by Lindegaard (2003), where experimental results with a model ship are used to document performance improvements due to acceleration feedback. Acceleration feedback can be implemented in conjunction with PID control without increasing the demand for control energy.

12.2.1 Linear Mass–Damper–Spring Systems

Consider the following two equivalent systems:

$$m\ddot{x} + d\dot{x} + kx = 0 \quad (12.57)$$

$$\ddot{x} + 2\zeta\omega_n\dot{x} + \omega_n^2x = 0 \quad (12.58)$$

The step response is shown in Figure 12.13. From (12.57) and (12.58) it follows that

$$2\zeta\omega_n = \frac{d}{m}, \quad \omega_n^2 = \frac{k}{m} \quad (12.59)$$

For second-order systems it is convenient to introduce

$$\begin{aligned} \omega_n &= \sqrt{\frac{k}{m}} && \text{natural frequency (undamped oscillator corresponding to } d = 0) \\ \zeta &= \frac{d}{2m\omega_n} && \text{relative damping ratio} \end{aligned}$$

Damped Oscillator

For the damped system $d > 0$, the frequency of the oscillation will be smaller than the natural frequency. This can be explained by considering the eigenvalues of the mass–damper–spring system (12.58):

$$\lambda_{1,2} = -\underbrace{\zeta\omega_n}_a \pm j\omega \quad (12.60)$$

From Figure 12.14 it is seen that

$$a^2 + \omega^2 = \omega_n^2, \quad \zeta = \frac{a}{\omega_n} = \cos(\phi) \quad (12.61)$$

Matlab

The step responses in Figure 12.13 is computed using (see ExMDS.m):

```

wn = 1; % natural frequency

subplot(211)
t = 0:0.01:20;
z = 0.5; sys = tf([wn*wn],[1 2*z*wn wn*wn]); step(sys,t)
hold on
z = 1.0; sys = tf([wn*wn],[1 2*z*wn wn*wn]); step(sys,t)
z = 2.0; sys = tf([wn*wn],[1 2*z*wn wn*wn]); step(sys,t)
hold off

subplot(212)
t = 0:0.01:50;
z = 0.1; sys = tf([wn*wn],[1 2*z*wn wn*wn]); step(sys,t)
hold on
sys = tf([wn*wn],[1 0 wn*wn]); step(sys,t)
hold off

```

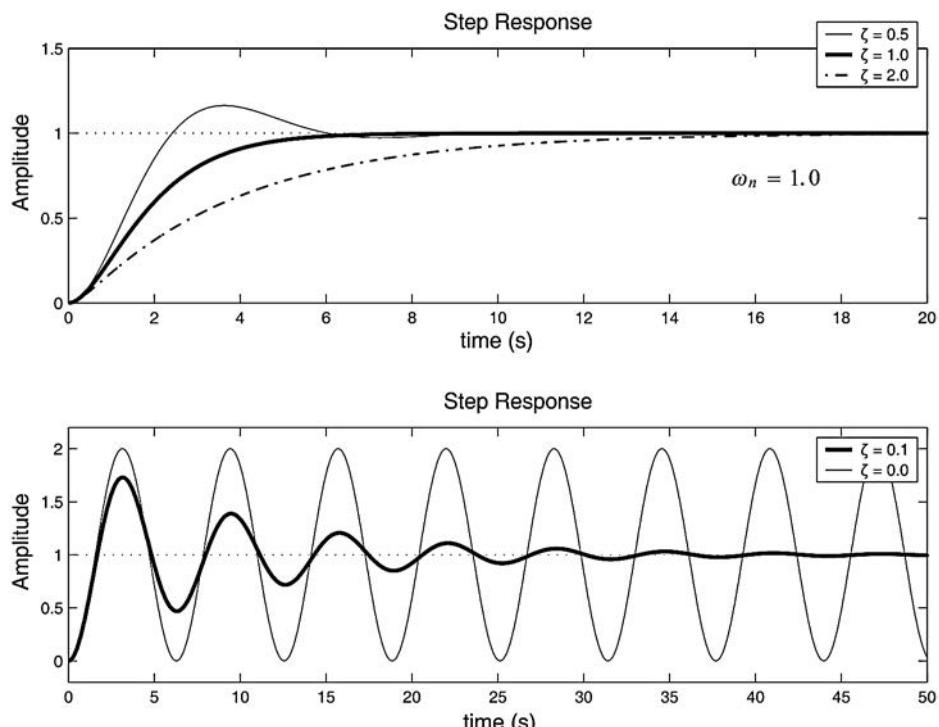


Figure 12.13 The upper plot shows a mass–damper–spring system for different relative damping ratios. The lower plot shows the undamped oscillator together with a damped oscillator. The plots are generated by ExMDS.m.

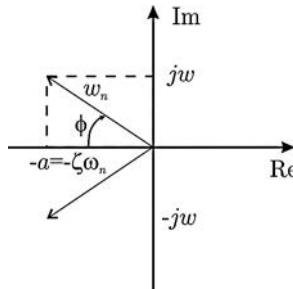


Figure 12.14 Graphical illustration of natural frequency ω_n , frequency of the damped system ω and absolute damping factor a .

and

$$a = \text{absolute damping factor}$$

$$\omega = \text{frequency of oscillation (damped system)}$$

The undamped oscillator is obtained by choosing $a = 0$. It is convenient to set

$$\omega = r\omega_n \quad (12.62)$$

where r is a reduction factor denoting the ratio between the natural frequency ω_n and the frequency ω of the linearly damped system. For marine craft a reduction of 0.5% in the natural frequency is common (Faltinsen, 1990). Hence,

$$r = 1 - \frac{0.5}{100} = 0.995 \quad (12.63)$$

From (12.61) and (12.62) it is seen that

$$a^2 + (r\omega_n)^2 = \omega_n^2 \quad (12.64)$$

⇓

$$a = \underbrace{\sqrt{1 - r^2}}_{\zeta} \omega_n \quad (12.65)$$

For $r = 0.995$ we obtain $\zeta = 0.1$, which is quite typical for a ship with bilge keels while the heave and pitch motions usually are more damped, for instance $\zeta = 0.2$. Next,

$$\begin{aligned} \frac{d}{m} &= 2\zeta\omega_n \\ &= 2\zeta\sqrt{\frac{k}{m}} \end{aligned} \quad (12.66)$$

which yields the following formula for linear damping:

$$d = 2\zeta\sqrt{km}, \quad \zeta = \sqrt{1 - r^2} \quad (12.67)$$

This formula is quite useful to determine the linear damping in *heave*, *roll* and *pitch* of an uncontrolled marine craft (open loop) since the mass m and spring (metacentric) coefficient k are easily obtained by other methods (see Chapters 3–5). The frequency of oscillation is

$$\omega = \sqrt{\frac{k}{m} - \left(\frac{d}{2m}\right)^2} \quad (12.68)$$

which for $d = 0$ reduces to the natural frequency of the undamped oscillator:

$$\omega \stackrel{d=0}{=} \sqrt{\frac{k}{m}} = \omega_n \quad (12.69)$$

Damping in *surge*, *sway* and *yaw*, however, cannot be determined by formula (12.67) since $k = 0$ in a pure *mass-damper* system. Linear damping for such a system:

$$m\ddot{x} + d\dot{x} = \tau \quad (12.70)$$

can be found by specifying the time constant $T > 0$. Let $T = m/d$ such that (12.70) becomes

$$T\ddot{x} + \dot{x} = \frac{1}{d}\tau \quad (12.71)$$

which yields the following design formula:

$$d = \frac{m}{T} \quad (12.72)$$

for the *mass-damper* system. Equations (12.67) and (12.72) will be referred to as the *linear damping formulae* for a *mass-damper-spring* and a *mass-damper* system, respectively. A relationship between the time constant T and the natural frequency ω_n in a PD-controlled system can be derived by considering (12.70) under feedback:

$$\tau = -K_d\dot{x} - K_p x \quad (12.73)$$

This gives the closed-loop system

$$m\ddot{x} + (d + K_d)\dot{x} + K_p x = 0 \quad (12.74)$$

and

$$2\zeta\omega_n = \frac{d + K_d}{m} \quad (12.75)$$

$$\omega_n = \sqrt{\frac{K_p}{m}} \quad (12.76)$$

In closed loop, K_p and K_d are positive constants and the natural period $T_n = 2\pi/\omega_n$ satisfies

$$\begin{aligned} 2\zeta\omega_n &= 2\zeta \frac{2\pi}{T_n} \\ &= \frac{d + K_d}{m} \\ &= \frac{1}{T} + \frac{K_d}{m} \end{aligned} \quad (12.77)$$

If $K_d/m \approx 1/T$, corresponding to increasing $1/T$ to $2/T$ by feedback control, the following useful relationship between the time constant and the natural period of a PD-controlled mass-damper system is obtained:

$$T \approx \frac{T_n}{2\pi\zeta} \quad (12.78)$$

Example 12.5 (Linear Damping in Roll and Pitch for Submarines)

Consider the linear pitch equation (7.265):

$$(I_y - M_{\dot{q}})\ddot{\theta} - M_q\dot{\theta} + \overline{BG}_z W \theta = \tau_5$$

Hence, the linear damping coefficient can be computed by using (12.67):

$$-M_q = 2\sqrt{1-r^2} \sqrt{\overline{BG}_z W(I_y - M_{\dot{q}})} > 0$$

where $M_{\dot{q}}$, W and \overline{BG}_z are assumed to be known and $r > 0$ is a design parameter. For roll a similar expression is obtained (see (7.273)):

$$-K_p = 2\sqrt{1-r^2} \sqrt{\overline{BG}_z W(I_x - K_p)} > 0$$

Example 12.6 (Linear Damping in Yaw for Ships and Underwater Vehicles)

Consider the Nomoto model (see Section 7.1.4):

$$(I_z - N_r)\ddot{r} - N_r r = N_\delta \delta \quad (12.79)$$

Assume that the moment of inertia $I_z - N_r$ is known. The linear damping coefficient N_r can be estimated by specifying the time constant. If it is assumed that the closed-loop yawing motion has a natural period $T_n = 150$ s and relative damping ratio $\zeta = 1.0$ (critically damped), it is possible to compute an estimate of the time constant in yaw using (12.78):

$$T \approx \frac{150 \text{ s}}{2\pi \times 1.0} = 23.8 \text{ s} \quad (12.80)$$

and from (12.72) the unknown hydrodynamic derivative becomes

$$-N_r = \frac{I_z - N_r}{T} \quad (12.81)$$

12.2.2 Acceleration Feedback

It is possible to extend the results of Section 12.2.1 to include acceleration feedback. Consider a mass-damper-spring system:

$$m\ddot{x} + d\dot{x} + kx = \tau + w \quad (12.82)$$

Let the control law be

$$\tau = \tau_{\text{PID}} - K_m\ddot{x} \quad (12.83)$$

where $K_m > 0$ is the acceleration feedback gain and τ_{PID} represents a conventional PID controller. This yields

$$(m + K_m)\ddot{x} + d\dot{x} + kx = \tau_{\text{PID}} + w \quad (12.84)$$

or equivalently

$$\ddot{x} + \frac{d}{m + K_m}\dot{x} + \frac{k}{m + K_m}x = \frac{1}{m + K_m}\tau_{\text{PID}} + \frac{1}{m + K_m}w \quad (12.85)$$

From this expression it is noticed that besides increasing the mass from m to $m + K_m$, acceleration feedback also reduces the gain in front of the disturbance w from $1/m$ to $1/(m + K_m)$. Hence, the system is expected to be less sensitive to an external disturbance w if acceleration feedback is applied.

This design can be further improved by introducing a frequency-dependent *virtual mass* in the following form (Sagatun *et al.*, 2001):

$$\tau = \tau_{\text{PID}} - h_m(s)\ddot{x} \quad (12.86)$$

If $h_m(s)$ is chosen as a low-pass filter:

$$h_m(s) = \frac{K_m}{T_m s + 1} \quad (12.87)$$

with gain $K_m > 0$ and time constant $T_m > 0$, it is seen that

$$\underbrace{\left(m + \frac{K_m}{T_m s + 1} \right)}_{m_{\text{total}}(s)} \ddot{x} + d\dot{x} + kx = \tau_{\text{PID}} + w \quad (12.88)$$

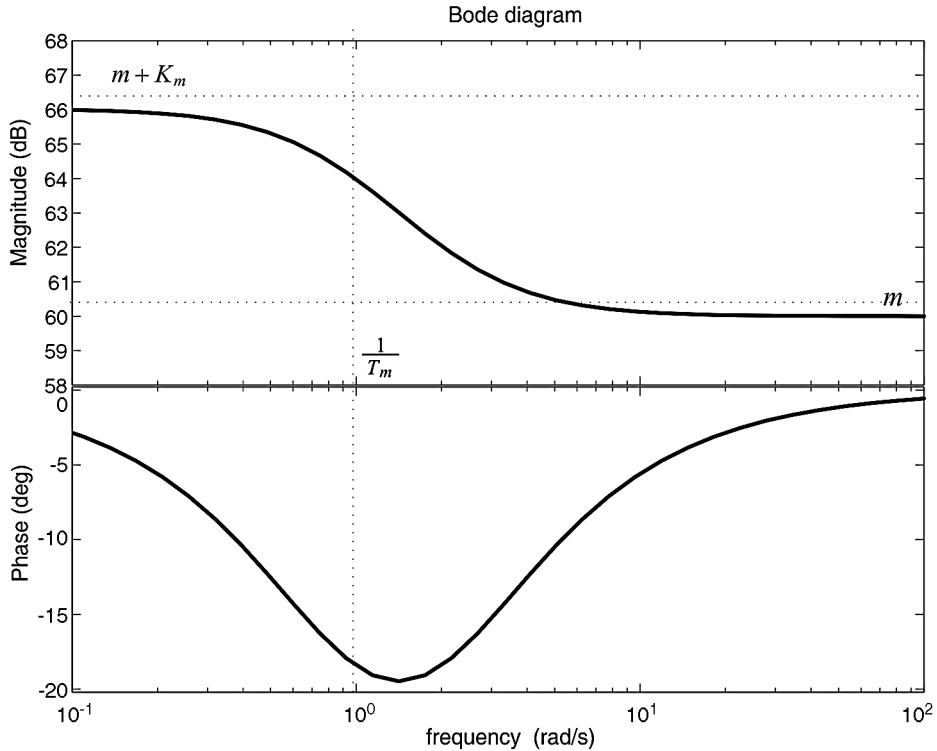


Figure 12.15 Total mass $m_{\text{total}}(s)$ as a function of frequency for $m = K_m = 1000$ (60 dB) and $T_m = 1.0$ s.

where the total mass of the system in closed loop is

$$m_{\text{total}}(s) = m + \frac{K_m}{T_m s + 1} = \frac{m T_m s + (m + K_m)}{T_m s + 1} \quad (12.89)$$

Hence, it can be concluded that the total mass is $m + K_m$ at low frequencies ($s \rightarrow 0$) while at high frequencies ($s \rightarrow \infty$) the total mass $m + K_m$ reduces to m . This is shown in Figure 12.15.

The filter $h_m(s)$ can be chosen rather arbitrarily depending on the application. For instance, a low-pass filter will remove high-frequency acceleration feedback components while a notch structure can be used to remove first-order wave-induced forces. This is seen by letting

$$g(s) = \frac{1}{m + h_m(s)} \quad (12.90)$$

such that (12.88) takes the form

$$\ddot{x} + g(s)d\dot{x} + g(s)kx = g(s)\tau_{\text{PID}} + g(s)w \quad (12.91)$$

where $g(s)$ is chosen such that the disturbance w is suppressed in a limited frequency band (low-pass, high-pass and notch). It will next be shown how a PID controller can be designed independently of the acceleration feedback loop.

12.2.3 PID Control with Acceleration Feedback

Consider the controller:

$$\tau = \underbrace{kx_d}_{\text{reference feedforward}} - \underbrace{\left(K_p \tilde{x} + K_d \dot{x} + K_i \int_0^t \tilde{x}(\tau) d\tau \right)}_{\text{PID controller}} - \underbrace{h_m(s) \ddot{x}}_{\text{acceleration feedback}} \quad (12.92)$$

with gains $K_p > 0$, $K_d > 0$ and $K_i > 0$ and tracking error $\tilde{x} = x - x_d$ (see Figure 12.16).

For simplicity, assume that $h_m(s) = K_m$ and $K_i = 0$. This gives

$$\tau = kx_d - (K_p \tilde{x} + K_d \dot{x}) - K_m \ddot{x} \quad (12.93)$$

The closed-loop system becomes

$$(m + K_m)\ddot{x} + (d + K_d)\dot{x} + (k + K_p)\tilde{x} = w \quad (12.94)$$

such that

$$\omega_n = \sqrt{\frac{k + K_p}{m + K_m}} \quad (12.95)$$

$$\zeta = \frac{d + K_d}{2(m + K_m)\omega_n} \quad (12.96)$$

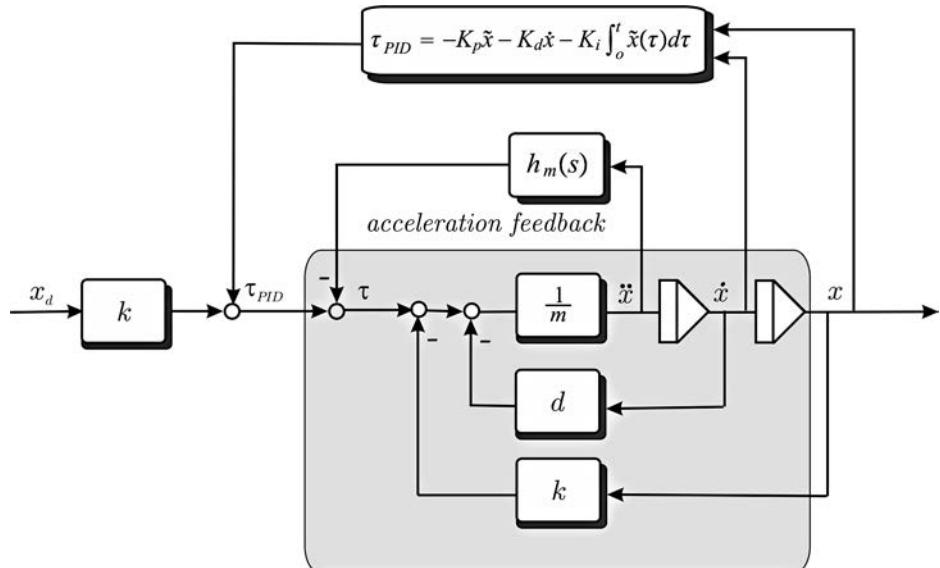


Figure 12.16 Acceleration feedback (inner loop) and PID feedback (outer loop).

Pole placement of the mass–damper–spring system suggests that K_p and K_d can be computed by specifying ω_n and ζ in (12.95) and (12.96). Solving for K_p and K_d , yields

$$K_p = (m + K_m)\omega_n^2 - k \quad (12.97)$$

$$K_d = 2\zeta\omega_n(m + K_m) - d \quad (12.98)$$

such that (12.94) becomes

$$\ddot{x} + 2\zeta\omega_n\dot{x} + \omega_n^2x = \omega_n^2x_d + \frac{1}{m + K_m}w \quad (12.99)$$

$$\Downarrow \{m + K_m \gg 1\}$$

$$\frac{x}{x_d}(s) \approx \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \quad (12.100)$$

This is a good approximation for $m + K_m \gg 1$. An even better approach is to add integral action $K_i > 0$ to compensate for a large constant disturbance w . Let the PID controller be written as

$$\tau = \underbrace{kx_d}_{\text{reference feedforward}} - \underbrace{K_p \left(1 + T_d s + \frac{1}{T_i s} \right) \tilde{x}}_{\text{PID}} - \underbrace{K_m \ddot{x}}_{\text{acceleration feedback}} \quad (12.101)$$

where $T_d = K_d/K_p$ and $T_i = K_p/K_i$ are the derivative and integral time constants, respectively. A *rule-of-thumb* is to choose

$$\frac{1}{T_i} \approx \frac{\omega_n}{10} \quad (12.102)$$

which states that the integrator is 10 times slower than the natural frequency ω_n . This yields

$$K_i = \frac{\omega_n}{10} K_p = \frac{\omega_n}{10} [(m + K_m)\omega_n^2 - k] \quad (12.103)$$

The natural frequency ω_n can be related to the system bandwidth ω_b by using the following definition:

Definition 12.1 (Control Bandwidth)

The control bandwidth of a system $y = h(s)u$ with negative unity feedback is defined as the frequency ω_b at which the loop transfer function $l(s) = h(s) \cdot 1$ is

$$|l(j\omega)|_{\omega=\omega_b} = \frac{\sqrt{2}}{2}$$

or equivalently

$$20 \log |l(j\omega)|_{\omega=\omega_b} = -3 \text{ dB}$$

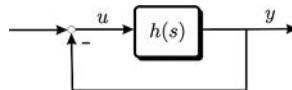


Figure 12.17 Closed-loop feedback system.

From this definition it can be shown that the control bandwidth of a second-order system:

$$h(s) = \frac{\omega_n^2}{s^2 + 2\xi\omega_n s + \omega_n^2} \quad (12.104)$$

with negative unity feedback is (see Figure 12.17)

$$\omega_b = \omega_n \sqrt{1 - 2\xi^2 + \sqrt{4\xi^4 - 4\xi^2 + 2}} \quad (12.105)$$

For a critically damped system, $\xi = 1.0$, this expression reduces to

$$\omega_b = \omega_n \sqrt{\sqrt{2} - 1} \approx 0.64 \omega_n \quad (12.106)$$

Table 12.2 summarizes the pole-placement algorithm.

Example 12.7 (Ship Autopilot Design)

Consider the Nomoto model (Nomoto et al., 1957):

$$T\ddot{\psi} + \dot{\psi} = K\delta \quad (12.107)$$

where ψ is the yaw angle and δ is the rudder angle (control input). From (12.82) it is seen that

$$m = \frac{T}{K}, \quad d = \frac{1}{K}, \quad k = 0 \quad (12.108)$$

Table 12.2 PID and acceleration feedback pole-placement algorithm

-
1. Specify the bandwidth $\omega_b > 0$ and the relative damping ratio $\xi > 0$
 2. Compute the natural frequency: $\omega_n = \frac{1}{\sqrt{1-2\xi^2+\sqrt{4\xi^4-4\xi^2+2}}} \omega_b$
 3. Specify the gain: $K_m \geq 0$ (optionally acceleration feedback)
 4. Compute the P gain: $K_p = (m + K_m)\omega_n^2 - k$
 5. Compute the D gain: $K_d = 2\xi\omega_n(m + K_m) - d$
 6. Compute the I gain: $K_i = \frac{\omega_n}{10} K_p$
-

The PID and acceleration feedback controller gains are found by using pole placement in terms of the design parameters K_m , ω_n and ζ , resulting in

$$\begin{aligned} K_m &\geq 0 \\ K_p &= \frac{T + KK_m}{K} \omega_n^2 > 0 \\ K_d &= \frac{T + KK_m}{K} 2\zeta\omega_n - \frac{1}{K} > 0 \\ K_i &= \frac{T + KK_m}{10K} \omega_n^3 > 0 \end{aligned}$$

For $K_m = 0$ (no angular acceleration feedback in yaw) this reduces to a conventional autopilot of PID type with gains:

$$\begin{aligned} K_p &= \frac{\omega_n^2 T}{K} > 0 \\ K_d &= \frac{2\zeta\omega_n T - 1}{K} > 0 \\ K_i &= \frac{\omega_n^3 T}{10K} > 0 \end{aligned}$$

12.2.4 MIMO Nonlinear PID Control with Acceleration Feedback

The PID control concept can be generalized to nonlinear mechanical systems by exploiting the kinematic equations of motion in the design. Consider the nonlinear model

$$\dot{\boldsymbol{\eta}} = \mathbf{J}_\Theta(\boldsymbol{\eta})\boldsymbol{v} \quad (12.109)$$

$$\mathbf{M}\ddot{\boldsymbol{v}} + \mathbf{C}(\boldsymbol{v})\dot{\boldsymbol{v}} + \mathbf{D}(\boldsymbol{v})\boldsymbol{v} + \mathbf{g}(\boldsymbol{\eta}) = \boldsymbol{\tau} + \boldsymbol{w} \quad (12.110)$$

where $\boldsymbol{\eta}$ and \boldsymbol{v} are assumed to be measured. Consider the control law

$$\boldsymbol{\tau} = \mathbf{g}(\boldsymbol{\eta}) - \mathbf{H}_m(s)\dot{\boldsymbol{v}} + \mathbf{J}_\Theta^\top(\boldsymbol{\eta})\boldsymbol{\tau}_{\text{PID}} \quad (12.111)$$

with acceleration feedback $\mathbf{H}_m(s)\dot{\boldsymbol{v}}$, gravity compensation $\mathbf{g}(\boldsymbol{\eta})$ and PID controller

$$\boldsymbol{\tau}_{\text{PID}} = -K_p \tilde{\boldsymbol{\eta}} - K_d \dot{\boldsymbol{\eta}} - K_i \int_0^t \tilde{\boldsymbol{\eta}}(\tau) d\tau \quad (12.112)$$

For simplicity, assume that $K_i = \mathbf{0}$ and $\mathbf{H}_m(s) = \mathbf{K}_m$ (PD control with fixed gain acceleration feedback). This yields the closed-loop system

$$\mathbf{H}\dot{\boldsymbol{v}} + [\mathbf{C}(\boldsymbol{v}) + \mathbf{D}(\boldsymbol{v}) + \mathbf{K}_d^*(\boldsymbol{\eta})]\boldsymbol{v} + \mathbf{J}_\Theta^\top(\boldsymbol{\eta})\mathbf{K}_p \tilde{\boldsymbol{\eta}} = \boldsymbol{w} \quad (12.113)$$

where $\tilde{\eta} = \eta - \eta_d$,

$$\mathbf{K}_d^*(\eta) = \mathbf{J}_\Theta^\top(\eta) \mathbf{K}_d \mathbf{J}_\Theta(\eta) \quad (12.114)$$

and

$$\mathbf{H} = \mathbf{M} + \mathbf{K}_m$$

In the stability analysis it is assumed that $\dot{\eta}_d = \mathbf{0}$, that is regulation of η to $\eta_d = \text{constant}$. A Lyapunov function candidate for this system is

$$V = \underbrace{\frac{1}{2} \mathbf{v}^\top \mathbf{H} \mathbf{v}}_{\substack{\text{kinetic} \\ \text{energy}}} + \underbrace{\frac{1}{2} \tilde{\eta}^\top \mathbf{K}_p \tilde{\eta}}_{\substack{\text{potential} \\ \text{energy}}} \quad (12.115)$$

where $\mathbf{H} = \mathbf{H}^\top > 0$ and $\mathbf{K}_p = \mathbf{K}_p^\top > 0$. Time differentiation of (12.115) along the trajectories of \mathbf{v} and $\tilde{\eta}$ yields

$$\begin{aligned} \dot{V} &= \mathbf{v}^\top \mathbf{H} \dot{\mathbf{v}} + \dot{\eta}^\top \mathbf{K}_p \tilde{\eta} \\ &= \mathbf{v}^\top (\mathbf{H} \dot{\mathbf{v}} + \mathbf{J}_\Theta^\top(\eta) \mathbf{K}_p \tilde{\eta}) \end{aligned} \quad (12.116)$$

since $\dot{\tilde{\eta}} = \dot{\eta} - \dot{\eta}_d = \dot{\eta}$ and $\dot{\eta}^\top = \mathbf{v}^\top \mathbf{J}_\Theta^\top(\eta)$. Substituting (12.113) into (12.116) yields

$$\begin{aligned} \dot{V} &= \mathbf{v}^\top (\mathbf{w} - [\mathbf{C}(\mathbf{v}) + \mathbf{D}(\mathbf{v}) + \mathbf{K}_d^*(\eta)] \mathbf{v}) \\ &= \mathbf{v}^\top \mathbf{w} - \mathbf{v}^\top [\mathbf{D}(\mathbf{v}) + \mathbf{K}_d^*(\eta)] \mathbf{v} \end{aligned} \quad (12.117)$$

since $\mathbf{v}^\top \mathbf{C}(\mathbf{v}) \mathbf{v} = 0$ for all \mathbf{v} ; see Property 7.2 in Section 7.5.

If $\mathbf{w} = \mathbf{0}$, Krasovskii–LaSalle’s Theorem A.2 in Appendix A.1 can be used to prove that the system (12.109)–(12.110) with nonlinear PD control ($\mathbf{K}_i = \mathbf{0}$) is *globally asymptotically stable* (GAS) if $\mathbf{J}_\Theta(\eta)$ is defined for all η (no representation singularity). Moreover, the trajectories will converge to the set Ω found from

$$\dot{V}(\mathbf{x}) = -\mathbf{v}^\top [\mathbf{D}(\mathbf{v}) + \mathbf{K}_d^*(\eta)] \mathbf{v} \equiv 0 \quad (12.118)$$

which is true for $\mathbf{v} = \mathbf{0}$. Therefore,

$$\Omega = \{(\tilde{\eta}, \mathbf{v}) : \mathbf{v} = \mathbf{0}\} \quad (12.119)$$

Now, $\mathbf{v} \equiv \mathbf{0}$ implies that $\mathbf{H} \dot{\mathbf{v}} = -\mathbf{J}_\Theta^\top(\eta) \mathbf{K}_p \tilde{\eta}$, which is nonzero as long as $\tilde{\eta} \neq \mathbf{0}$. Hence, the system cannot get “stuck” at an equilibrium point value other than $\tilde{\eta} = \mathbf{0}$. Since the equilibrium point $(\tilde{\eta}, \mathbf{v}) = (\mathbf{0}, \mathbf{0})$ is the largest invariant set M in Ω , the equilibrium point is GAS according to Theorem A.2.

In the case $\mathbf{w} \neq \mathbf{0}$ but $\dot{\mathbf{w}} = \mathbf{0}$, the system trajectories will converge to a ball about the origin $(\tilde{\eta}, \mathbf{v}) = (\mathbf{0}, \mathbf{0})$. The radius of the ball depends on the magnitude of the disturbance \mathbf{w} . This is referred to as uniform ultimate boundedness (UUB).

If integral action is included with $K_i > 0$ (PID control), it is possible to prove local asymptotic stability (LAS) also for the case $\mathbf{w} \neq \mathbf{0}$. This result is well known from robotics (Arimoto and Miyazaki, 1984). The bias term \mathbf{w} can also be removed by using parameter adaptation (Fossen *et al.*, 2001).

12.2.5 Case Study: Heading Autopilot for Ships and Underwater Vehicles

The principal blocks of a heading angle autopilot system, shown in Figure 12.18, are:

Control System: The feedback control system provides the necessary commands to track the desired yaw angle ψ_d . The output is the yaw moment τ_N .

Control Allocation: This module distributes the output from the feedback control system, usually the yaw moment τ_N , to the actuators (rudders and in some cases propellers and thrusters) in an optimal manner (see Section 12.3). For single-screw ships the controller yaw moment τ_N will simply be a function of the rudder command δ_c .

Reference Model: The autopilot reference model computes smooth trajectories ψ_d , r_d and \dot{r}_d needed for *course-changing* maneuvers. *Course-keeping* is the special case then $\psi_d = \text{constant}$ and $r_d = \dot{r}_d = 0$ (see Section 10.2.1).

Compass and Yaw Gyro: The compass measures the yaw angle ψ which is needed for feedback. In some cases a yaw rate gyro is available for yaw rate feedback, that is feedback from $r = \dot{\psi}$.

Observer/Wave Filter: In its simplest form the first-order wave-induced motion components ψ_w and r_w are filtered out from the measurements $y_1 = \psi + \psi_w$ and $y_2 = r + r_w$, and consequently prevented from entering the feedback loop. This is known as *wave filtering*, where the output of the filter is the LF motion components ψ and r . This is necessary to avoid excessive rudder action. In cases where y_2 is not measured the wave filter must be constructed as a state observer so that r can be estimated from the yaw angle measurement y_1 ; see Sections 11.3.5, 11.4.2 and 11.4.3.

Wind Feedforward: In cases where a wind sensor is available for *wind speed* and *direction*, a wind model can be used for wind feedforward. This is often advantageous since the integral action term in the PID controller does not have to integrate up the wind disturbance term. However, an accurate model of the wind force and moment as a function of ship speed and wind direction is needed to implement wind feedforward.

The different autopilot blocks of Figure 12.18 needed to implement a PID control law based on the Nomoto model will now be discussed.

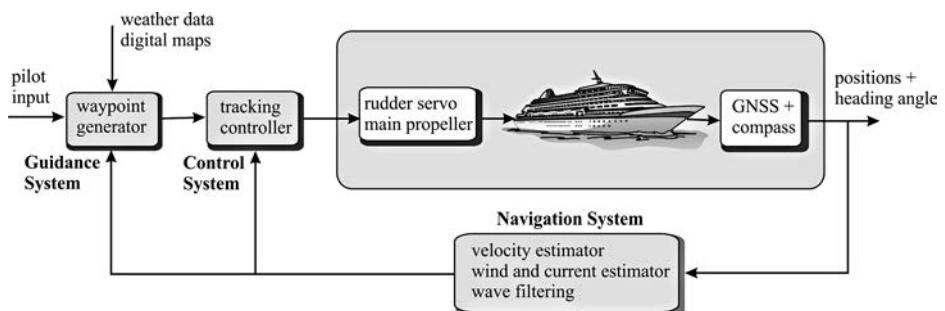


Figure 12.18 Block diagram of a heading autopilot system.

Autopilot Reference Model

A modern autopilot must have both course-keeping and turning capabilities. This can be obtained in one design by using a reference model to compute the desired states ψ_d , r_d and \dot{r}_d needed for turning, while

$$\psi_d = \text{constant} \quad (12.120)$$

can be treated as a special case of turning. A simple third-order filter for this purpose was derived in Section 10.2.1. Moreover,

$$\frac{\psi_d}{\psi_r}(s) = \frac{\omega_n^3}{(s + \omega_n)(s^2 + 2\zeta\omega_n s + \omega_n^2)} \quad (12.121)$$

where the reference ψ_r is the operator input, ζ is the relative damping ratio and ω_n is the natural frequency. Notice that

$$\lim_{t \rightarrow \infty} \psi_d(t) = \psi_r \quad (12.122)$$

and that $\dot{\psi}_d$ and $\ddot{\psi}_d$ are smooth and bounded for steps in ψ_r . This is the main motivation for choosing a third-order model since a second-order model will result in steps in $\dot{\psi}_d$ for steps in ψ_r .

In many cases it is advantageous to limit the desired yaw rate $|r_d| \leq r_{\max}$ during turning. This can be done by including a saturating element in the reference model (see Van Amerongen, 1982, 1984). The yaw acceleration $a_d = \dot{\psi}_d$ can also be limited such that $|a_d| \leq a_{\max}$ by using a second saturating element. The resulting state-space model including velocity and acceleration saturating elements becomes

$$\dot{\psi}_d = \text{sat}(r_d) \quad (12.123)$$

$$\dot{r}_d = \text{sat}(a_d) \quad (12.124)$$

$$\dot{a}_d = -(2\zeta + 1)\omega_n \text{sat}(a_d) - (2\zeta + 1)\omega_n^2 \text{sat}(r_d) + \omega_n^3(\psi_r - \psi_d) \quad (12.125)$$

The saturating element is defined as

$$\text{sat}(x) := \begin{cases} \text{sgn}(x)x_{\max} & \text{if } |x| \geq x_{\max} \\ x & \text{else} \end{cases} \quad (12.126)$$

The autopilot reference model has been simulated in Matlab with yaw rate limitation $r_{\max} = 1.0 \text{ deg/s}$, acceleration limit $a_{\max} = 0.5 \text{ deg/s}^2$ and command $\psi_r = 30 \text{ deg}$. The results are shown in Figure 12.19. Notice that the unlimited (linear) case yields unsatisfactorily high values for r_d .

The main motivation for using a rate-limiting element in the reference model is that the course-changing maneuver will be described by three phases (positive turn):

- I:** Start of turn, acceleration ($r_d > 0$ and $0 < \dot{r}_d \leq a_{\max}$)
- II:** Steady turning ($r_d = r_{\max}$ and $\dot{r}_d = 0$)
- III:** End of turn, deceleration ($r_d > 0$ and $-a_{\max} \leq \dot{r}_d < 0$)

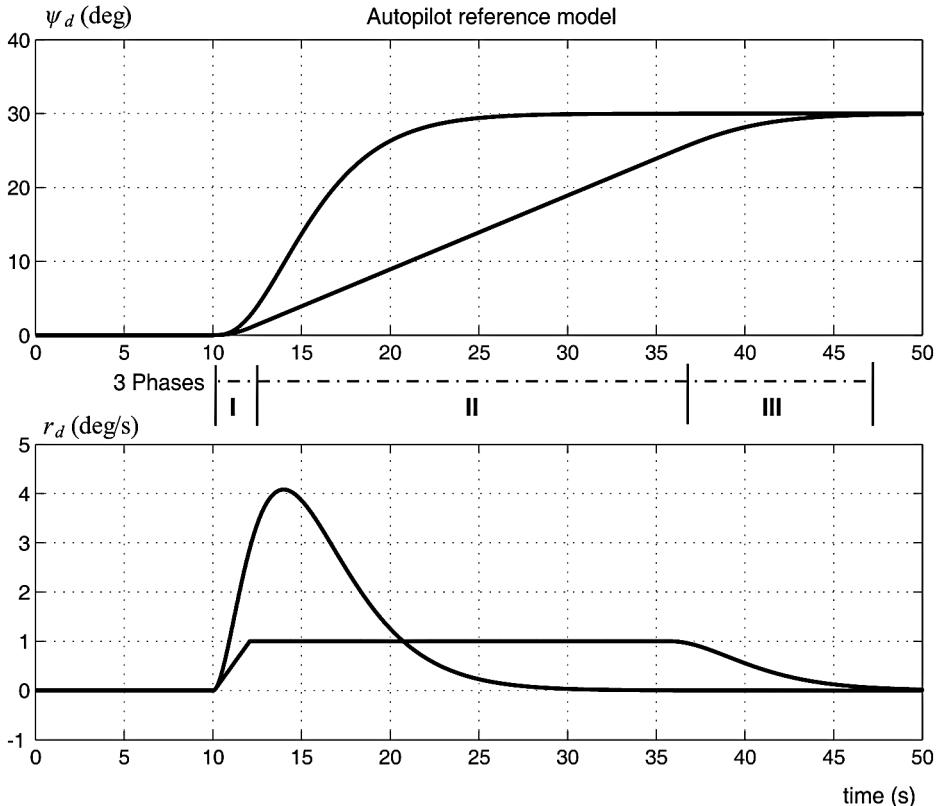


Figure 12.19 The plots show the effect of including a rate limiter of $r_{\max} = 1$ deg/s in a third-order reference model for heading. Notice that r_d becomes very high in the linear case while ψ_d looks satisfactory in both cases.

For a negative turn the signs of the turning rate and acceleration must be changed. The three phases are advantageous when performing a large change in course. The effect of a saturating element and nonlinear damping in a reference model are also demonstrated in Example 10.2 in Section 10.2.1.

A more sophisticated method for generating heading reference signals could be to use optimization techniques to compute the desired yaw angle, but then at the expense of a more complicated software algorithm to be implemented in real time.

Conventional PID Control

The autopilot systems of *Sperry* and *Minorsky* were both SISO control systems where the heading angle of the ship was measured by a gyro compass (see Section 9.1). Today, this signal is fed back to a computer in which a PID control system (autopilot) is implemented in software. The autopilot compares the operator setpoint (desired heading) with the measured heading and computes the rudder command, which is then transmitted to the rudder servo for corrective action.

The main difference between the autopilot systems of Sperry and Minorsky and the modern autopilot is the increased functionality that has been added with sophisticated features such as:

- Wave filtering, which avoids first-order wave forces being fed back to the actuators (see Section 11.3.5).
- Adaptation to varying environmental conditions, shallow water effects and time-varying model parameters, for instance changes in mass and center of gravity.
- Wind feedforward for accurate and rapid course-changing maneuvers.
- Reference feedforward using a dynamic model, ψ_d , r_d and \dot{r}_d , for course-changing maneuvers. Course-keeping is obtained by using a constant reference signal, $\psi_d = \text{constant}$, as input to the reference model.

Full State Feedback Control

Consider the Nomoto model of Section 7.2 in the following form:

$$(I_z - N_r)\dot{r} - N_r r = \tau_{\text{wind}} + \tau_N \quad (12.127)$$

where τ_{wind} is an optional input for wind feedforward and τ_N is the yaw moment generated by the controller. The constants $m = I_z - N_r$, $d = -N_r$ and

$$T = \frac{m}{d} = \frac{I_z - N_r}{-N_r} \quad (12.128)$$

are introduced such that

$$\dot{r} + \frac{1}{T}r = \frac{1}{m}(\tau_{\text{wind}} + \tau_N) \quad (12.129)$$

The yaw moment can be generated by a single rudder:

$$\tau_N = N_\delta \delta \quad (12.130)$$

or several actuators $u_i (i = 1, \dots, r)$ satisfying

$$\tau_N = \mathbf{b}^\top \mathbf{u}, \quad \mathbf{u} = [u_1, \dots, u_r]^\top \quad (12.131)$$

Assume that both ψ and r are measured by using a compass and a rate gyro. A PID controller for heading control is (see Section 12.2.3)

$$\tau_N(s) = -\hat{\tau}_{\text{wind}} + \tau_{\text{FF}}(s) - \underbrace{K_p \left(1 + T_d s + \frac{1}{T_i s} \right) \tilde{\psi}(s)}_{\tau_{\text{PID}}} \quad (12.132)$$

where τ_N is the controller yaw moment, τ_{FF} is a feedforward term to be decided, $\tilde{\psi} = \psi - \psi_d$ is the heading error and

$$K_p > 0 \quad \text{proportional gain constant}$$

$$T_d > 0 \quad \text{derivative time constant}$$

$$T_i > 0 \quad \text{integral time constant}$$

The wind feedforward term $\hat{\tau}_{\text{wind}}$ is an estimate of the wind moment τ_{wind} using wind coefficients and an anemometer measuring wind speed V_w and direction β_w . An estimate of the wind yaw moment can be computed according to (see Section 8.1)

$$\hat{\tau}_{\text{wind}} = \frac{1}{2} \rho_a V_{rw}^2 C_N(\gamma_{rw}) A_{Lw} L_{oa} \quad (12.133)$$

where the relative wind speed and angle of attack are

$$V_{rw} = \sqrt{u_{rw}^2 + v_{rw}^2} \quad (12.134)$$

$$\gamma_{rw} = -\text{atan2}(v_{rw}, u_{rw}) \quad (12.135)$$

The relative velocities depend on the heading angle ψ , wind direction β_w and wind speed V_w according to

$$u_{rw} = u - u_w = u - V_w \cos(\beta_w - \psi) \quad (12.136)$$

$$v_{rw} = v - v_w = v - V_w \sin(\beta_w - \psi) \quad (12.137)$$

When wind feedforward is implemented it is important that the wind measurements are low-pass filtered to avoid rapid changes in heading command. Wind feedforward is an optional term since the integrator in the PID control law can compensate for a slowly varying wind moment as well. The main difference will be the response time. In general, wind feedforward will be much faster than integral action since the integrator needs several minutes to remove a large wind component during the start-up of an autopilot system. Integral action works fairly well during fixed heading (stationkeeping and transit) while in a maneuvering situation large course deviations might be expected. Consequently, it is advantageous to implement wind feedforward to reduce the loads on the integrator and obtain maximum performance during start-up and in maneuvering situations. However, if the wind coefficients are poorly known, the closed-loop system can be destabilized by the wind feedforward term so care must be taken.

A continuous-time representation of the controller (12.132) is

$$\tau_N = -\hat{\tau}_{\text{wind}} + \tau_{FF} - K_p \tilde{\psi} - \underbrace{K_p T_d}_{K_d} \tilde{r} - \underbrace{\frac{K_p}{T_i}}_{K_i} \int_0^t \tilde{\psi}(\tau) d\tau \quad (12.138)$$

where $\tilde{r} := r - r_d$ and $\tilde{\psi} := \psi - \psi_d$. The controller gains can be found by pole placement; see Table 12.2 in Section 12.2.3. By specifying the control bandwidth ω_b , we get

$$\omega_n = \frac{1}{\sqrt{1 - 2\xi^2 + \sqrt{4\xi^4 - 4\xi^2 + 2}}} \omega_b \quad (12.139)$$

$$\begin{aligned} K_p &= m\omega_n^2 \\ K_d &= m \left(2\xi\omega_n - \frac{1}{T} \right) \stackrel{T \gg 0}{\approx} 2\xi\omega_n m \\ K_i &= \frac{\omega_n}{10} K_p \end{aligned}$$

The relative damping ratio ξ is usually chosen in the range 0.8–1.0, which means that the only tunable parameter is the control bandwidth ω_b (typically 0.01 rad/s for large tankers and 0.1 rad/s for smaller ships and underwater vehicles). This makes the system very easy to tune. However, it is important to have a good estimate of $m = I_z - N_f$ to obtain good performance.

Control Allocation

For a rudder-controlled craft, the input command is computed from (12.130), implying that

$$\delta = \frac{1}{N_\delta} \tau_N \quad (12.140)$$

In the case of several actuators, the generalized inverse can be used to compute \mathbf{u} from (12.131) if the scalar $\mathbf{b}^\top \mathbf{b} \neq 0$ (see Section 12.3). This gives

$$\mathbf{u} = \mathbf{b}(\mathbf{b}^\top \mathbf{b})^{-1} \tau_N \quad (12.141)$$

Reference Feedforward

The *feedforward* term τ_{FF} in (12.132) is determined such that perfect tracking during course-changing maneuvers is obtained. Using Nomoto's first-order model (12.129) as a basis for feedforward control, suggests that *reference feedforward* should be implemented according to

$$\tau_{FF} = m \left(\dot{r}_d + \frac{1}{T} r_d \right) \quad (12.142)$$

Substituting (12.142) and (12.132) into (12.129), the error dynamics becomes

$$\ddot{e} + \frac{1}{T}\dot{e} = \frac{1}{m} \tau_{PID} \quad (12.143)$$

where $e = \psi - \psi_d$. Since this system is linear, the closed-loop system can be analyzed in the frequency plane by using *Bode* plots. Consider the transfer function

$$h(s) = \frac{e}{\tau_{PID}}(s) = \frac{T/m}{s(Ts + 1)} \quad (12.144)$$

and let

$$\begin{aligned} h_{\text{PID}}(s) &= K_p \left(1 + T_d s + \frac{1}{T_i s} \right) \\ &= K_p \frac{T_i T_d s^2 + T_d s + 1}{T_i s} \end{aligned} \quad (12.145)$$

Hence, the loop transfer function becomes

$$\begin{aligned} l(s) &= h(s)h_{\text{PID}}(s) \\ &= \frac{T}{m} \frac{K_p (T_i T_d s^2 + T_i s + 1)}{T_i s^2 (T s + 1)} \end{aligned} \quad (12.146)$$

A frequently used approximation for (12.145) is found by assuming that $T_i \gg T_d$ such that $T_i \approx T_i + T_d$. Hence,

$$\begin{aligned} h_{\text{PID}}(s) &= K_p \left(1 + T_d s + \frac{1}{T_i s} \right) \\ &\approx K_p \frac{1 + (T_i + T_d)s + T_d T_i s^2}{T_i s} \\ &= K_p \frac{(1 + T_i s)(1 + T_d s)}{T_i s} \end{aligned} \quad (12.147)$$

Output Feedback using Only Compass Measurements

In many cases ships are only equipped with a gyrocompass for feedback control. If this is the case, the rate can be estimated using an observer, as shown in Sections 11.3.5 and 11.4.2. This approach also gives wave filtering. Alternatively, the D term in the controller must be replaced with a limited differentiator:

$$r(s) \approx \frac{T_d s}{\alpha T_d s + 1} \psi(s), \quad 0 < \alpha \ll 1 \quad (12.148)$$

such that the high-frequency components of $\psi(s)$ are filtered out. If we apply the low-pass filter

$$h_{\text{LP}}(s) = \frac{1}{\alpha T_d s + 1} \quad (12.149)$$

to all terms in the PID controller, (12.147) takes the form

$$h_{\text{PID}}(s) = K_p \frac{(T_i s + 1)(T_d s + 1)}{T_i s (\alpha T_d s + 1)}$$

The controller can be implemented in the time domain as

$$\tau_N = \tau_{FF} - K_p \tilde{\psi}_{LP} - \underbrace{K_p T_d}_{K_d} \tilde{r}_{LP} - \underbrace{K_p / T_i}_{K_i} \int_0^t \tilde{\psi}_{LP}(\tau) d\tau \quad (12.150)$$

with two filters:

$$\tilde{\psi}_{LP}(s) = \frac{1}{\alpha T_d s + 1} \tilde{\psi}(s), \quad \tilde{r}_{LP}(s) = \frac{T_d s}{\alpha T_d s + 1} \tilde{\psi}(s) \quad (12.151)$$

The parameter $0 < \alpha < 1$ is usually chosen as 0.1 while $T_i = 10T_d$, such that

$$\frac{1}{T_i} \ll \frac{1}{T_d} \ll \frac{1}{\alpha T_d} \quad (12.152)$$

12.2.6 Case Study: Heading Autopilot with Acceleration Feedback for Ships and Underwater Vehicles

An autopilot system can be extended to exploit acceleration feedback by differentiating the output of a yaw rate gyro r_{gyro} according to

$$\dot{r} \approx \frac{s}{s + \omega_f} r_{gyro} \quad (12.153)$$

The filter frequency ω_f must, however, be larger than the control bandwidth ω_b . In most cases this is easy to satisfy since ω_f can be chosen as high as 10–50 Hz if an accurate yaw-rate gyro is applied. A discrete-time representation of the filter (12.153) is found in Appendix B.3. This is particularly useful for smaller marine craft, which are more vulnerable to environmental forces than large marine craft. The main idea is to increase the moment of inertia by yaw rate feedback, such that external disturbances are suppressed; see Section 12.2.3. Consider the controller

$$\tau_N = -\hat{\tau}_{wind} + \underbrace{\tau_{FF} - K_p \tilde{\psi} - K_d \tilde{r} - K_i \int_0^t \tilde{\psi}(\tau) d\tau}_{\text{PID}} - \underbrace{K_m \dot{r}}_{\text{acceleration feedback}} \quad (12.154)$$

$$\tau_{FF} = (m + K_m) \left(\dot{r}_d + \frac{1}{T} r_d \right) \quad (12.155)$$

Notice that the term K_m must be included in τ_{FF} correspondingly. Substituting these expressions into (12.129) yields the closed-loop error dynamics

$$(m + K_m) \ddot{e} + \left(\frac{m}{T} + K_d \right) \dot{e} + K_p e + K_i \int_0^t e(\tau) d\tau = 0 \quad (12.156)$$

Based on Table 12.2 in Section 12.2.3, this suggests the following pole-placement algorithm for a critically damped system ($\zeta = 1$) with bandwidth ω_b :

$$\omega_n = 1.56\omega_b \quad (12.157)$$

$$K_p = (m + K_m)\omega_n^2 \quad (12.158)$$

$$K_d = 2\xi\omega_n(m + K_m) - \frac{m}{T} \stackrel{T \gg 0}{\approx} 2\xi\omega_n(m + K_m) \quad (12.159)$$

$$K_i = \frac{\omega_n}{10} K_p \quad (12.160)$$

where the additional moment of inertia K_m can be specified as a percentage (0–100 %) of the total moment of inertia m according to

$$K_m = \frac{\alpha}{100}m, \quad \alpha \in [0, 100] \quad (12.161)$$

The only tunable parameter in addition to K_m is the control bandwidth ω_b and this makes the system very easy to tune. However, it is important to have a good estimate of $m = I_z - N_r$ to obtain good performance.

A final implementation issue is the problem of first-order wave-induced forces. Using a wave filter for ψ , r and \dot{r} is recommended if all these signals are used in feedback. Wave filtering for systems using velocity and acceleration feedback is discussed by Lindegaard and Fossen (2001a) and Lindegaard (2003).

12.2.7 Case Study: Linear Cross-Tracking System for Ships and Underwater Vehicles

Often it is of primary importance to steer a ship, a submersible or a rig along a desired *path* with a prescribed *speed*. The path is usually defined in terms of *waypoints* using the *Cartesian* coordinates $(x_k, y_k) \in \mathbb{R}^2$. Waypoint guidance systems can be designed as trajectory-tracking controllers. In its simplest form this involves the use of a classical autopilot system where the yaw angle command ψ_d is generated such that the *cross-track error* is minimized. This can be done in a multivariable controller, for instance of \mathcal{H}_∞ or LQG type, or by including an additional PID tracking error control-loop in the autopilot. A waypoint trajectory-tracking system is usually designed such that the ship can move forward with reference speed U_d at the same time as the path cross-track error is minimized. The desired path can be generated using a route management system or by specifying the desired route by waypoints; see Section 10.2. If weather data are available, the optimal route can be generated such that the effects of wind and water resistance are minimized.

When designing a 3 DOF trajectory-tracking control system, the solution will depend on the number of available actuators. For most craft only two controls are needed: thrust T for *speed control* and a rudder δ for *steering control*.

Consider a path parametrized by two waypoints: $\mathbf{p}_k = [x_k, y_k]^\top$ and $\mathbf{p}_{k+1} = [x_{k+1}, y_{k+1}]^\top$, respectively. Next, we introduce a path-fixed reference frame $\{p\} = (x_p, y_p, z_p)$ with origin o_n in \mathbf{p}_k , whose x_p axis has been rotated a positive angle:

$$\alpha_k := \text{atan2}(y_{k+1} - y_k, x_{k+1} - x_k) \quad (12.162)$$

relative to the x axis of the inertial reference frame $\{n\} = \{x, y, z\}$. Recall from (10.56) that the along-track distance and cross-track errors are

$$s(t) = (x(t) - x_k) \cos(\alpha_k) + (y(t) - y_k) \sin(\alpha_k) \quad (12.163)$$

$$e(t) = -(x(t) - x_k) \sin(\alpha_k) + (y(t) - y_k) \cos(\alpha_k) \quad (12.164)$$

Consequently, the error term e represents the deviation to the path in the y direction in NED coordinates.

Since the craft is moving along a straight line, the sway velocity v and yaw angle ψ will be small. The cross-track error expressed in the path-fixed reference frame $\{p\}$ is

$$e = y_p \quad (12.165)$$

and the kinematic equations reduce to

$$\dot{x}_p = u \cos(\psi) - v \sin(\psi) \stackrel{v \approx 0 \text{ and } \psi \approx 0}{\approx} U \quad (12.166)$$

$$\dot{y}_p = u \sin(\psi) + v \cos(\psi) \stackrel{v \approx 0 \text{ and } \psi \approx 0}{\approx} U\psi \quad (12.167)$$

Consequently, the craft is moving with approximately constant speed $U = \sqrt{u^2 + v^2} \approx u$ along the path. A conventional cross-track controller is usually designed by using Nomoto's model in the following form:

$$\dot{y}_p = U\psi \quad (12.168)$$

$$\dot{\psi} = r \quad (12.169)$$

$$Tr + r = K\delta + b \quad (12.170)$$

$$\dot{b} = 0 \quad (12.171)$$

where b is a bias term and δ is the control input. Consequently,

$$e(s) = h_\delta(s)\delta(s) + h_b(s)b(s) \quad (12.172)$$

where

$$h_\delta(s) = \frac{e}{\delta}(s) = \frac{KU}{s^2(1 + Ts)} \quad (12.173)$$

$$h_b(s) = \frac{e}{b}(s) = \frac{U}{s^2(1 + Ts)} \quad (12.174)$$

This is a linear system and it is straightforward to design a PID controller:

$$\delta = -K_p e - K_d \dot{e} - K_i \int_0^t e(\tau) d\tau \quad (12.175)$$

for the regulation of e to zero. Integral action is needed in order to compensate for the bias term b representing environmental forces and the rudder offset.

12.2.8 Case Study: LOS Path-Following Control for Ships and Underwater Vehicles

A line-of-sight (LOS) path-following controller can be designed for conventional craft by representing the desired path by waypoints, as described in Section 10.3. This is particularly useful for underwater vehicles and surface vessels in transit operations where the user can specify the path by straight lines using a digital chart. For curved paths, the approach of Section 12.2.9 can be used.

If the craft is equipped with a conventional heading autopilot, an outer feedback loop representing the guidance system can be designed as shown in Figure 12.20. This is practical since a commercial autopilot system can be treated as a black box where the outer-loop LOS algorithm computes heading commands to the autopilot. For this purpose, the guidance laws of Section 10.3.2 can be used to steer along the LOS vector which again forces the craft to track the path. When designing path-following control systems both the desired *heading* and *course* angles can be used since

$$\psi_d = \chi_d - \beta \quad (12.176)$$

where the sideslip angle is given by

$$\beta = \arcsin\left(\frac{v}{U}\right) \quad (12.177)$$

Notice that β depends on the surge and sway velocities, implying that β must be computed using Doppler or GNSS velocity measurements, for instance. Alternatively, a state estimator for β can be designed.

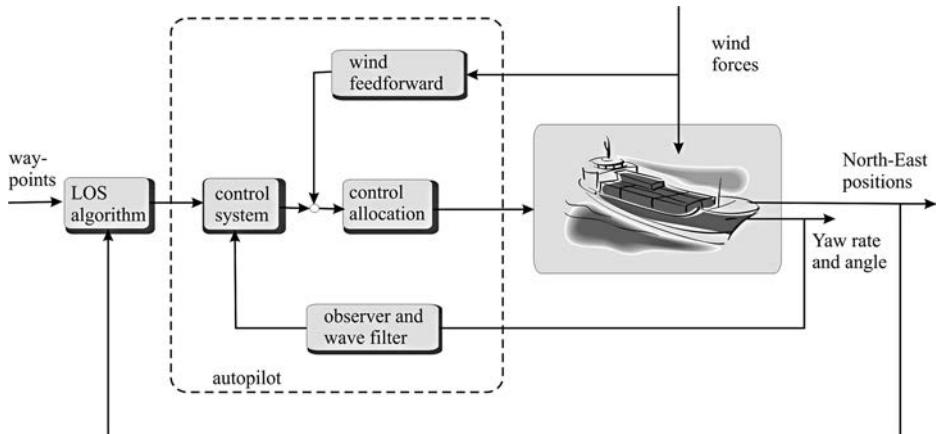


Figure 12.20 Conventional autopilot used in conjunction with an LOS guidance algorithm in the outer loop.

The heading autopilot is usually a PID controller with feedforward (see Section 12.2.6):

$$\tau_N = -\hat{\tau}_{wind} + \tau_{FF} - K_p \tilde{\psi} - K_d \dot{\tilde{\psi}} - K_i \int_0^t \tilde{\psi}(\tau) d\tau \quad (12.178)$$

$$\tau_{FF} = m \left(\dot{r}_d + \frac{1}{T} r_d \right) \quad (12.179)$$

where $\tilde{\psi} = \psi - \psi_d$, and $K_p > 0$, $K_d > 0$ and $K_i > 0$ are the controller gains.

Body x axis aligned to the LOS vector: If the sideslip angle β is unknown, the body x axis of the craft can be aligned with the LOS vector to the price of a tracking offset (see Section 10.3.2).

Enclosure-based steering: In this approach, the desired heading angle ψ_d is chosen as:

$$\psi_d = \chi_d = \text{atan2}(y_{los} - y, x_{los} - x) \quad (12.180)$$

and the body x axis of the craft points in the direction of the LOS intersection point $p_{los}^n = [x_{los}, y_{los}]^\top$, as shown Figure 10.9.

Velocity and LOS vectors aligned: In order to align the velocity and LOS vectors, the desired course angle χ_d must be specified such that the velocity vector points towards the intersection point p_{los} . The course angle command χ_d needed to accomplish this can be computed using one of the following guidance algorithms (see Section 10.3.2):

Enclosure-based steering: The course angle is chosen as

$$\chi_d = \text{atan2}(y_{los} - y, x_{los} - x) \quad (12.181)$$

and mapped into a heading command by

$$\begin{aligned} \psi_d &= \chi_d - \beta \\ &= \chi_d - \arcsin\left(\frac{v}{U}\right) \end{aligned} \quad (12.182)$$

This approach requires velocity measurements.

Lookahead-based steering: The course angle command (Breivik and Fossen, 2009)

$$\chi_d = \chi_p + \chi_r(e) \quad (12.183)$$

is chosen as the sum of the *path-tangential angle* χ_p and the *velocity-path relative angle* $\chi_r(e)$ to ensure that the velocity is directed toward a point on the path that is located a *lookahead distance* $\Delta > 0$ (Papoulias, 1991) ahead of the direct projection of p^n on to the path. In this context

$$\chi_p = \alpha_k \quad (12.184)$$

$$\chi_r(e) = \arctan(-K_p e) \quad (12.185)$$

where χ_r is a proportional controller and $e(t)$ is the cross-track error given by

$$e(t) = -[x(t) - x_k] \sin(\alpha_k) + [y(t) - y_k] \cos(\alpha_k) \quad (12.186)$$

Lookahead-based steering can be implemented in terms of the heading controller (12.178) by using the transformation

$$\begin{aligned}\psi_d &= \chi_d - \beta \\ &= \chi_p + \chi_r - \beta\end{aligned}\quad (12.187)$$

If the sideslip angle β is unknown, a PI controller

$$\chi_r(e) = \arctan \left(-K_p e - K_i \int_0^t e(\tau) d\tau \right) \quad (12.188)$$

together with the approximation $\psi_d \approx \chi_p + \chi_r$ can be used to compensate for sideslip.

When moving along a piece wise linear path made up of n straight-line segments connected by $n + 1$ waypoints, a switching mechanism for selecting the next waypoint is needed. Waypoint (x_{k+1}, y_{k+1}) can be selected on a basis of whether the craft lies within a *circle of acceptance* with radius R_{k+1} around (x_{k+1}, y_{k+1}) . Moreover, if the craft positions (x, y) at time t satisfy

$$[x_{k+1} - x(t)]^2 + [y_{k+1} - y(t)]^2 \leq R_{k+1}^2 \quad (12.189)$$

the next waypoint (x_{k+1}, y_{k+1}) should be selected. This is described more closely in Section 10.3.2, which also discusses extensions from 2-D to 3-D path-following control.

12.2.9 Case Study: Path-Following Control for Ships and Underwater Vehicles using Serret-Frenet Coordinates

In Section 10.4.2 a guidance law for curved parametrized paths was presented. The guidance law assumes that there exists a parametrized path

$$\mathbf{P}_d^\eta(\varpi) = \begin{bmatrix} x_d(\varpi) \\ y_d(\varpi) \end{bmatrix} \quad (12.190)$$

as a function of the path variable ϖ . The main idea is to use a kinematic controller to compute yaw commands r_d to a yaw rate feedback control system that turns the marine craft such that the predefined path is followed. In 2-D this is a simple rotation about the vertical axis. The kinematic controller can be designed using a dynamic model of the marine craft by specifying a reference frame that moves along the path. This reference frame is usually chosen as the *Serret–Frenet frame* (see Frenet, 1847, Serret, 1851), as shown in Figure 10.19. The kinematic controller can be implemented in cascade with the yaw rate controller, as illustrated in Figure 12.21.

We will look at the implementation aspects of the path-following controller by considering the yaw dynamics of the marine craft in the following form:

$$(I_z - N_r)\dot{r} - N_r r = \tau_N \quad (12.191)$$

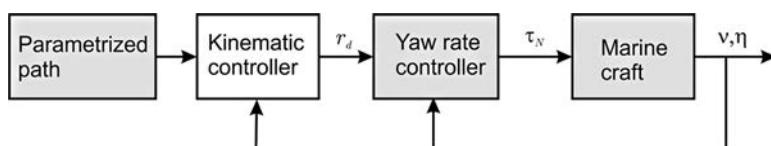


Figure 12.21 Cascaded kinematic and yaw rate controller for path-following control.

where $I_z - N_r > 0$ and $-N_r > 0$ are constant parameters. The controller yaw moment τ_N can easily be designed to regulate r to r_d , for instance by using the following feedback control law:

$$\tau_N = -N_r r - K_p(r - r_d) \quad (12.192)$$

where $K_p > 0$ is a design parameter and r_d is the desired yaw rate generated by the kinematic controller. In Section 10.4.2 it was shown that

$$r_d = \left(1 - \frac{(m - X_{\dot{u}})}{(m - Y_{\dot{v}})}\right)^{-1} \left(\dot{\chi}_d + \kappa U_d - K_1 \tilde{\chi}_{SF} - \frac{Y_v}{(m - Y_{\dot{v}})} \left(\tan(\beta) - \frac{v_c}{U \cos(\beta)} \right) \right) \quad (12.193)$$

where $\tilde{\chi}_{SF} = \chi_{SF} - \chi_d$ renders the equilibrium point $(s, e, \tilde{\chi}_{SF}) = (0, 0, 0)$ UGAS and ULES. Unfortunately, this expression requires knowledge of the sideslip angle

$$\beta = \arcsin\left(\frac{v}{U}\right) \quad (12.194)$$

and the current velocity v_c . One way to avoid this is to replace the term proportional to $\tan(\beta)$ with an integral term. This of course is based on the assumption that the sideslip angle changes slowly. The PI guidance law then takes the following form:

$$r_d = \left(1 - \frac{(m - X_{\dot{u}})}{(m - Y_{\dot{v}})}\right)^{-1} \left(\dot{\chi}_d + \kappa U_d - K_p \tilde{\chi}_{SF} - K_i \int_0^t \tilde{\chi}_{SF}(\tau) d\tau \right) \quad (12.195)$$

where $K_p = 2\lambda$ and $K_i = \lambda^2$ are parametrized using $\lambda > 0$ as a design parameter and (see Section 10.4.2):

$$\chi_d = \arctan\left(\frac{-e}{\Delta}\right) \quad (12.196)$$

$$U_d = U \cos(\chi_{SF}) + K_2 s \quad (12.197)$$

The dynamic equations of the guidance law are

$$\dot{s} = U \cos(\chi_{SF}) - (1 - \kappa e) U_d \quad (12.198)$$

$$\dot{e} = U \sin(\chi_{SF}) - \kappa U_d s \quad (12.199)$$

$$\dot{\chi}_{SF} = r + \dot{\beta} - \kappa U_d \quad (12.200)$$

An alternative approach to integral action is to use a state estimator to estimate the ocean currents (Encarnacao *et al.*, 2000). Cascaded design techniques based on backstepping and Lyapunov analysis are discussed by Lapierre and Soetanto (2007) and Børhaug and Pettersen (2006), while Breivik and Fossen (2004a) present an alternative approach for Serret–Frenet path-following control where the path curvature κ is superfluous.

12.2.10 Case Study: Dynamic Positioning Control System for Ships and Floating Structures

Control systems for stationkeeping and low-speed maneuvering are commonly known as dynamic positioning (DP) systems. The Norwegian classification society DnV (1990) defines a DP vessel according to:

Dynamically positioned vessel: a free-floating vessel which maintains its position (fixed location or predetermined track) exclusively by means of thrusters.

It is, however, possible to exploit rudder forces in DP also by using the propeller to generate rudder lift forces (Lindegard and Fossen, 2003).

For ships that are anchored, additional spring forces are introduced into the control model. These systems are referred to as position mooring (PM) systems (see Section 12.2.11). Optimality with respect to changing weather conditions will be discussed in Section 13.3.10 using the concept of weather optimal positioning control (WOPC).

DP and PM Systems

In the 1960s, systems for automatic control of the horizontal position, in addition to the heading, were developed. Systems for the simultaneous control of the three horizontal motions (surge, sway and yaw) are today commonly known as *DP systems* and are used in a wide range of marine operations such as stationkeeping, drilling and offloading, as illustrated in Figure 12.22. More recently anchored positioning systems or *PM systems* have been designed; see Section 12.2.11. For a free-floating vessel the thrusters are the prime actuators for stationkeeping, while for a PM system the assistance of thrusters are only complementary since most of the position-keeping is provided by a deployed anchor system. Different DP applications are described more closely in Strand and Sørensen (2000).

DP systems have traditionally been a *low-speed* application, where the basic DP functionality is either to keep a fixed position and heading or to move slowly from one location to another (*marked positioning*). In addition, specialized tracking functions for cable and pipe-layers, and operations of ROVs have been included. The traditional *autopilot* and *waypoint-tracking* functionalities have also been included in modern DP systems. The trend today is that *high-speed* operation functionality merges with classical DP functionality, resulting in a *unified system* for all speed ranges and types of operations.

The first DP systems were designed using conventional PID controllers in cascade with low-pass and/or notch filters to suppress the wave-induced motion components. This was based on the assumption that the interactions were negligible (Sargent and Cowgill, 1976, and Morgan, 1978). From the middle of the 1970s a new model-based control concept utilizing stochastic optimal control theory and Kalman filtering techniques was employed with the DP problem by Balchen *et al.* (1976). The Kalman filter is used to separate the LF and WF motion components such that only feedback from the LF motion components is used (see Chapter 11). Later extensions and modifications of this work have been proposed by numerous authors; see Balchen *et al.* (1980a, 1980b), Grindle *et al.* (1980a, 1980b), Fung and Grindle (1983), Sælid *et al.* (1983) and more lately Fossen *et al.* (1996), Sørensen *et al.* (1996, 2000), Fossen and Grøvlen (1998) and Fossen and Strand (1999a).

Roll and Pitch Damping in DP

Traditionally DP systems have been designed for 3 DOF low-speed trajectory-tracking control by means of thrusters and propellers. However, extensions to 5 DOF control for the purpose of roll and pitch damping of semi-submersibles has been proposed by Sørensen and Strand (1998). It is well known that for marine structures with a small waterplane area and low metacentric height, which results in relatively low hydrostatic restoration compared to the inertia forces, an unintentional coupling phenomenon between

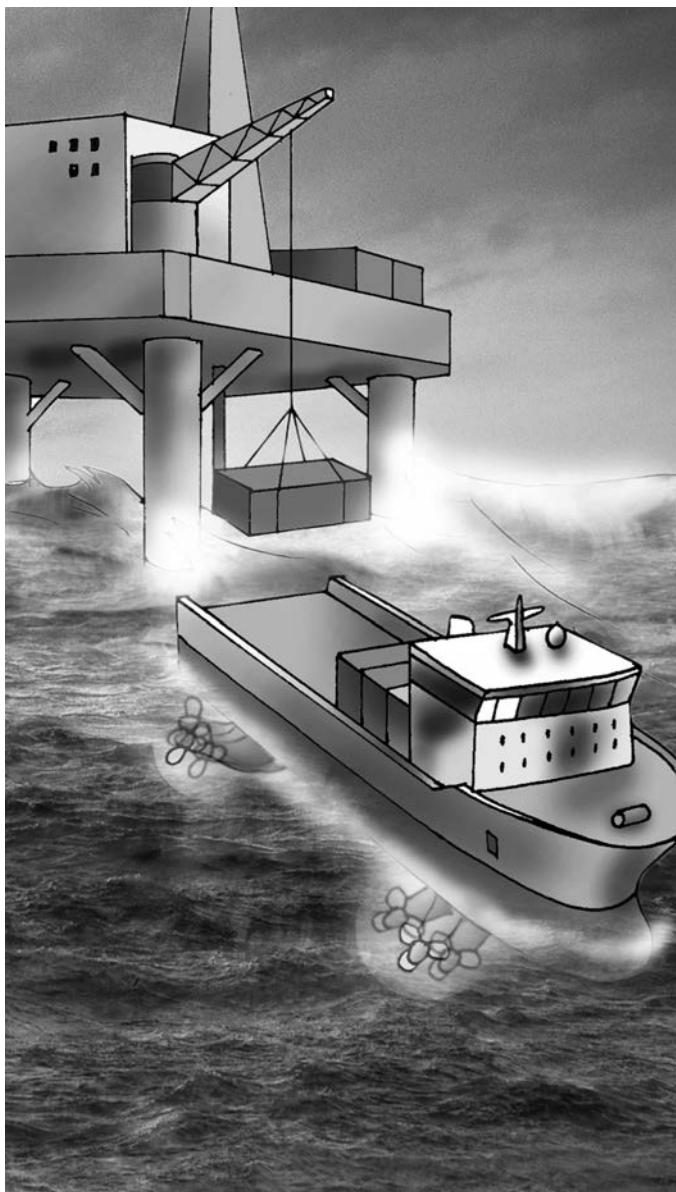


Figure 12.22 Dynamically positioned supply vessel used in offshore offloading. Illustration by Bjarne Stenberg/Department of Marine Technology, NTNU.

the vertical and the horizontal planes through the thruster action can be invoked. Examples are found in semi-submersibles and SWATHs, which typically have natural periods in roll and pitch in the range of 35–65 s. If the inherent vertical damping properties are small, the amplitudes of roll and pitch may be emphasized by the thruster's induction by up to 2° – 5° in the resonance range. These oscillations have

caused discomfort to the vessel's crew and have in some cases limited the operation. Hence, the motions in both the horizontal and vertical planes should be considered in the controller design, as proposed in Sørensen and Strand (2000).

Optimal Setpoint Chasing in DP for Drilling and Intervention Vessels

Further extension in the development of DP systems includes extended functionality adapted for the particular marine operation considered. In Sørensen *et al.* (2001) a function for optimal setpoint chasing in DP of drilling and intervention vessels is proposed in order to minimize riser angle offsets at the sea bed and on the vessel.

Controller and Observer Models

For DP systems an LF controller model will be employed for feedback since dynamics at higher frequencies are negligible in stationkeeping. Recall from Section 7.3.2 that

$$\dot{\eta}_p = v \quad (12.201)$$

$$M\ddot{v} + Dv = b_p + \tau + \tau_{wind} + \tau_{wave} \quad (12.202)$$

where VP coordinates have been employed (see Section 7.5.3). The bias term is expressed in $\{b\}$ using the transformation $b_p = R(\psi)^T b$.

The North-East positions and heading measurements are related to η_p by

$$\eta = R(\psi)\eta_p \quad (12.203)$$

In some cases additional measurements are available such as GNSS and Doppler log velocity v as well as anemometer measurements, which can be used to compute an estimate of the generalized wind forces τ_{wind} . The bias b is treated as an unknown state due to wave drift, ocean currents and unmodeled dynamics.

DP Control System

The craft is exposed to *waves*, *ocean currents* and *wind*. The observer–controller must be robust and compensate for environmental forces and unmodeled dynamics. These are the most important design requirements in an industrial vessel control system since a full-state feedback controller will not work in bad weather unless the environmental forces are included in the design specifications. In commercial DP systems it is therefore necessary to include the following features:

- *Integral action* to compensate for slowly varying forces (bias term b) due to ocean currents, second-order wave drift forces and unmodeled dynamics.
- *Wind feedforward control* to compensate for *mean* wind forces. Wind gust cannot be compensated for since the actuators do not have the capacity for moving a large vessel in the frequency range of the wind gust.
- *Wave filtering* to avoid where first-order wave-induced oscillations are fed back to the control system as explained in Chapter 11. This is an important feature since the actuators cannot move a large vessel fast enough to suppress the disturbances.
- *State estimator* for noise filtering and estimation of unmeasured states, for instance linear and angular velocities. The main tool for this is the Kalman filter, alternatively nonlinear and passive observers as described in Sections 11.3–11.4.
- *Optimal allocation of thrust* where the main goal is to compute optimal setpoints for thrusters, rudders and other actuators based on the force and moment commands generated by the DP control system. This is treated in detail in Section 12.3.

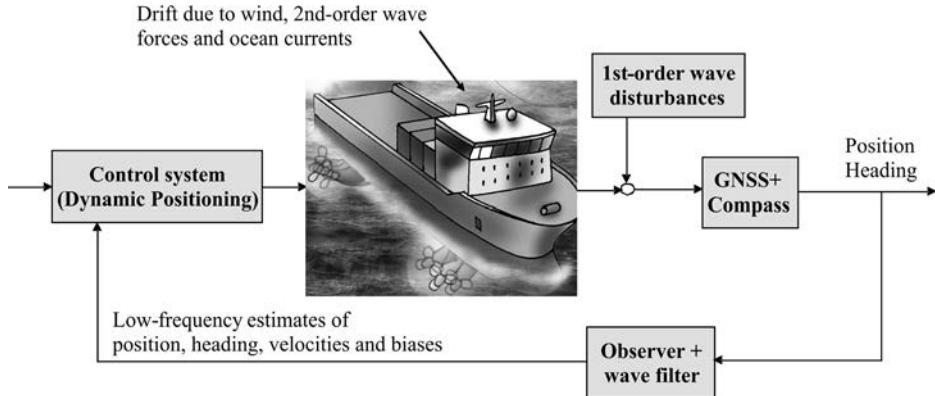


Figure 12.23 Dynamic positioning system. The observer can be implemented as a Kalman filter or a passive observer. Illustration by Bjarne Stenberg.

The different blocks in a closed-loop DP system is shown in Figure 12.23. The control system can be designed as a MIMO nonlinear PID controller using the results in Section 12.2.4. Moreover,

$$\tau = -\hat{\tau}_{\text{wind}} + \mathbf{R}^T(\eta)\boldsymbol{\tau}_{\text{PID}} \quad (12.204)$$

where $\hat{\tau}_{\text{wind}}$ is an estimate of the generalized wind forces and the PID controller is expressed in $\{n\}$ according to:

$$\boldsymbol{\tau}_{\text{PID}} = -\mathbf{K}_p \tilde{\eta} - \mathbf{K}_d \dot{\eta} - \mathbf{K}_i \int_0^t \tilde{\eta}(\tau) d\tau \quad (12.205)$$

By combining (12.204) and (12.205), the DP control law becomes

$$\boldsymbol{\tau} = -\hat{\tau}_{\text{wind}} - \mathbf{R}^T(\eta)\mathbf{K}_p \tilde{\eta} - \underbrace{\mathbf{R}^T(\eta)\mathbf{K}_d\mathbf{R}(\eta)}_{\mathbf{K}_d^*} \mathbf{v} - \mathbf{R}^T(\eta)\mathbf{K}_i \int_0^t \tilde{\eta}(\tau) d\tau \quad (12.206)$$

where

$$\mathbf{K}_d^* := \mathbf{R}^T(\eta)\mathbf{K}_d\mathbf{R}(\eta) \quad (12.207)$$

It is common to choose \mathbf{K}_d as a diagonal matrix and thus $\mathbf{K}_d^* = \mathbf{K}_d$. For the full-state feedback case, asymptotic stability follows using Lyapunov arguments (see Section 12.2.4). However, in order to implement the nonlinear PID controller a state estimator and wave filter must be designed. This is straightforward for the linearized DP model (12.201)–(12.203) where additional states for the WF motions can be augmented and used directly in a Kalman filter (see Section 11.3.6). GAS and convergence of the nonlinear PID controller (12.206) in combination with a linear Kalman filter cannot be guaranteed but the solution has been used in many industrial systems with excellent performance and robustness. Hence, from a practical point of view this is indeed a well-proven concept. An alternative approach could be to use linear vessel

parallel coordinates and a separation principle to guarantee asymptotic stability via a linear quadratic Gaussian optimal control philosophy (see Section 13.1.6).

It is, however, possible to prove UGAS for the nonlinear PID controller in combination with the nonlinear passive observer of Section 11.4 under certain conditions. Consider the passive observer:

$$\dot{\hat{\xi}} = A_w \hat{\xi} + K_1(\omega_o) \tilde{y} \quad (12.208)$$

$$\dot{\hat{\eta}} = R(y_3) \hat{v} + K_2 \tilde{y} \quad (12.209)$$

$$\dot{\hat{b}} = -T^{-1} \hat{b} + K_3 \tilde{y} \quad (12.210)$$

$$M \dot{\hat{v}} = -D \hat{v} + R^\top(y_3) \hat{b} + \tau + \tau_{\text{wind}} + R^\top(y_3) K_4 \tilde{y} \quad (12.211)$$

$$\hat{y} = \hat{\eta} + C_w \hat{\xi} \quad (12.212)$$

where drift is estimated using the bias term \hat{b} . For the DP controller (12.206), the drift forces have been compensated for by adding integral action in the controller. A PD controller motivated by (12.206), where slowly varying environmental forces are compensated by using the observer bias estimates, $R^\top(\psi) \hat{b}$, has been proposed by Loria *et al.* (2000):

$$\tau = -\hat{\tau}_{\text{wind}} - R^\top(\psi) K_p (\hat{\eta} - \eta_d) - K_d^* \hat{v} - R^\top(\psi) \hat{b} \quad (12.213)$$

Notice that the integral term in the controller (12.206) is removed and replaced by the bias estimate. It is then possible to show that the equilibrium point of the observer–controller is UGAS. The stability proof is based on a *separation principle*, which holds for nonlinear systems (Loria *et al.*, 2000).

Wind Feedforward

It is possible to implement wind feedforward τ_{wind} in DP control systems. However, this requires that the wind forces and moment are known as functions of the wind speed and direction, as well as ship hull parameters. Different wind models are presented in Section 8.1, suggesting that

$$\hat{\tau}_{\text{wind}} = \frac{1}{2} \rho_a V_{rw}^2 \begin{bmatrix} C_X(\gamma_{rw}) A_{F_w} \\ C_Y(\gamma_{rw}) A_{L_w} \\ C_N(\gamma_{rw}) A_{L_w} L_{oa} \end{bmatrix} \quad (12.214)$$

where the relative wind speed and angle of attack are

$$V_{rw} = \sqrt{u_{rw}^2 + v_{rw}^2} \quad (12.215)$$

$$\gamma_{rw} = -\text{atan}2(v_{rw}, u_{rw}) \quad (12.216)$$

The relative velocity components depend on the heading angle ψ , wind direction β_w and wind speed V_w according to

$$u_{rw} = u - u_w = u - V_w \cos(\beta_w - \psi) \quad (12.217)$$

$$v_{rw} = v - v_w = v - V_w \sin(\beta_w - \psi) \quad (12.218)$$

When wind feedforward is implemented, it is important that the wind measurements are low-pass filtered to avoid rapid changes in the actuator commands. Wind feedforward is an optional term since the integrator in the DP system can compensate for slowly varying wind forces as well. The main difference will be the response time. In general, wind feedforward will be much faster than integral action since the integrator needs several minutes to remove a large wind component during the start-up of the DP system.

12.2.11 Case Study: Position Mooring Control System for Ships and Floating Structures

Figure 12.24 illustrates different mooring strategies for ships and floating structures. The results of Section 12.2.10 can be generalized to PM systems by adding a spring to the model. Consider the model

$$\dot{\eta}_p = v \quad (12.219)$$

$$M\ddot{v} + Dv + K_p\eta_p = b_p + \tau + \tau_{wind} + \tau_{wave} \quad (12.220)$$

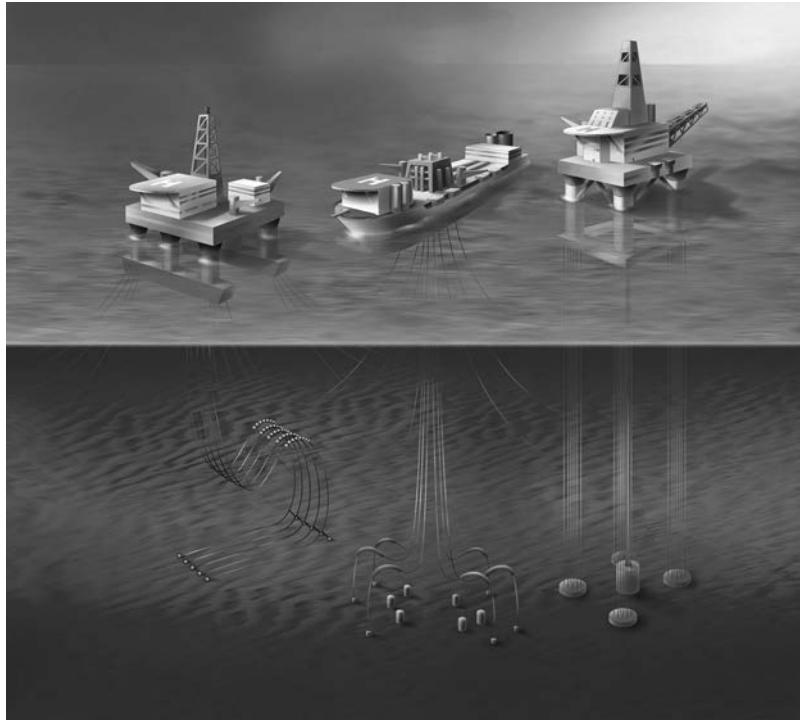


Figure 12.24 Mooring systems for a submersible, FPSO and platform. Illustration by Bjarne Stenberg.

where $\boldsymbol{v} = [u, v, r]^\top$ and $\boldsymbol{\eta} = [N, E, \psi]^\top$. For this system:

$$\mathbf{M} = \mathbf{M}^\top = \begin{bmatrix} m_{11} & 0 & 0 \\ 0 & m_{22} & m_{23} \\ 0 & m_{32} & m_{33} \end{bmatrix} \quad (12.221)$$

$$\mathbf{D} = \mathbf{D}^\top = \begin{bmatrix} d_{11} & 0 & 0 \\ 0 & d_{22} & d_{23} \\ 0 & d_{32} & d_{33} \end{bmatrix} \quad (12.222)$$

$$\mathbf{K} = \text{diag}\{k_{11}, k_{22}, k_{33}\} \quad (12.223)$$

The additional spring $\mathbf{K}\boldsymbol{\eta}_p$ due to the mooring system adds spring stiffness in surge, sway and yaw described by the parameters $k_{11} > 0$, $k_{22} > 0$ and $k_{33} \geq 0$. With this in mind, two different design philosophies for mooring systems are quite common:

- *Turret mooring systems* have cables that are connected to the turret via bearings. This allows the vessel to rotate around the anchor legs. In this case, the rotational spring can be neglected such that $k_{33} = 0$. The turret can be mounted either internally or externally. An external turret is fixed, with appropriate reinforcements, to the bow or stern of the ship. In the internal case the turret is placed within the hull in a moon pool. A moon pool is a wet porch, that is an opening in the floor or base of the hull giving access to the water below, allowing technicians or researchers to lower tools and instruments into the sea. Turret mooring systems allow the vessel to rotate in the horizontal plane (yaw) into the direction where environmental loading due to wind, waves and ocean currents is minimal. This is referred to as weathervaning.
- *Spread mooring systems* are used to moor Floating Production, Storage and Offloading (FPSO) units, tankers and floating platforms (see Figure 12.24). The system consists of mooring lines attached somewhere to the vessel. The drawback with a spread mooring system is that it restrains the vessel from rotating ($k_{33} > 0$) and hence weathervaning is impossible. On the other hand, it is relatively inexpensive to equip an existing vessel with mooring lines that can be attached directly to the hull.

For thruster-assisted PM systems the thrusters are complementary to the mooring system and the main idea is to provide the system with additional damping, for instance by using a D controller:

$$\boldsymbol{\tau} = -\mathbf{K}_d \boldsymbol{v} \quad (12.224)$$

The mooring term $\mathbf{K}\boldsymbol{\eta}_p$ is in fact a P controller but additional spring forces can be included by position feedback if necessary. Integral action is not used in PM systems, since the ship is only allowed to move within a limited radius from the equilibrium point or field-zero point (FZP). If the vessel moves outside the specified radius of the mooring system, a stabilizing control system of PD type can be used to drive the vessel inside the circle again. This is usually done in an energy perspective since it is important to reduce the fuel consumption of PM systems. Consequently, in bad weather it will be more optimal to use additional thrust to stay on the circle rather than move the vessel to the FZP. In good weather, no control action is needed since the vessel is free to move within the circle.

PM systems have been commercially available since the 1980s, and provide a flexible solution for floating structures for drilling and oil and gas exploitation on the smaller and marginal fields (Sørensen *et al.*, 2000). Modeling and control of turret-moored ships are complicated problems since the mooring

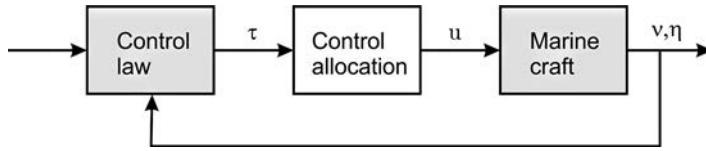


Figure 12.25 Block diagram showing the control allocation block in a feedback control system.

forces and moments are inherently nonlinear (Strand *et al.*, 1998). The control design of PM systems using nonlinear theory is addressed by Strand (1999).

12.3 Control Allocation

For marine craft in n DOF it is necessary to distribute the generalized control forces $\tau \in \mathbb{R}^n$ to the actuators in terms of control inputs $u \in \mathbb{R}^r$ as shown in Figure 12.25 (Fossen and Johansen, 2006). If $r > n$ this is an *overactuated control* problem while $r < n$ is referred to as *underactuated control*; see the discussion in Section 9.4. The input matrix is square for $r = n$, that is the number of actuators is equal to the number of DOFs.

Computation of u from τ is a model-based optimization problem, which in its simplest form is unconstrained while physical limitations such as input amplitude and rate saturations imply that a constrained optimization problem must be solved. Another complication is actuators that can be rotated at the same time as they produce control forces. An example is azimuth thrusters on an offshore supply vessel. This increases the number of available controls from r to $r + p$, where p denotes the number of rotatable actuators for which additional nonlinearities are introduced.

12.3.1 Actuator Models

The control force due to a propeller, a rudder or a fin can be written (assuming linearity)

$$F = ku \quad (12.225)$$

where k is the force coefficient and u is the control input depending on the actuator considered; see Table 12.3. The linear model $F = ku$ can also be used to describe nonlinear monotonic control forces. For instance, if the rudder force F is quadratic in rudder angle δ , that is $F = k \delta |\delta|$, the choice $u = \delta |\delta|$, which has a unique inverse $\delta = \text{sgn}(u)\sqrt{|u|}$, satisfies (12.225).

Table 12.3 Definition of actuators and control variables

Actuator	u (control input)	α (control input)	f^\top (force vector)
Main propellers (longitudinal)	Pitch and rpm	–	$[F, 0, 0]$
Tunnel thrusters (transverse)	Pitch and rpm	–	$[0, F, 0]$
Azimuth (rotatable) thruster	Pitch and rpm	Angle	$[F \cos(\alpha), F \sin(\alpha), 0]$
Aft rudders	Angle	–	$[0, F, 0]$
Stabilizing fins	Angle	–	$[0, 0, F]$

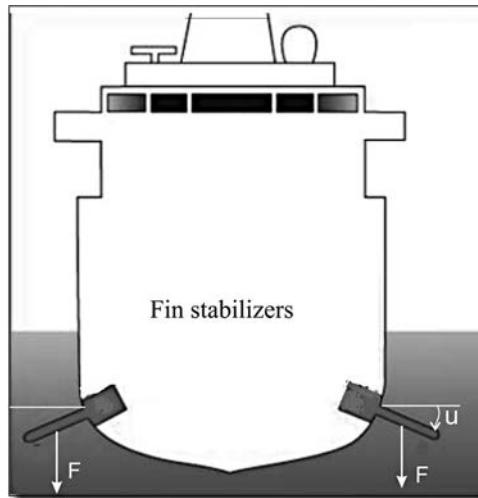


Figure 12.26 Fin stabilized ship where the vertical force $F = ku$ is proportional to the angle u for small deflections.

For marine craft the most common actuators are:

- **Main propellers:** The main propellers of the craft are mounted aft of the hull, usually in conjunction with rudders. They produce the necessary force F_x in the x direction needed for transit.
- **Tunnel thrusters:** These are transverse thrusters going through the hull of the craft. The propeller unit is mounted inside a transverse tube and produces a force F_y in the y direction. Tunnel thrusters are only effective at low speeds, which limits their use to low-speed maneuvering and stationkeeping.
- **Azimuth thrusters:** Thruster units that can be rotated an angle α about the z axis and produce two force components (F_x, F_y) in the horizontal plane are usually referred to as azimuth thrusters. They are usually mounted under the hull of the craft and the most sophisticated units are retractable. Azimuth thrusters are frequently used in DP systems since they can produce forces in different directions. Hence, this becomes an overactuated control problem that can be optimized with respect to power and possible failure situations.
- **Aft rudders:** Rudders are the primary steering device for conventional marine craft. They are located aft of the craft and the rudder force F_y will be a function of the rudder deflection (the drag force in the x direction is usually neglected in the control analysis). A rudder force in the y direction will produce a yaw moment that can be used for steering control.
- **Stabilizing fins:** Stabilizing fins are used for the damping of vertical vibrations and roll motions (see Figure 12.26). They produce a force F_z in the z directions that is a function of the fin deflection. For small angles this relationship is linear. Fin stabilizers can be retractable, allowing for selective use in bad weather. The lift forces are small at low speed so the most effective operating condition is in transit.
- **Control surfaces:** Control surfaces can be mounted at different locations to produce lift and drag forces. For underwater vehicles these could be fins for diving, rolling and pitching and rudders for steering.
- **Water jets:** Water jets are an alternative to main propellers aft of the ship. They are usually used for high-speed craft.

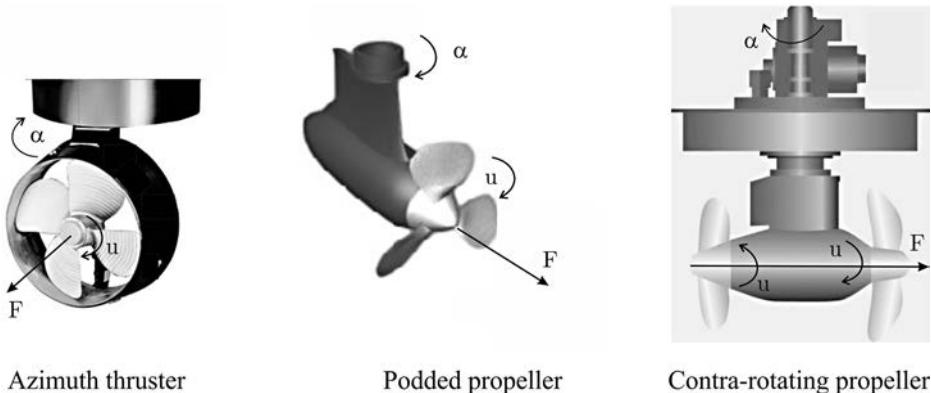


Figure 12.27 Propellers that can be rotated an angle α to produce a force F in an arbitrary direction.

The forces and moments in 6 DOF corresponding to the force vector $\mathbf{f} = [F_x, F_y, F_z]^\top$ can be written (see Table 12.3)

$$\boldsymbol{\tau} = \begin{bmatrix} \mathbf{f} \\ \mathbf{r} \times \mathbf{f} \end{bmatrix} = \begin{bmatrix} F_x \\ F_y \\ F_z \\ F_z l_y - F_y l_z \\ F_x l_z - F_z l_x \\ F_y l_x - F_x l_y \end{bmatrix} \xrightarrow{4 \text{ DOF}} \boldsymbol{\tau} = \begin{bmatrix} F_x \\ F_y \\ F_z \\ F_z l_y - F_y l_z \\ F_y l_x - F_x l_y \\ F_x l_z - F_z l_x \end{bmatrix} \quad (12.226)$$

where $\mathbf{r} = [l_x, l_y, l_z]^\top$ are the moment arms. For rotatable (azimuth) thrusters the control force F will be a function of the rotation angle α and propeller revolution u (see Figure 12.27). Consequently, an azimuth thruster in the horizontal plane will have two force components, $F_x = F \cos(\alpha)$ and $F_y = F \sin(\alpha)$, while the main propeller aft of the ship only produces a longitudinal force $F_x = F$ (see Table 12.3).

Thrust Configuration and Force Coefficient Matrices

The control forces and moments $\mathbf{f} = [u_1, \dots, u_n]^\top$ are conveniently expressed as

$$\mathbf{f} = \mathbf{K}\mathbf{u} \quad (12.227)$$

where $\mathbf{u} = [u_1, \dots, u_r]^\top$ is a vector of control inputs and $\mathbf{K} \in \mathbb{R}^{r \times r}$ is a diagonal force coefficient matrix given by

$$\mathbf{K} = \text{diag}\{K_1, \dots, K_r\}, \quad \mathbf{K}^{-1} = \text{diag} \left\{ \frac{1}{K_1}, \dots, \frac{1}{K_r} \right\} \quad (12.228)$$

The actuator forces and moments relate to the control forces and moments by

$$\begin{aligned}\tau &= \mathbf{T}(\alpha) f \\ &= \mathbf{T}(\alpha) \mathbf{K} u\end{aligned}\quad (12.229)$$

where $\alpha = [\alpha_1, \dots, \alpha_p]^\top \in \mathbb{R}^p$ is a vector of azimuth angles and $\mathbf{T}(\alpha) \in \mathbb{R}^{n \times r}$ is the thrust configuration matrix. For a marine craft equipped with r actuators for operation in n DOFs, the thrust configuration matrix describes the geometry or locations of the actuators.

Thrust Configuration Matrix for Nonrotatable Thrusters: The trivial case refers to a marine craft equipped with nonrotatable thrusters such that

$$\mathbf{T} = [t_1, \dots, t_r] = \text{constant}$$

The thrust configuration matrix is defined in terms of a set of column vectors $t_i \in \mathbb{R}^n$. In 4 DOF (*surge*, *sway*, *roll* and *yaw*) the column vectors for some standard actuators are

$$t_i = \underbrace{\begin{bmatrix} 1 \\ 0 \\ 0 \\ -l_{y_i} \end{bmatrix}}_{\text{main propeller}}, \quad t_i = \underbrace{\begin{bmatrix} 0 \\ 1 \\ -l_{z_i} \\ l_{x_i} \end{bmatrix}}_{\text{tunnel thruster and aft rudder}}, \quad t_i = \underbrace{\begin{bmatrix} 0 \\ 0 \\ l_{y_i} \\ 0 \end{bmatrix}}_{\text{stabilizing fin}}$$

An example using this representation is found in Section 13.1.5 discussing fin and rudder control systems.

Thrust Configuration Matrix for Rotatable Thrusters: For marine craft equipped with azimuth thrusters in combination with nonrotatable thrusters we write:

$$\mathbf{T}(\alpha) = [t_1, \dots, t_r] \quad (12.230)$$

where $\alpha = [\alpha_1, \dots, \alpha_p]^\top \in \mathbb{R}^p$ is a vector of azimuth angles. The thrust configuration matrix is defined in terms of a set of column vectors $t_i \in \mathbb{R}^n$. In 4 DOF (*surge*, *sway* *roll* and *yaw*) the column vectors take the following form:

$$t_i = \underbrace{\begin{bmatrix} \cos(\alpha_i) \\ \sin(\alpha_i) \\ -l_{z_i} \sin(\alpha_i) \\ l_{x_i} \sin(\alpha_i) - l_{y_i} \cos(\alpha_i) \end{bmatrix}}_{\text{azimuth thruster}}, \quad t_i = \underbrace{\begin{bmatrix} 1 \\ 0 \\ 0 \\ -l_{y_i} \end{bmatrix}}_{\text{main propeller}}, \quad t_i = \underbrace{\begin{bmatrix} 0 \\ 1 \\ -l_{z_i} \\ l_{x_i} \end{bmatrix}}_{\text{tunnel thruster and aft rudder}}, \quad t_i = \underbrace{\begin{bmatrix} 0 \\ 0 \\ l_{y_i} \\ 0 \end{bmatrix}}_{\text{stabilizing fin}}$$

An example using this representation is found in Section 12.3.5 discussing dynamic positioning systems.

Extended Thrust Configuration Matrix for Rotatable Actuators: When solving the control allocation optimization problem an alternative representation to (12.230) is the extended thrust configuration matrix. Equation (12.230) is nonlinear in the controls α . This implies that a nonlinear optimization problem must be solved. In order to avoid this, the rotatable thrusters can be treated as two forces. Consider a rotatable thruster in the horizontal plane (the same methodology can be used for thrusters that can be rotated in the vertical plane):

$$\begin{aligned} F_{x_i} &= F_i \cos(\alpha_i) \\ &= K_i u_i \cos(\alpha_i) \end{aligned} \quad (12.231)$$

$$\begin{aligned} F_{y_i} &= F_i \sin(\alpha_i) \\ &= K_i u_i \sin(\alpha_i) \end{aligned} \quad (12.232)$$

Next, the extended force vector is defined according to

$$\mathbf{f}_e := \mathbf{K}_e \mathbf{u}_e \quad (12.233)$$

such that

$$\boldsymbol{\tau} = \mathbf{T}_e \mathbf{K}_e \mathbf{u}_e \quad (12.234)$$

where \mathbf{T}_e and \mathbf{K}_e are the extended thrust configuration and coefficient matrices, respectively, and \mathbf{u}_e is a vector of extended control inputs where the azimuth controls are defined as

$$u_{ix} := u_i \cos(\alpha_i) \quad (12.235)$$

$$u_{iy} := u_i \sin(\alpha_i) \quad (12.236)$$

This approach was used by Sørdalen (1997b). The following example illustrates how this model can be established for an underwater vehicle equipped with two main propellers and two azimuth thrusters in the horizontal plane (Fossen *et al.*, 2009).

Example 12.8 (Thrust Configuration Matrix for an Underwater Vehicle)

The forces and moment X , Y and N in surge, sway and yaw, respectively, for the AUV thruster configuration shown in Figure 12.28 satisfy

$$\boldsymbol{\tau} = \mathbf{T}(\alpha) \mathbf{K} \mathbf{u} \quad (12.237)$$

¶

$$\begin{bmatrix} X \\ Y \\ N \end{bmatrix} = \begin{bmatrix} \cos(\alpha_1) & \cos(\alpha_2) & 1 & 1 \\ \sin(\alpha_1) & \sin(\alpha_2) & 0 & 0 \\ l_{x1} \sin(\alpha_1) & l_{x2} \sin(\alpha_2) & -l_{y3} & -l_{y4} \end{bmatrix} \cdot \begin{bmatrix} K_1 & 0 & 0 & 0 \\ 0 & K_2 & 0 & 0 \\ 0 & 0 & K_3 & 0 \\ 0 & 0 & 0 & K_4 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix} \quad (12.238)$$

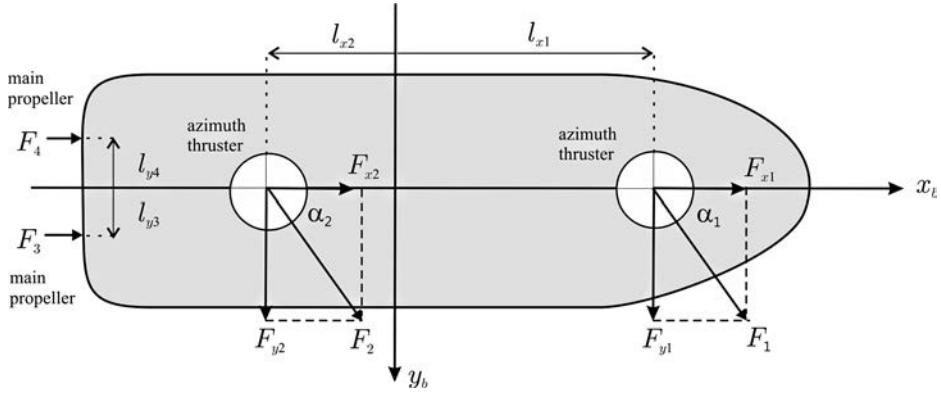


Figure 12.28 AUV equipped with two azimuth thrusters (forces F_1 and F_2) and two main propellers (forces F_3 and F_4). The azimuth forces are decomposed along the x and y axes.

The extended thrust vector \mathbf{u}_e corresponding to (12.234) and (12.235)–(12.236) satisfies

$$\boldsymbol{\tau}_e = \mathbf{T}_e \mathbf{K}_e \mathbf{u}_e \quad (12.239)$$

iff

$$\begin{bmatrix} X \\ Y \\ N \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & l_{x1} & 0 & l_{x2} & -l_{y3} & -l_{y4} \end{bmatrix} \cdot \begin{bmatrix} K_1 & 0 & 0 & 0 & 0 & 0 \\ 0 & K_1 & 0 & 0 & 0 & 0 \\ 0 & 0 & K_2 & 0 & 0 & 0 \\ 0 & 0 & 0 & K_2 & 0 & 0 \\ 0 & 0 & 0 & 0 & K_3 & 0 \\ 0 & 0 & 0 & 0 & 0 & K_4 \end{bmatrix} \begin{bmatrix} u_{1x} \\ u_{1y} \\ u_{2x} \\ u_{2y} \\ u_3 \\ u_4 \end{bmatrix} \quad (12.240)$$

Notice that $\mathbf{T}_e = \text{constant}$ while $\mathbf{T}(\alpha)$ depends on α . This means that the extended control input vector \mathbf{u}_e can be solved directly from (12.234), for instance using the pseudo-inverse approach presented in Section 12.3.2. This is not the case for (12.230), which represents a nonlinear optimization problem. If \mathbf{u}_e is computed using the pseudo-inverse, the azimuth control can be derived from the extended control vector elements by mapping the pairs (u_{1x}, u_{1y}) and (u_{2x}, u_{2y}) according to

$$u_1 = \sqrt{u_{1x}^2 + u_{1y}^2}, \quad \alpha_1 = \text{atan2}(u_{1y}, u_{1x}) \quad (12.241)$$

$$u_2 = \sqrt{u_{2x}^2 + u_{2y}^2}, \quad \alpha_2 = \text{atan2}(u_{2y}, u_{2x}) \quad (12.242)$$

The last two controls u_3 and u_4 are elements five and six in \mathbf{u}_e .

12.3.2 Unconstrained Control Allocation for Nonrotatable Actuators

The simplest allocation problem is the one where all control forces are produced by thrusters in fixed directions alone or in combination with rudders and control surfaces. This implies that

$$\alpha = \alpha_0 = \text{constant}, \quad \mathbf{T} = \mathbf{T}(\alpha_0) \quad (12.243)$$

It will be assumed that the allocation problem is *unconstrained*; that is there are no bounds on the vector elements f_i , α_i and u_i , and their time derivatives. Saturating control and constrained control allocation are discussed in Sections 12.3.3–12.3.4.

For marine craft where the configuration matrix \mathbf{T} is square or nonsquare ($r \geq n$), that is there are equal or more control inputs than controllable DOFs, it is possible to find an “optimal” distribution of control forces \mathbf{f} for each DOF by using an explicit method. Consider the unconstrained least-squares (LS) optimization problem (Fossen and Sagatun, 1991):

$$\begin{aligned} J &= \min_{\mathbf{f}} \{ \mathbf{f}^\top \mathbf{W} \mathbf{f} \} \\ \text{subject to: } \boldsymbol{\tau} - \mathbf{T} \mathbf{f} &= \mathbf{0} \end{aligned} \quad (12.244)$$

Here \mathbf{W} is a positive definite matrix, usually diagonal, weighting the control forces. For marine craft that have both control surfaces and propellers, the elements in \mathbf{W} should be selected so that using the control surfaces are considerably less expensive than using the propellers.

Explicit Solution to the LS Optimization Problem using Lagrange Multipliers

Consider the Lagrangian (Fossen, 1994)

$$L(\mathbf{f}, \boldsymbol{\lambda}) = \mathbf{f}^\top \mathbf{W} \mathbf{f} + \boldsymbol{\lambda}^\top (\boldsymbol{\tau} - \mathbf{T} \mathbf{f}) \quad (12.245)$$

where $\boldsymbol{\lambda} \in \mathbb{R}^r$ is a vector of Lagrange multipliers. Consequently, differentiating the Lagrangian L with respect to \mathbf{f} yields

$$\frac{\partial L}{\partial \mathbf{f}} = 2 \mathbf{W} \mathbf{f} - \mathbf{T}^\top \boldsymbol{\lambda} = \mathbf{0} \quad (12.246)$$

\Downarrow

$$\mathbf{f} = \frac{1}{2} \mathbf{W}^{-1} \mathbf{T}^\top \boldsymbol{\lambda} \quad (12.247)$$

Next, assume that $\mathbf{T} \mathbf{W}^{-1} \mathbf{T}^\top$ is nonsingular such that

$$\boldsymbol{\tau} = \mathbf{T} \mathbf{f} = \frac{1}{2} \mathbf{T} \mathbf{W}^{-1} \mathbf{T}^\top \boldsymbol{\lambda} \quad (12.248)$$

\Downarrow

$$\boldsymbol{\lambda} = 2(\mathbf{T} \mathbf{W}^{-1} \mathbf{T}^\top)^{-1} \boldsymbol{\tau} \quad (12.249)$$

Substituting the Lagrange multipliers $\lambda = 2(\mathbf{T}\mathbf{W}^{-1}\mathbf{T}^\top)^{-1}\boldsymbol{\tau}$ into (12.247) yields

$$\mathbf{f} = \underbrace{\mathbf{W}^{-1}\mathbf{T}^\top(\mathbf{T}\mathbf{W}^{-1}\mathbf{T}^\top)^{-1}}_{\mathbf{T}_w^\dagger}\boldsymbol{\tau} \quad (12.250)$$

where the matrix

$$\mathbf{T}_w^\dagger = \mathbf{W}^{-1}\mathbf{T}^\top(\mathbf{T}\mathbf{W}^{-1}\mathbf{T}^\top)^{-1} \quad (12.251)$$

is recognized as the *generalized inverse*. For the case $\mathbf{W} = \mathbf{I}$, that is equally weighted control forces, (12.251) reduces to the right *Moore–Penrose pseudo-inverse*

$$\mathbf{T}^\dagger = \mathbf{T}^\top(\mathbf{T}\mathbf{T}^\top)^{-1} \quad (12.252)$$

Since

$$\mathbf{f} = \mathbf{T}_w^\dagger\boldsymbol{\tau} \quad (12.253)$$

the control input vector \mathbf{u} can be computed from (12.230) as

$$\mathbf{u} = \mathbf{K}^{-1}\mathbf{T}_w^\dagger\boldsymbol{\tau} \quad (12.254)$$

Notice that this solution is valid for all $\boldsymbol{\alpha}_0$ but not optimal with respect to a time-varying $\boldsymbol{\alpha}_0$ (only \mathbf{f}). Optimality with respect to $\boldsymbol{\alpha}$ in addition to (12.247) is discussed in Section 12.3.4.

Matlab

The generalized inverse for the case $\mathbf{T} = \mathbf{T}(\boldsymbol{\alpha}_0) = \text{constant}$ is implemented in the Matlab MSS toolbox as

```
u=ualloc(K,T,W,tau)
```

12.3.3 Constrained Control Allocation for Nonrotatable Actuators

In industrial systems it is important to minimize the power consumption by taking advantage of the additional control forces in an overactuated control problem. From a critical point of view concerning safety it is also important to take into account actuator limitations such as saturation, wear and tear as well as other constraints. In general this leads to a *constrained optimization problem*.

Explicit Solution using Piecewise Linear Functions

An explicit solution approach for parametric quadratic programming has been developed by Tøndel *et al.* (2003a) while applications to marine craft are presented by Johansen *et al.* (2005). In this work the constrained optimization problem is formulated as

$$\begin{aligned} J &= \min_{\mathbf{f}, \mathbf{s}, \bar{\mathbf{f}}} \{ \mathbf{f}^\top \mathbf{W} \mathbf{f} + \mathbf{s}^\top \mathbf{Q} \mathbf{s} + \beta \bar{\mathbf{f}} \} \\ \text{subject to:} \\ \mathbf{T}\mathbf{f} &= \boldsymbol{\tau} + \mathbf{s} \\ \mathbf{f}_{\min} &\leq \mathbf{f} \leq \mathbf{f}_{\max} \\ -\bar{\mathbf{f}} &\leq f_1, f_2, \dots, f_r \leq \bar{\mathbf{f}} \end{aligned} \quad (12.255)$$

where $\mathbf{s} \in \mathbb{R}^n$ is a vector of *slack variables*. The first term of the criterion corresponds to the LS criterion (12.244), while the third term is introduced to minimize the largest force $\hat{\bar{f}} = \max_i |f_i|$ among the actuators. The constant $\beta \geq 0$ controls the relative weighting of the two criteria. This formulation ensures that the constraints $f_i^{\min} \leq f_i \leq f_i^{\max}$ ($i = 1, \dots, r$) are satisfied, if necessary by allowing the resulting generalized force $\mathbf{T}\mathbf{f}$ to deviate from its specification $\boldsymbol{\tau}$. To achieve accurate generalized force, the slack variable should be close to zero. This is obtained by choosing the weighting matrix $\mathbf{Q} \gg \mathbf{W} > 0$. Moreover, saturation is handled in an optimal manner by minimizing the combined criterion (12.255).

Letting

$$\mathbf{z} = [\mathbf{f}^\top, \mathbf{s}^\top, \bar{\mathbf{f}}]^\top \in \mathbb{R}^{r+n+1} \quad (12.256)$$

and

$$\mathbf{p} = [\boldsymbol{\tau}^\top, \mathbf{f}_{\min}^\top, \mathbf{f}_{\max}^\top, \beta]^\top \in \mathbb{R}^{n+2r+1} \quad (12.257)$$

denotes the parameter vector, it is straightforward to see that the optimization problem (12.255) can be reformulated as a QP problem:

$$\begin{aligned} J &= \min_{\mathbf{z}} \{ \mathbf{z}^\top \Phi \mathbf{z} + \mathbf{z}^\top \mathbf{R} \mathbf{p} \} \\ \text{subject to:} \\ \mathbf{A}_1 \mathbf{z} &= \mathbf{C}_1 \mathbf{p} \\ \mathbf{A}_2 \mathbf{z} &\leq \mathbf{C}_2 \mathbf{p} \end{aligned} \quad (12.258)$$

where

$$\Phi = \begin{bmatrix} \mathbf{W} & \mathbf{0}_{r \times n} & \mathbf{0}_{r \times 1} \\ \mathbf{0}_{n \times r} & \mathbf{Q} & \mathbf{0}_{n \times 1} \\ \mathbf{0}_{1 \times r} & \mathbf{0}_{1 \times n} & 0 \end{bmatrix}, \quad \mathbf{R} = \left[\mathbf{0}_{(r+n+1) \times (n+2r)} \begin{bmatrix} \mathbf{0}_{(r+n) \times 1} \\ 1 \end{bmatrix} \right]$$

$$\mathbf{A}_1 = \begin{bmatrix} \mathbf{T} - \mathbf{I}_{n \times n} & \mathbf{0}_{n \times 1} \end{bmatrix}, \quad \mathbf{C}_1 = \begin{bmatrix} \mathbf{I}_{n \times n} & \mathbf{0}_{n \times (2r+1)} \end{bmatrix}$$

$$\mathbf{A}_2 = \begin{bmatrix} -\mathbf{I}_{r \times r} & \mathbf{0}_{r \times n} & \mathbf{0}_{r \times 1} \\ \mathbf{I}_{r \times r} & \mathbf{0}_{r \times n} & \mathbf{0}_{r \times 1} \\ \vdots & \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} & \vdots \\ \mathbf{I}_{r \times r} & \mathbf{0}_{r \times n} & \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} \\ \mathbf{I}_{r \times r} & \mathbf{0}_{r \times n} - & \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} \end{bmatrix}, \quad \mathbf{C}_2 = \begin{bmatrix} \mathbf{0}_{r \times n} & -\mathbf{I}_{r \times r} & \mathbf{0}_{r \times r} & \mathbf{0}_{r \times 1} \\ \mathbf{0}_{r \times n} & \mathbf{0}_{r \times r} & \mathbf{I}_{r \times r} & \mathbf{0}_{r \times 1} \\ \mathbf{0}_{r \times n} & \mathbf{0}_{r \times r} & \mathbf{0}_{r \times r} & \mathbf{0}_{r \times 1} \\ \mathbf{0}_{r \times n} & \mathbf{0}_{r \times r} & \mathbf{0}_{r \times r} & \mathbf{0}_{r \times 1} \end{bmatrix}$$

Since $\mathbf{W} > 0$ and $\mathbf{Q} > 0$ this is a convex quadratic program in \mathbf{z} parametrized by \mathbf{p} . Convexity guarantees that a global solution can be found. The optimal solution $\mathbf{z}^*(\mathbf{p})$ to this problem is a continuous piecewise linear function $\mathbf{z}^*(\mathbf{p})$ defined on any subset

$$\mathbf{p}_{\min} \leq \mathbf{p} \leq \mathbf{p}_{\max} \quad (12.259)$$

of the parameter space. Moreover, an exact representation of this piecewise linear function can be computed offline using multiparametric QP (mp-QP) algorithms (Tøndel *et al.*, 2003b) or the *Matlab Multi-Parametric Toolbox* (MPT) by Kvasnica *et al.* (2004). Consequently, it is not necessary to solve the QP (12.255) in real time for the current value of τ and the parameters f_{\min} , f_{\max} and β if they are allowed to vary. In fact, it suffices to evaluate the known piecewise linear function $\mathbf{z}^*(\mathbf{p})$ as a function of the given parameter vector \mathbf{p} , which can be done efficiently with a small amount of computations. For details of the implementation aspects of the mp-QP algorithm see Johansen *et al.* (2004) and references therein. An online control allocation algorithm is presented in Tøndel *et al.* (2003a).

Explicit Solution for Varying α using Piecewise Linear Functions

An extension of the mp-QP algorithm to marine craft equipped with azimuth thrusters and rudders has been given by Tøndel *et al.* (2003a). A propeller with a rudder can produce a thrust vector within a range of directions and magnitudes in the horizontal plane for low-speed maneuvering and stationkeeping. The set of attainable thrust vectors is nonconvex because significant lift can be produced by the rudder only with forward thrust. The attainable thrust region can, however, be decomposed into a finite union of convex polyhedral sets. A similar decomposition can be made for azimuth thrusters including forbidden sectors. Hence, this can be formulated as a mixed-integer-like convex QP problem, and by using, arbitrarily, number, of rudders as well as thrusters, other propulsion devices can be handled. Actuator rate and position constraints are also taken into account. Using a mp-QP software, an explicit piecewise linear representation of the least-squares optimal control allocation law can be precomputed. The method has been tested on a scale model of a supply vessel by Tøndel *et al.* (2003a) and a scale model of a floating platform by Spjøtvold (2008).

Explicit Solutions based on Minimum Norm and Null-Space Methods

In flight and aerospace control systems, the problems of control allocation and saturating control have been addressed by Durham (1993, 1994a, 1994b). Durham also discusses an explicit solution to avoid saturation, referred to as the “direct method”. By noticing that there are infinite combinations of admissible controls that generate control forces on the boundary of the closed subset of attainable controls, the “direct method” calculates admissible controls in the interior of the attainable forces as scaled-down versions of the unique solutions for force demands. Unfortunately it is not possible to minimize the norm of the control forces on the boundary or some other constraint since the solutions on the boundary are unique. The computational complexity of the algorithm is proportional to the square of the number of controls, which can be problematic in real-time applications.

In Bordignon and Durham (1995) the null-space interaction method is used to minimize the norm of the control vector, when possible, and still access the attainable forces to overcome the drawbacks of the “direct method”. This method is also explicit but much more computationally intensive. For instance, 20 independent controls imply that up to 3.4 billion points have to be checked at each sample. In Durham (1999) a computationally simple and efficient method to obtain near-optimal solutions is described. The method is based on prior knowledge of the controls’ effectiveness and limits such that precalculation of several generalized inverses can be done.

Iterative Solutions

An alternative to the explicit solution could be to use an iterative solution to solve the QP problem. The m-file function `quadprog.m` in the Matlab optimization toolbox can be used for computer simulations, while a stand alone compiled QP solver must be implemented in a real-time application. The drawback with the iterative solution is that several iterations may have to be performed at each sample in order to find the optimal solution. An advantage of the iterative approach is that there is more flexibility for online reconfiguration, as, for example, a change in \mathbf{W} may require that the explicit solutions are recalculated. Computational complexity is also greatly reduced by a “warm start”; that is the numerical solver is initialized with the solution of the optimization problem computed at the previous sample.

12.3.4 Constrained Control Allocation for Azimuth Thrusters

The control allocation problem for marine craft equipped with azimuth thrusters is in general a *nonconvex* optimization problem that is hard to solve. The primary constraint is

$$\boldsymbol{\tau} = \mathbf{T}(\boldsymbol{\alpha}) f \quad (12.260)$$

where $\boldsymbol{\alpha} \in \mathbb{R}^p$ denotes the azimuth angles. The azimuth angles must be computed at each sample together with the control inputs $\mathbf{u} \in \mathbb{R}^r$ which are subject to both amplitude and rate saturations. In addition, azimuth thrusters can only operate in feasible sectors $\alpha_{i,\min} \leq \alpha_i \leq \alpha_{i,\max}$ at a limiting turning rate $\dot{\alpha}_i$. Another problem is that the inverse

$$\mathbf{T}_w^\dagger(\boldsymbol{\alpha}) = \mathbf{W}^{-1} \mathbf{T}^\top(\boldsymbol{\alpha}) [\mathbf{T}(\boldsymbol{\alpha}) \mathbf{W}^{-1} \mathbf{T}^\top(\boldsymbol{\alpha})]^{-1} \quad (12.261)$$

can be singular for certain $\boldsymbol{\alpha}$ values. The consequence of such a singularity is that no force is produced in certain directions. This may greatly reduce dynamic performance and maneuverability as the azimuth angles can be changed only slowly. This suggests that the following criterion should be minimized (Johansen *et al.*, 2004):

$$J = \min_{f, \alpha, s} \left\{ \sum_{i=1}^r \bar{P}_i |f_i|^{3/2} + s^\top Q s + (\alpha - \alpha_0)^\top \Omega (\alpha - \alpha_0) \right. \\ \left. + \frac{\varrho}{\varepsilon + \det(T(\alpha) W^{-1} T^\top(\alpha))} \right\} \quad (12.262)$$

subject to:

$$\begin{aligned} T(\alpha) f &= \tau + s \\ f_{\min} &\leq f \leq f_{\max} \\ \alpha_{\min} &\leq \alpha \leq \alpha_{\max} \\ \Delta\alpha_{\min} &\leq \alpha - \alpha_0 \leq \Delta\alpha_{\max} \end{aligned}$$

where

- $\sum_{i=1}^r \bar{P}_i |f_i|^{3/2}$ represents power consumption where $\bar{P}_i > 0$ ($i = 1, \dots, r$) are positive weights.
- $s^\top Q s$ penalizes the error s between the commanded and achieved generalized force. This is necessary in order to guarantee that the optimization problem has a feasible solution for any τ and α_0 . The weight $Q > 0$ is chosen to be large enough so that the optimal solution is $s \approx \mathbf{0}$ whenever possible.
- $f_{\min} \leq f \leq f_{\max}$ is used to limit the use of force (saturation handling).
- $\alpha_{\min} \leq \alpha \leq \alpha_{\max}$ denotes the feasible sectors of the azimuth angles.
- $\Delta\alpha_{\min} \leq \alpha - \alpha_0 \leq \Delta\alpha_{\max}$ ensures that the azimuth angles do not move too much within one sample, taking α_0 equal to the angles at the previous sample. This is equivalent to limiting $\dot{\alpha}$, that is the turning rate of the thrusters.
- The term

$$\frac{\varrho}{\varepsilon + \det(T(\alpha) W^{-1} T^\top(\alpha))}$$

is introduced to avoid singular configurations given by $\det(T(\alpha) W^{-1} T^\top(\alpha)) = 0$. To avoid division by zero, $\varepsilon > 0$ is chosen as a small number, while $\varrho > 0$ is scalar weight. A large ϱ ensures high maneuverability at the cost of higher power consumption and vice versa.

The optimization problem (12.262) is a nonconvex nonlinear program and requires a significant amount of computations at each sample (Nocedal and Wright, 1999). The nonlinear program is solved by using iterations as shown in Figure 12.29. The following two implementation strategies are attractive alternatives to nonlinear program efforts.

Iterative Solutions using Quadratic Programming

The problem (12.262) can be locally approximated with a *convex* QP problem by assuming that:

1. The power consumption can be approximated by a quadratic term in f near the last force f_0 such that $f = f_0 + \Delta f$.
2. The singularity avoidance penalty can be approximated by a linear term linearized about the last azimuth angle α_0 such that $\alpha = \alpha_0 + \Delta\alpha$.

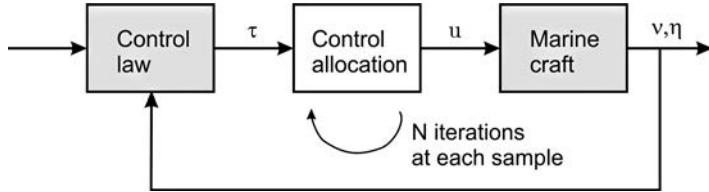


Figure 12.29 Control allocation using an iterative solution.

The resulting QP criterion is (Johansen *et al.*, 2004)

$$\begin{aligned}
 J = \min_{\Delta f, \Delta \alpha, s} & \left\{ (f_0 + \Delta f)^\top P (f_0 + \Delta f) + s^\top Q s + \Delta \alpha^\top \Omega \Delta \alpha \right. \\
 & \left. + \frac{\partial}{\partial \alpha} \left(\frac{\varrho}{\varepsilon + \det(\mathbf{T}(\alpha) \mathbf{W}^{-1} \mathbf{T}^\top(\alpha))} \right) \Big|_{\alpha_0} \Delta \alpha \right\} \quad (12.263)
 \end{aligned}$$

subject to:

$$\begin{aligned}
 s + \mathbf{T}(\alpha_0) \Delta f + \frac{\partial}{\partial \alpha} (\mathbf{T}(\alpha_0) f)|_{\alpha_0, f_0} \Delta \alpha &= \tau - \mathbf{T}(\alpha_0) f_0 \\
 f_{\min} - f_0 \leq \Delta f &\leq f_{\max} - f_0 \\
 \alpha_{\min} - \alpha_0 \leq \Delta \alpha &\leq \alpha_{\max} - \alpha_0 \\
 \Delta \alpha_{\min} \leq \Delta \alpha &\leq \Delta \alpha_{\max}
 \end{aligned}$$

The convex QP problem (12.263) can be solved by using standard software for numerical optimization, for instance the m-file function `quadprog.m` in the Matlab optimization toolbox.

Iterative Solutions using Linear Programming

Linear approximations to the thrust allocation problem have been discussed by Webster and Sousa (1999) and Lindfors (1993). In Lindfors (1993) the azimuth thrust constraints

$$|f_i| = \sqrt{[f_i \cos(\alpha_i)]^2 + [f_i \sin(\alpha_i)]^2} \leq f_i^{\max} \quad (12.264)$$

are represented as circles in the $(f_i \cos \alpha_i, f_i \sin \alpha_i)$ plane. The nonlinear program is transformed to a linear programming (LP) problem by approximating the azimuth thrust constraints by straight lines forming a polygon. If eight lines are used to approximate the circles (octagons), the worst case errors will be less than $\pm 4.0\%$. The criterion to be minimized is a linear combination of $\|f\|$, that is magnitude of force in the x and y directions, weighted against the magnitudes $|\sqrt{[f_i \cos(\alpha_i)]^2 + [f_i \sin(\alpha_i)]^2}|$ representing azimuth thrust. Hence, singularities and azimuth rate limitations are not weighted in the cost function. If these are important, the QP formulation should be used.

Explicit Solution using the Singular Value Decomposition and Filtering Techniques

An alternative method to solve the constrained control allocation problem is to use the singular value decomposition (SVD) and a filtering scheme to control the azimuth directions such that they are aligned with the direction where most force is required, paying attention to singularities (Sørdalen, 1997b).

Results from sea trials have been presented in Sør dalen (1997a). A similar technique using the damped least-squares algorithm has been reported in Berge and Fossen (1997), where the results are documented by controlling a scale model of a supply vessel equipped with four azimuth thrusters.

12.3.5 Case Study: DP Control Allocation System

Most DP ships use thrusters to maintain their position and heading. Both *fixed pitch* (FP) and *controllable pitch* (CP) propellers are available for this purpose.

Fixed-Speed CP and Variable-Speed FP Propellers

The thrust F from a *variable-speed FP* propeller can be modeled as

$$F(n) = Kn |n| \quad (\text{or } F(n) = Kn) \quad (12.265)$$

where $K = \text{constant}$ is the thrust coefficient and n is the propeller revolutions per minute (rpm). Some propellers show linear behavior in n while others are quadratic. Even combinations of the linear and quadratic behavior are observed in practice.

CP propellers are screw blade propellers where the blades can be turned under the control of a hydraulic servo. This introduces a second control variable, pitch p , which is used to obtain the desired thrust F for different propeller revolutions n . If P is the “traveled distance per revolution” and D is the propeller diameter then $p = P/D$ represents the pitch ratio.

The thrust from a *fixed-speed CP* propeller can be approximated by

$$F(n, p) = K(n)|(p - p_0)|(p - p_0) \quad (\text{or } F(n, p) = K(n)(p - p_0)) \quad (12.266)$$

where the force coefficient $K(n)$ now depends on the propeller revolution. Again, thrust is quadratic, alternatively linear, in $p - p_0$ or combinations of both. The pitch offset is denoted as p_0 . For DP ships using fixed-speed CP propellers it is common to operate at one or two fixed propeller revolutions such that only p is used for active control by the DP system; see Example 12.9.

For ships in transit a constant demand for thrust and power suggests that a fixed-speed CP propeller should be used while low-speed applications such as DP operations require little thrust in good weather, suggesting that a variable-speed FP propeller might be advantageous (see Figure 12.30). Notice that the fixed-speed CP propeller also requires power at zero thrust.

Example 12.9 (Experimental Thrust Characteristics)

The supply vessel in Fossen et al. (1996) is equipped with a main propeller and tunnel thrusters. The measured thrust is shown as asterisks in Figure 12.31 while the solid lines are least-square fits to the quadratic thrust function (12.266). The main propeller operated at $n = 122$ rpm and $n = 160$ rpm, while the tunnel thruster ran at $n = 236$ rpm resulting in

$$\begin{aligned} \text{Main propeller} \quad F(122, p) &= 370 |p| p & F(236, p) &= 137 |p| p \\ \text{Tunnel thruster} \quad F(160, p) &= 655 |p| p \end{aligned}$$

Actuator Configuration and Thrust Coefficient Matrices

Recall from Section 12.3 that the forces and moment $\tau \in \mathbb{R}^3$ (*surge*, *sway* and *yaw*) can be written

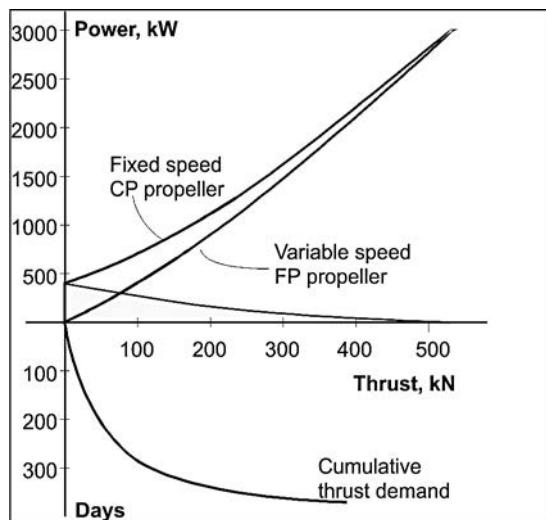


Figure 12.30 Power consumption of fixed-speed CP and variable speed FP propellers.

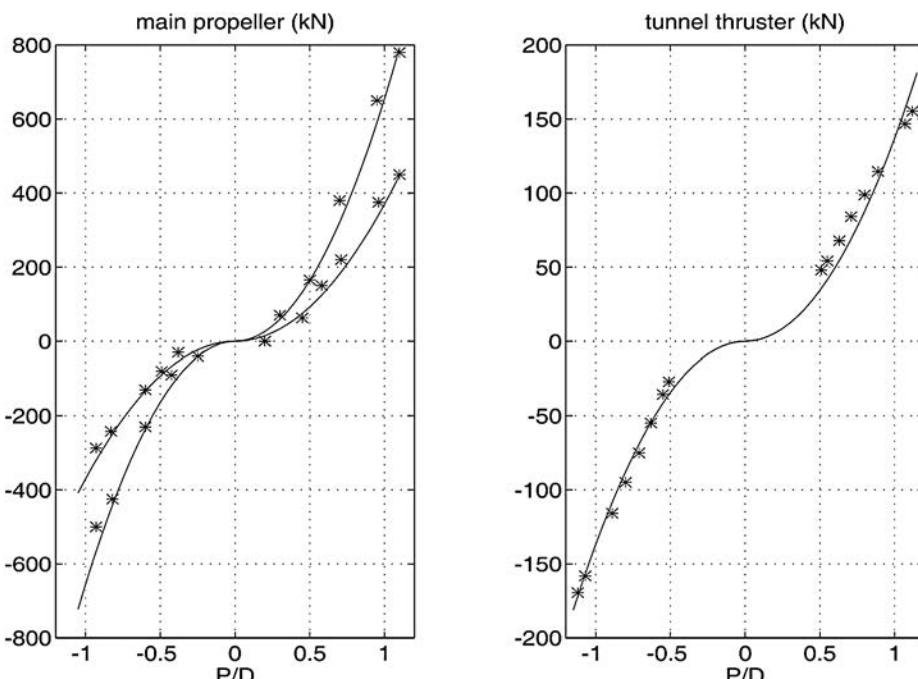


Figure 12.31 Thrust $F(n, p) = K(n)p |p|$ versus pitch p for a main propeller (left-hand plot) and a tunnel thruster (right-hand plot). The asterisks are experimental measured values and the solid lines are least-square fits to a quadratic model.

$$\tau = \mathbf{T}(\alpha) f \quad (12.267)$$

$$f = \mathbf{K} u \quad (12.268)$$

where $f \in \mathbb{R}^r$ (r = number of thrusters) is the thrust force vector and $u \in \mathbb{R}^r$ is a DP control variable given by

$$\begin{aligned} \text{CP: } \mathbf{u} &= [|p_1|p_1, |p_2|p_2, \dots, |p_r|p_r]^\top, & (\text{or } \mathbf{u} = [p_1, p_2, \dots, p_r]^\top) \\ \text{FP: } \mathbf{u} &= [|n_1|n_1, |n_2|n_2, \dots, |n_r|n_r]^\top, & (\text{or } \mathbf{u} = [n_1, n_2, \dots, n_r]^\top) \end{aligned} \quad (12.269)$$

The *thrust coefficient matrix* \mathbf{K} is a diagonal matrix of thrust coefficients given by

$$\mathbf{K} = \text{diag}\{K_1(n_1), K_2(n_2), \dots, K_r(n_r)\} \quad (12.270)$$

The *actuator configuration matrix* $\mathbf{T}(\alpha) \in \mathbb{R}^{3 \times r}$ only depends on the location of the actuators and possible angles α used for rotatable thrusters (azimuth thruster).

Example 12.10 (Supply Vessel Thrust Configuration and Coefficient Matrices)

Computation of $\mathbf{T}(\alpha)$ can be illustrated by considering the supply vessel in Figure 12.32, which is equipped with two main propellers (aft of the ship), two tunnel thrusters and two azimuth thrusters, which can be rotated to arbitrary angles α_1 and α_2 , and therefore produce thrust in different directions; see Figure 12.32. Hence, we have eight control variables (six rpm setpoints and two azimuth angles) for 3 DOF. The control variables are assigned according to (clockwise numbering of u_i):

u_1, α_1	fore azimuth thruster	u_4	aft tunnel thruster
u_2	fore tunnel thruster	u_5	starboard main propeller
u_3, α_2	aft azimuth thruster	u_6	port main propeller

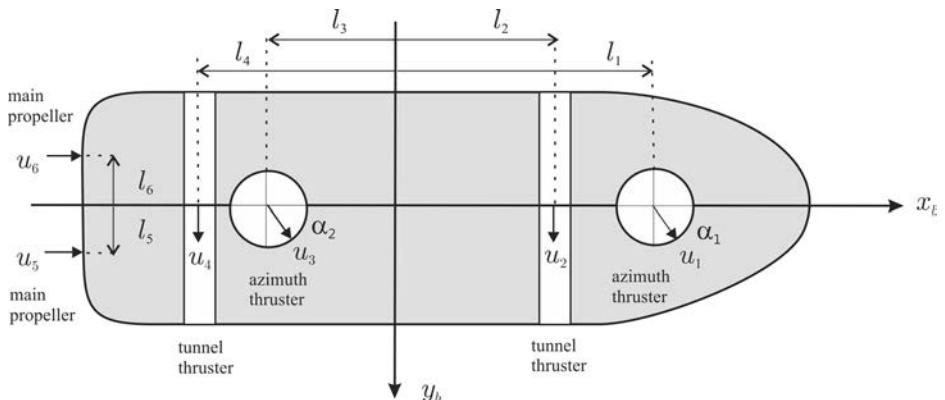


Figure 12.32 Schematic drawing showing the thruster configuration for an offshore supply vessel.

From Section 12.3 it follows that

$$\mathbf{K} = \text{diag}\{K_1, K_2, K_3, K_4, K_5, K_6\} \quad (12.271)$$

$$\mathbf{T}(\boldsymbol{\alpha}) = \begin{bmatrix} \cos(\alpha_1) & 0 & \cos(\alpha_2) & 0 & 1 & 1 \\ \sin(\alpha_1) & 1 & \sin(\alpha_2) & 1 & 0 & 0 \\ l_1 \sin(\alpha_1) & l_2 & l_3 \sin(\alpha_2) & l_4 & -l_5 & -l_6 \end{bmatrix} \quad (12.272)$$

where l_i ($i = 1, \dots, r$) are the moment arms in yaw. It is also seen that $l_5 = -l_6$ (symmetrical location of the main propellers). The thrust demands are defined such that positive thrust results in positive motion according to the VP axis system. The resulting forces and moment are

$$\boldsymbol{\tau} = \mathbf{T}(\boldsymbol{\alpha})\mathbf{K}\mathbf{u} \quad (12.273)$$

‡

$$\begin{bmatrix} X \\ Y \\ N \end{bmatrix} = \begin{bmatrix} \cos(\alpha_1) & 0 & \cos(\alpha_2) & 0 & 1 & 1 \\ \sin(\alpha_1) & 1 & \sin(\alpha_2) & 1 & 0 & 0 \\ l_1 \sin(\alpha_1) & l_2 & l_3 \sin(\alpha_2) & l_4 & -l_5 & -l_6 \end{bmatrix} \cdot \begin{bmatrix} K_1 & 0 & 0 & 0 & 0 & 0 \\ 0 & K_2 & 0 & 0 & 0 & 0 \\ 0 & 0 & K_3 & 0 & 0 & 0 \\ 0 & 0 & 0 & K_4 & 0 & 0 \\ 0 & 0 & 0 & 0 & K_5 & 0 \\ 0 & 0 & 0 & 0 & 0 & K_6 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \\ u_6 \end{bmatrix} \quad (12.274)$$

One of the advantages of the model representation (12.267) is that input uncertainties only appear in the diagonal force coefficient matrix \mathbf{K} , since $\mathbf{T}(\boldsymbol{\alpha})$ will be perfectly known. In fact, this decomposition is highly advantageous since it can be exploited when designing the feedback control system where robust measures for uncertainties in \mathbf{K} must be taken.

Example 12.11 (Supply Vessel Thrust Allocation)

In order to implement a DP control system for the vessel shown in Figure 12.32 a thrust allocation algorithm is needed. The simplest algorithm is the generalized inverse

$$\mathbf{u} = \mathbf{K}^{-1}\mathbf{T}^\dagger(\boldsymbol{\alpha})\boldsymbol{\tau} \quad (12.275)$$

$$\mathbf{T}^\dagger(\boldsymbol{\alpha}) = \mathbf{W}^{-1}\mathbf{T}^\top(\boldsymbol{\alpha})[\mathbf{T}(\boldsymbol{\alpha})\mathbf{W}^{-1}\mathbf{T}^\top(\boldsymbol{\alpha})]^{-1} \quad (12.276)$$

where $\mathbf{W} = \mathbf{W}^T > 0$ is a positive definite weighting matrix, usually chosen to be diagonal. \mathbf{W} should be selected so that using the tunnel and azimuth thrusters is less expensive (small K_i value) than using the main propellers (large K_i value). This solution is easy to use for constant azimuth angles $\boldsymbol{\alpha}$. As soon as $\boldsymbol{\alpha}$ is allowed to vary or the control input saturates, a strategy for this must be developed. This significantly complicates the control allocation software. Many companies solve this in an ad hoc manner and the price is extensive failure testing in order to cover all failure situations. An alternative to this is to use an optimal solution for varying $\boldsymbol{\alpha}$ and limited thrust \mathbf{f} such as the one presented in Section 12.3. This of course requires an iterative solver to be implemented in the control loop and extensive testing is needed to verify that the optimal solution is convergent and stable. For a system with quadratic thrust characteristics, the computed \mathbf{u} values must be mapped to pitch or rpm commands. If $u_i = |p_i|p_i$, it is straightforward to verify that

$$p_i = \text{sgn}(u_i)\sqrt{|u_i|} \quad (12.277)$$

The generalized inverse

$$\mathbf{T}^\dagger(\boldsymbol{\alpha}) = \frac{1}{\det[\mathbf{T}(\boldsymbol{\alpha})\mathbf{W}^{-1}\mathbf{T}^\top(\boldsymbol{\alpha})]} \mathbf{W}^{-1}\mathbf{T}^\top(\boldsymbol{\alpha}) \text{adj}[\mathbf{T}(\boldsymbol{\alpha})\mathbf{W}^{-1}\mathbf{T}^\top(\boldsymbol{\alpha})] \quad (12.278)$$

will be a function of the azimuth angles α_1 and α_2 in Figure 12.32. The expression for the determinant in (12.278) will be nonzero for all combinations of α_1 and α_2 , since the craft has more actuators than needed for 3 DOF stabilization (overactuated). However, for some craft a singular configuration may exist; that is the determinant becomes zero for certain combinations of α_i ($i = 1, \dots, r$). The expression for the determinant can also be used to compute optimal angles α_1 and α_2 in a minimum energy sense by simply maximizing the determinant with respect to α_1 and α_2 .

13

Advanced Motion Control Systems

State-of-the-art motion control systems are usually designed using PID control methods, as described in Chapter 12. This chapter presents more advanced methods for optimal and nonlinear control of marine craft. The main motivation for this is design simplicity and performance. Nonlinear control theory can often yield a more intuitive design than linear theory. Linearization destroys model properties and the results can be a more complicated design process with limited physical insight. Chapter 13 is written for the advanced user who wants to exploit a more advanced model and use this model to improve the performance of the control system. Readers of this chapter need background in optimal and nonlinear control theory.

Preview of the Chapter

Chapter 13 starts with linear quadratic optimal control theory (Section 13.1) with the focus on regulation, trajectory-tracking control and disturbance feedforward. Optimal motion control systems are designed by considering the linearized equations of motion (Section 7.5.3) in the following form:

$$\dot{\mathbf{x}} = \mathbf{Ax} + \mathbf{Bu} + \mathbf{Ew} \quad (13.1)$$

For a marine craft, the linear model (13.1) is based on several assumptions such as zero or constant cruise speed u together with the assumptions that the velocities v , w , p , q and r are small. In addition, the kinematic equation $\dot{\boldsymbol{\eta}} = \mathbf{J}_\Theta(\boldsymbol{\eta})\mathbf{v}$ must be linearized under a set of assumptions on the Euler angles ϕ , θ and ψ .

When linearizing the equations of motion, several model properties such as symmetry of the inertia matrix \mathbf{M} , skew-symmetry of the Coriolis and centripetal matrix $\mathbf{C}(\mathbf{v})$ and positiveness of the damping matrix $\mathbf{D}(\mathbf{v})$ are destroyed, and this often complicates the control design. Also physical properties that are important tools for good engineering judgement are lost. This is seen by comparing the LQ design procedure with the nonlinear techniques in Sections 13.2–13.4. It is also demonstrated how the nonlinear controllers can be related to the PID control design methods in Chapter 12 in particular, under the assumption of setpoint regulation. Often it is useful to think about the nonlinear controller as a PID control system where additional terms are added to obtain global stability results. Keeping this in mind, it is also possible to derive a nonlinear controller using advanced methods and then use engineering insight to simplify the representation of the controller. The resulting controller should be as simple as possible

but still contain the most important terms when implementing the algorithm into a computer. In fact, a so-called simplified nonlinear controller will be recognized as a PID controller with additional terms. Many nonlinear methods are popular due to their simplicity and design flexibility. The assumptions on u, v, w, p, q, r and ϕ, θ, ψ which are needed when linearizing the models are also avoided.

The nonlinear design methods in this chapter are based on the robot-like model of Fossen (1991):

$$\dot{\eta} = J_\theta(\eta)v \quad (13.2)$$

$$M\ddot{v} + C(v)v + D(v)v + g(\eta) = \tau + w \quad (13.3)$$

It is important to understand the physical properties of the model in order to know which terms in the model can be omitted when deriving a model-based nonlinear controller. This is an important question since model inaccuracies can destabilize a feedback control system. Often better results are obtained when uncertain terms are chosen to be zero in the controller.

13.1 Linear Quadratic Optimal Control

Optimal control deals with the problem of finding a control law for a given system such that a certain optimality criterion is achieved. This is usually a cost function that depends on the state and control variables. The optimal control law is a set of differential equations that minimize the cost functional and it can be derived using Pontryagin's maximum principle (a necessary condition) or by solving the Hamilton–Jacobi–Bellman equation (a sufficient condition). We will limit our discussion to linear systems and quadratic cost functions. This is referred to as linear quadratic (LQ) optimal control theory (Athans and Falb, 1966).

13.1.1 Linear Quadratic Regulator

A fundamental design problem is the regulator problem, where it is necessary to regulate the outputs $y \in \mathbb{R}^m$ of the system to zero or a constant value while ensuring that they satisfy time-response specifications. A linear quadratic regulator (LQR) can be designed for this purpose by considering the state-space model

$$\dot{x} = Ax + Bu \quad (13.4)$$

$$y = Cx \quad (13.5)$$

where $x \in \mathbb{R}^n$, $u \in \mathbb{R}^r$ and $y \in \mathbb{R}^m$. In order to design a linear optimal control law the system (A, B, C) must be controllable while observability (see Definition 11.2 in Section 11.2.3) is necessary if some of the states must be estimated. Controllability for linear time-invariant systems is given by the following definition

Definition 13.1 (Controllability)

The state and input matrix (A, B) must satisfy the controllability condition to ensure that there exists a control $u(t)$ that can drive any arbitrary state $x(t_0)$ to another arbitrary state $x(t_1)$ for $t_1 > t_0$. The controllability condition requires that the matrix (Gelb et al., 1988)

$$\mathcal{C} = [B \mid AB \mid \cdots \mid (A)^{n-1}B] \quad (13.6)$$

must be of full row rank such that a right inverse exists.

The feedback control law for the system (13.4)–(13.5) is found by minimizing the quadratic cost function

$$\begin{aligned} J &= \min_u \left\{ \frac{1}{2} \int_0^T (\mathbf{y}^\top \mathbf{Q} \mathbf{y} + \mathbf{u}^\top \mathbf{R} \mathbf{u}) dt \right. \\ &\quad \left. = \frac{1}{2} \int_0^T (\mathbf{x}^\top \mathbf{C}^\top \mathbf{Q} \mathbf{C} \mathbf{x} + \mathbf{u}^\top \mathbf{R} \mathbf{u}) dt \right\} \end{aligned} \quad (13.7)$$

where $\mathbf{R} = \mathbf{R}^\top > 0$ and $\mathbf{Q} = \mathbf{Q}^\top \geq 0$ are the weighting matrices. The steady-state solution to this problem is (Athans and Falb, 1966)

$$\mathbf{u} = \underbrace{-\mathbf{R}^{-1} \mathbf{B}^\top \mathbf{P}_\infty \mathbf{x}}_G \quad (13.8)$$

$$\mathbf{P}_\infty \mathbf{A} + \mathbf{A}^\top \mathbf{P}_\infty - \mathbf{P}_\infty \mathbf{B} \mathbf{R}^{-1} \mathbf{B}^\top \mathbf{P}_\infty + \mathbf{C}^\top \mathbf{Q} \mathbf{C} = \mathbf{0} \quad (13.9)$$

where $\mathbf{P}_\infty = \lim_{t \rightarrow \infty} \mathbf{P}(t)$. The optimal feedback control system is illustrated in Figure 13.1.

Matlab

The steady-state LQR feedback control law is computed as (see the script `ExLQR.m`)

```
Q = diag([1]); % user editable tracking error weights (dim m x m)
R = diag([1]); % user editable input weights (dim r x r)

% System matrices
A = [0 1; -1 -2]; % user editable state matrix (dim n x n)
B = [0; 1]; % user editable input matrix (dim n x r)
C = [1 0]; % user editable output matrix (dim m x n)

% Compute the optimal feedback gain matrix G
[K, P, E] = lqr(A, B, C'*Q*C, R);
G = -K
```

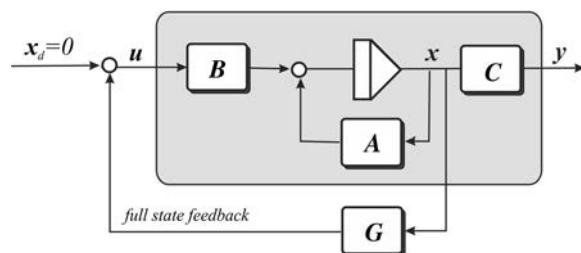


Figure 13.1 Block diagram showing the linear quadratic regulator (LQR).

The Matlab function `lqr.m` also returns the eigenvalues of the closed-loop system

$$\dot{x} = (A + BG)x \quad (13.10)$$

denoted by the symbol Ξ .

13.1.2 LQR Design for Trajectory Tracking and Integral Action

The LQR can be redesigned to track a time-varying reference trajectory $x_d \in \mathbb{R}^n$ for a large class of mechanical systems possessing certain structural properties. This section presents a simple solution to this problem while a more general solution is presented in Section 13.1.3.

Transformation of the LQ Tracker to a Setpoint Regulation Problem

In order to transform a trajectory-tracking problem to a setpoint regulation problem *reference feedforward* can be used. Unmeasured slowly varying or constant disturbances are compensated for by including integral action. This is usually done by augmenting an integral state $\dot{z} = e$ to the system model. A mass–damper–spring system will be used to demonstrate the design methodology.

Example 13.1 (Mass–Damper–Spring Trajectory-Tracking Problem)

Consider the mass–damper–spring system

$$\begin{aligned}\dot{x} &= v \\ m\dot{v} + dv + kx &= \tau\end{aligned}$$

Let

$$\tau = \tau_{FF} + \tau_{LQ} \quad (13.11)$$

where the feedforward term is chosen as

$$\tau_{FF} = m\dot{v}_d + dv_d + kx_d \quad (13.12)$$

such that

$$m\ddot{e} + d\dot{e} + ke = \tau_{LQ} \quad (13.13)$$

where $e = x - x_d$ and $\dot{e} = v - v_d$. The desired states are computed using a reference model:

$$\dot{x}_d = v_d \quad (13.14)$$

$$\dot{v}_d = \phi(v_d, r) \quad (13.15)$$

where r is the setpoint. The trajectory-tracking control problem has now been transformed to an LQ setpoint regulation problem given by (13.13), which can be written in state-space form as

$$\begin{aligned}\dot{x} &= \underbrace{\begin{bmatrix} 0 & 1 \\ -\frac{d}{m} & -\frac{k}{m} \end{bmatrix}}_A x + \underbrace{\begin{bmatrix} 0 \\ \frac{1}{m} \end{bmatrix}}_B u \\ e &= \underbrace{\begin{bmatrix} 1 & 0 \end{bmatrix}}_C x\end{aligned}$$

where $x = [e, \dot{e}]^\top$ and $u = \tau_{LQ}$.

Integral Action

In Example 13.1 it was shown that a feedforward term τ_{FF} could transform the LQ trajectory-tracking problem to an LQR problem. For the system model

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} \quad (13.16)$$

integral action is obtained by augmenting the integral state $\mathbf{z} \in \mathbb{R}^m$ to the state vector. Let

$$\dot{\mathbf{z}} = \mathbf{y} = \mathbf{C}\mathbf{x} \quad (13.17)$$

where the \mathbf{C} matrix is used to extract potential integral states from the \mathbf{x} vector. This system is a standard LQR problem:

$$\dot{\mathbf{x}}_a = \mathbf{A}_a\mathbf{x}_a + \mathbf{B}_a\mathbf{u} \quad (13.18)$$

where $\mathbf{x}_a = [\mathbf{z}^\top, \mathbf{x}^\top]^\top$ and

$$\mathbf{A}_a = \begin{bmatrix} \mathbf{0} & \mathbf{C} \\ \mathbf{0} & \mathbf{A} \end{bmatrix}, \quad \mathbf{B}_a = \begin{bmatrix} \mathbf{0} \\ \mathbf{B} \end{bmatrix} \quad (13.19)$$

The control objective is regulation of \mathbf{x}_a to zero using \mathbf{u} . This is obtained by choosing the performance index

$$J = \min_{\mathbf{u}} \left\{ \frac{1}{2} \int_0^t (\mathbf{x}_a^\top \mathbf{Q}_a \mathbf{x}_a + \mathbf{u}^\top \mathbf{R} \mathbf{u}) d\tau \right\} \quad (13.20)$$

where $\mathbf{R} = \mathbf{R}^\top > 0$ and $\mathbf{Q}_a = \mathbf{Q}_a^\top \geq 0$ are the weighting matrices. Hence, the solution of the LQR setpoint regulation problem is (see Section 13.1.1)

$$\begin{aligned} \mathbf{u} &= -\mathbf{R}^{-1} \mathbf{B}_a^\top \mathbf{P}_\infty \mathbf{x}_a \\ &= -\mathbf{R}^{-1} [\mathbf{0} \quad \mathbf{B}^\top] \begin{bmatrix} \mathbf{P}_{11} & \mathbf{P}_{12} \\ \mathbf{P}_{21} & \mathbf{P}_{22} \end{bmatrix} \begin{bmatrix} \mathbf{z} \\ \mathbf{x} \end{bmatrix} \\ &= -\underbrace{\mathbf{R}^{-1} \mathbf{B}^\top \mathbf{P}_{12}}_{\mathbf{K}_i} \mathbf{z} - \underbrace{\mathbf{R}^{-1} \mathbf{B}^\top \mathbf{P}_{22}}_{\mathbf{K}_p} \mathbf{x} \end{aligned} \quad (13.21)$$

where \mathbf{P}_{12} and \mathbf{P}_{22} are found by solving the *algebraic Riccati equation* (ARE)

$$\mathbf{P}_\infty \mathbf{A}_a + \mathbf{A}_a^\top \mathbf{P}_\infty - \mathbf{P}_\infty \mathbf{B}_a \mathbf{R}^{-1} \mathbf{B}_a^\top \mathbf{P}_\infty + \mathbf{Q}_a = \mathbf{0} \quad (13.22)$$

Notice that the feedback term \mathbf{u} includes feedback from the tracking errors \mathbf{e} and $\dot{\mathbf{e}}$ as well as the integral state

$$\mathbf{z} = \int_0^t \mathbf{e}(\tau) d\tau \quad (13.23)$$

13.1.3 General Solution of the LQ Trajectory-Tracking Problem

Consider the state-space model

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} + \mathbf{E}\mathbf{w} \quad (13.24)$$

$$\mathbf{y} = \mathbf{C}\mathbf{x} \quad (13.25)$$

The LQ trajectory-tracking control problem is addressed under the assumption that both the state vector \mathbf{x} and disturbance vector \mathbf{w} are measured or at least obtained by state estimation. If the estimated values are used for \mathbf{x} and \mathbf{w} , stability can be proven by applying a *separation principle*. This is known as LQG control in the literature and involves the design of a *Kalman filter* for reconstruction of the unmeasured states, which again requires that the system is *observable*. For simplicity, full-state feedback is assumed in this chapter. The interested reader is recommended to consult the extensive literature on LQG control for output feedback control; see Athans and Falb (1966) and Brian *et al.* (1989), for instance.

Reference Feedforward Assumptions

Consider a time-varying reference system:

$$\dot{\mathbf{x}}_d = \phi(\mathbf{x}_d, \mathbf{r}) \quad (13.26)$$

$$\mathbf{y}_d = \mathbf{C}\mathbf{x}_d \quad (13.27)$$

where $\mathbf{x}_d \in \mathbb{R}^n$ is the desired state, $\mathbf{y}_d \in \mathbb{R}^p$ ($p \leq n$) is the desired output, $\mathbf{r} \in \mathbb{R}^r$ ($r \leq n$) is the setpoint and $\phi : \mathbb{R}^n \times \mathbb{R}^r \rightarrow \mathbb{R}^p$. If linear theory is assumed the dynamics of the desired state can be conveniently represented by

$$\phi(\mathbf{x}_d, \mathbf{r}) = \mathbf{A}_d\mathbf{x}_d + \mathbf{B}_d\mathbf{r} \quad (13.28)$$

This is a linear reference model for trajectory-tracking control; see Section 10.2.1 for how to choose \mathbf{A}_d and \mathbf{B}_d . A special case is regulation:

$$\mathbf{y}_d = \mathbf{C}\mathbf{x}_d = \text{constant} \quad (13.29)$$

Disturbance Feedforward Assumptions

Two cases of disturbance feedforward are considered:

1. The disturbance vector $\mathbf{w} = \text{constant}$ for all $t > T_p$, where T_p is the present time. An example of this is a marine craft exposed to constant (or at least slowly varying) wind forces. This is a reasonable assumption since the *average* wind speed and direction are not likely to change in minutes.
2. The disturbance $\mathbf{w} = \mathbf{w}(t)$ varies as a function of time t for future time $t > T_p$. This is the case for most physical disturbances. However, a feedforward solution requires that \mathbf{w} is known (or at least estimated) for $t \geq 0$. In many cases this is unrealistic so the best we can do is to assume that $\mathbf{w}(t) = \mathbf{w}(T_p) = \text{constant}$, that is in a finite future time horizon so that it conforms to Case 1 above.

Control Objective

The control objective is to design a linear quadratic optimal trajectory-tracking controller using a time-varying smooth reference trajectory \mathbf{y}_d given by the system (13.26)–(13.27). Assume that the desired output $\mathbf{y}_d = \mathbf{C}\mathbf{x}_d$ is known for all time $t \in [0, T]$, where T is the final time. Define the error signal:

$$\begin{aligned} \mathbf{e} &:= \mathbf{y} - \mathbf{y}_d \\ &= \mathbf{C}(\mathbf{x} - \mathbf{x}_d) \end{aligned} \quad (13.30)$$

The goal is to design an optimal trajectory-tracking controller that tracks the desired output, that is regulates the error \mathbf{e} to zero while minimizing

$$J = \min_u \left\{ \frac{1}{2} \mathbf{e}^\top(T) \mathbf{Q}_f \mathbf{e}(T) + \frac{1}{2} \int_{t_0}^T (\mathbf{e}^\top \mathbf{Q} \mathbf{e} + \mathbf{u}^\top \mathbf{R} \mathbf{u}) dt \right\}$$

subject to $\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} + \mathbf{E}\mathbf{w}, \quad \mathbf{x}(0) = \mathbf{x}_0$

(13.31)

where $\mathbf{R} = \mathbf{R}^\top > 0$ and $\mathbf{Q} = \mathbf{Q}^\top \geq 0$ are the tracking error and control weighting matrices, respectively. The weight matrix $\mathbf{Q}_f = \mathbf{Q}_f^\top \geq 0$ can be included to add penalty to the final state. Notice that this is a *finite time-horizon* optimal control problem and it has to be solved by using the *differential Riccati equation* (DRE); see Athans and Falb (1966, pp. 793–801).

It is assumed that the desired output signal comes from a linear reference generator given by

$$\dot{\mathbf{x}}_d = \mathbf{A}_d \mathbf{x}_d + \mathbf{B}_d \mathbf{r} \quad (13.32)$$

$$\mathbf{y} = \mathbf{C} \mathbf{x}_d \quad (13.33)$$

where \mathbf{r} is a given reference input, which is filtered through the generator. \mathbf{C} is the same output matrix as in the plant. A special case of (13.31) is the one with no weight on the final state; that is $\mathbf{Q}_f = \mathbf{0}$, resulting in the quadratic performance index

$$J = \min_u \left\{ \frac{1}{2} \int_0^T (\mathbf{e}^\top \mathbf{Q} \mathbf{e} + \mathbf{u}^\top \mathbf{R} \mathbf{u}) dt \right\} \quad (13.34)$$

Substituting (13.30) into (13.34) yields the equivalent formulation

$$J = \min_u \left\{ \frac{1}{2} \int_0^T (\tilde{\mathbf{x}}^\top \tilde{\mathbf{Q}} \tilde{\mathbf{x}} + \mathbf{u}^\top \mathbf{R} \mathbf{u}) dt \right\} \quad (13.35)$$

where $\tilde{\mathbf{x}} = \mathbf{x} - \mathbf{x}_d$ and

$$\tilde{\mathbf{Q}} = \mathbf{C}^\top \mathbf{Q} \mathbf{C} \geq 0 \quad (13.36)$$

Linear Time-Varying Systems

It can be shown that the optimal control law is (Brian *et al.*, 1989)

$$\mathbf{u} = -\mathbf{R}^{-1} \mathbf{B}^\top [\mathbf{P}\mathbf{x} + \mathbf{h}_1 + \mathbf{h}_2] \quad (13.37)$$

where \mathbf{P} , \mathbf{h}_1 and \mathbf{h}_2 originate from the *Hamiltonian* system. \mathbf{P} accounts for the feedback part, \mathbf{h}_1 accounts for the feedforward part due to the time-varying nature of the reference signal \mathbf{y}_d and \mathbf{h}_2 accounts for the feedforward part due to the measurable time-varying disturbance \mathbf{w} . The equations that need to be solved are

$$\dot{\mathbf{P}} = -\mathbf{P}\mathbf{A} - \mathbf{A}^\top \mathbf{P} + \mathbf{P}\mathbf{B}\mathbf{R}^{-1}\mathbf{B}^\top \mathbf{P} - \tilde{\mathbf{Q}} \quad (13.38)$$

$$\dot{\mathbf{h}}_1 = -[\mathbf{A} - \mathbf{B}\mathbf{R}^{-1}\mathbf{B}^\top \mathbf{P}]^\top \mathbf{h}_1 + \tilde{\mathbf{Q}}\mathbf{x}_d \quad (13.39)$$

$$\dot{\mathbf{h}}_2 = -[\mathbf{A} - \mathbf{B}\mathbf{R}^{-1}\mathbf{B}^\top \mathbf{P}]^\top \mathbf{h}_2 - \mathbf{P}\mathbf{E}\mathbf{w} \quad (13.40)$$

with

$$\mathbf{P}(T) = \tilde{\mathbf{Q}}_f \quad (13.41)$$

$$\mathbf{h}_1(T) = -\tilde{\mathbf{Q}}_f \mathbf{x}_d(T) \quad (13.42)$$

$$\mathbf{h}_2(T) = \mathbf{0} \quad (13.43)$$

where $\tilde{\mathbf{Q}}_f = \mathbf{C}^\top \mathbf{Q}_f \mathbf{C}$. Equations (13.38)–(13.40) represent three differential equations: a matrix DRE and two vector differential equations (adjoint operators), respectively. Notice that the initial conditions for these equations are not known, but rather the final conditions are known. Consequently, they have to be integrated *backward* in time a priori to find the initial conditions, and then be executed forward in time again with the closed-loop plant from $[0, T]$.

There are different ways of doing this. A frequently used method is to discretize the system and run the resulting difference equation backward. A simple Euler integration routine for (13.38) is given below, where δ is set as a small *negative* sampling time. Moreover, using a first-order Taylor expansion

$$\mathbf{P}(t + \delta) \approx \mathbf{P}(t) + \delta\{-\mathbf{P}\mathbf{A} - \mathbf{A}^\top \mathbf{P} + \mathbf{P}\mathbf{B}\mathbf{R}^{-1}\mathbf{B}^\top \mathbf{P} - \tilde{\mathbf{Q}}\} \quad (13.44)$$

with $\mathbf{P}(T) = \tilde{\mathbf{Q}}_f$ produces $\mathbf{P}(0)$. Another procedure is to simulate backwards in time. The system

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, t) + \mathbf{G}(\mathbf{x}, t)\mathbf{u}, \quad t \in [T, 0] \quad (13.45)$$

can be simulated backwards in time by the following change of integration variable $t = T - \tau$ with $dt = -d\tau$, and

$$-\frac{d\mathbf{x}(T - \tau)}{d\tau} = \mathbf{f}(\mathbf{x}(T - \tau), T - \tau) + \mathbf{G}(\mathbf{x}(T - \tau), T - \tau)\mathbf{u}(T - \tau) \quad (13.46)$$

Let $\mathbf{z}(\tau) = \mathbf{x}(T - \tau)$; then

$$\frac{d\mathbf{z}(\tau)}{d\tau} = -\mathbf{f}(\mathbf{z}(\tau), T - \tau) - \mathbf{G}(\mathbf{z}(\tau), T - \tau)\mathbf{u}(T - \tau) \quad (13.47)$$

This system can now be simulated forward in time with the initial condition $\mathbf{z}(0) = \mathbf{x}(T)$.

The method is demonstrated in Example 13.2, where it is assumed that both \mathbf{x}_d and \mathbf{w} are time varying but known for all future t . A special case dealing with constant values for \mathbf{x}_d and \mathbf{w} will be studied later.

Example 13.2 (Optimal Time-Varying LQ Trajectory-Tracking Problem)

Consider a mass-damper-spring system:

$$m\ddot{x} + d\dot{x} + kx = u + w \quad (13.48)$$

where m is the mass, d is the damping coefficient, k is the spring stiffness coefficient, u is the input and w is the disturbance. Choosing the states as $x_1 = x$ and $x_2 = \dot{x}$, the following state-space realization is obtained:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\frac{k}{m} & -\frac{d}{m} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{m} \end{bmatrix} u + \begin{bmatrix} 0 \\ \frac{1}{m} \end{bmatrix} w \quad (13.49)$$

For simplicity, assume that $m = k = 1$ and $d = 2$ such that

$$\dot{\mathbf{x}} = \begin{bmatrix} 0 & 1 \\ -1 & -2 \end{bmatrix} \mathbf{x} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u + \begin{bmatrix} 0 \\ 1 \end{bmatrix} w \quad (13.50)$$

$$y = [1 \ 0] \mathbf{x} \quad (13.51)$$

where $\mathbf{x} = [x_1, x_2]^\top$. The disturbance signal is assumed to be known for all future time and is simply chosen as

$$w = \cos(t) \quad (13.52)$$

Similarly, the reference signal is assumed to be known for all future time and is given by the generator

$$\dot{\mathbf{x}}_d = \begin{bmatrix} 0 & 1 \\ -1 & -1 \end{bmatrix} \mathbf{x}_d + \begin{bmatrix} 0 \\ 1 \end{bmatrix} r \quad (13.53)$$

$$y_d = [1 \ 0] \mathbf{x}_d \quad (13.54)$$

where

$$r = \sin(t) \quad (13.55)$$

The Matlab MSS toolbox script `ExLQFinHor.m` demonstrates how forward and backward integration can be implemented for the mass-damper-spring system. The simulation results are shown in Figures 13.2–13.3.

Approximate Solution for Linear Time-Invariant Systems

Unfortunately, the theory dealing with the limiting case

$$J = \min_u \left\{ \frac{1}{2} \lim_{T \rightarrow \infty} \int_0^T (\mathbf{e}^\top \mathbf{Q} \mathbf{e} + \mathbf{u}^\top \mathbf{R} \mathbf{u}) dt \right\} \quad (13.56)$$

is not available. This solution is very useful since it represents a steady-state solution of the LQ trajectory-tracking problem. Fortunately, this problem can be circumvented by assuming that T is large but still limited; that is

$$0 \ll T_1 \leq T < \infty \quad (13.57)$$

where T_1 is a large constant. For $T \rightarrow \infty$ the solution of (13.38) will tend to the constant matrix \mathbf{P}_∞ satisfying the *algebraic Riccati equation* (ARE)

$$\mathbf{P}_\infty \mathbf{A} + \mathbf{A}^\top \mathbf{P}_\infty - \mathbf{P}_\infty \mathbf{B} \mathbf{R}^{-1} \mathbf{B}^\top \mathbf{P}_\infty + \tilde{\mathbf{Q}} = \mathbf{0} \quad (13.58)$$

This solution is interpreted as the steady-state solution of (13.38) where $\mathbf{P}(t) \approx \mathbf{P}_\infty$ for all $t \in [0, T_1]$. This is verified in the upper plot of Figure 13.3. Furthermore, it is assumed that

$$\mathbf{x}_d = \text{constant}, \quad \mathbf{w} = \text{constant}, \quad \forall t \in [0, T_1] \quad (13.59)$$

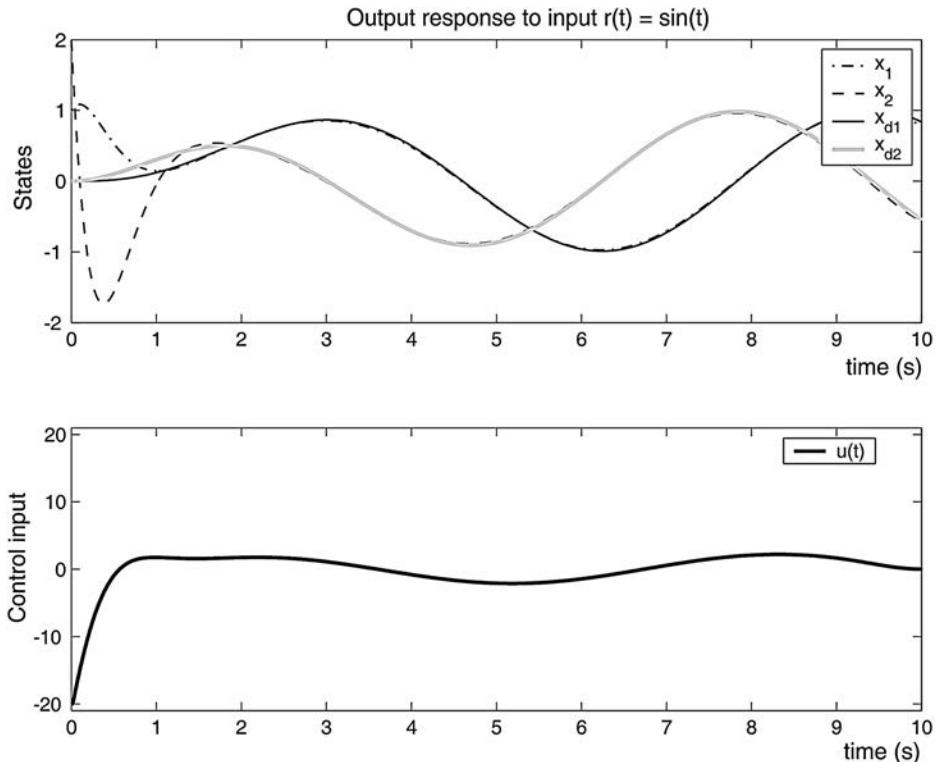


Figure 13.2 Upper plot: states x_1 and x_2 and the reference trajectories x_{d1} and x_{d2} as a function of time. Lower plot: optimal control u as a function of time.

In practice the assumption that \mathbf{x}_d is constant can be relaxed with \mathbf{x}_d being slowly varying compared to the state dynamics. A similar argument can be used for \mathbf{w} . It is also common to drop the disturbance feedforward term since integral action in the controller can compensate for nonzero slowly varying disturbances.

Next, if the eigenvalues of the matrix

$$\mathbf{A}_c = \mathbf{A} + \mathbf{B}\mathbf{G}_1 \quad \text{where} \quad \mathbf{G}_1 = -\mathbf{R}^{-1}\mathbf{B}^\top \mathbf{P}_\infty \quad (13.60)$$

have negative real parts

$$\lambda_i(\mathbf{A}_c) < 0 \quad (i = 1, \dots, n) \quad (13.61)$$

the steady-state solution for \mathbf{h}_1 and \mathbf{h}_2 in (13.39) and (13.40) on $[0, T_1]$ becomes

$$\mathbf{h}_{1\infty} = (\mathbf{A} + \mathbf{B}\mathbf{G}_1)^{-\top} \tilde{\mathbf{Q}}\mathbf{x}_d \quad (13.62)$$

$$\mathbf{h}_{2\infty} = -(\mathbf{A} + \mathbf{B}\mathbf{G}_1)^{-\top} \mathbf{P}_\infty \mathbf{E}\mathbf{w} \quad (13.63)$$

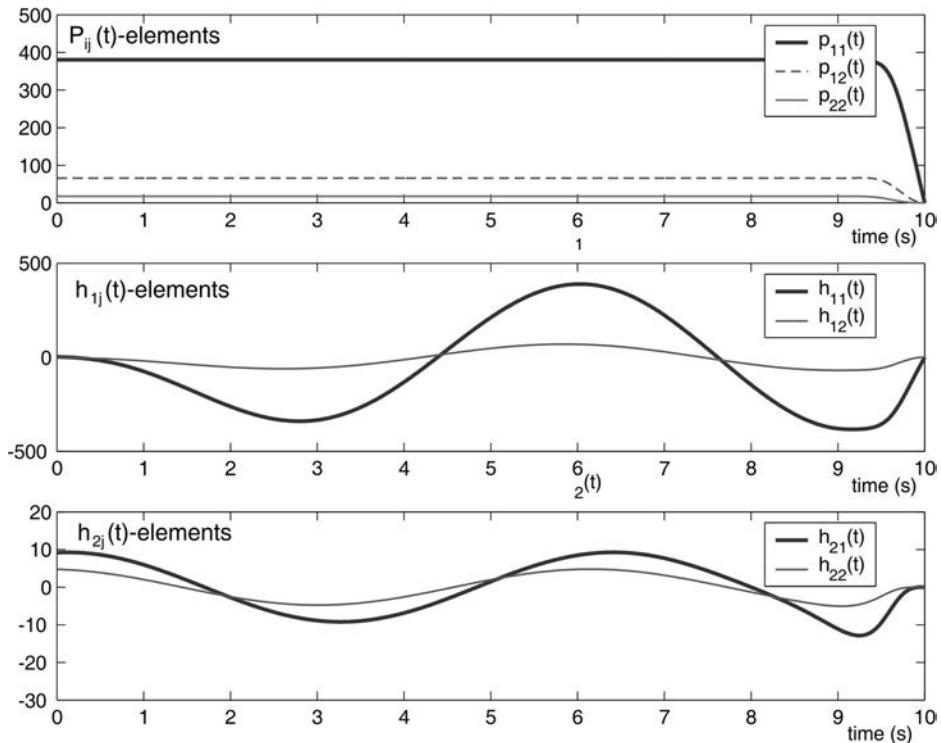


Figure 13.3 Optimal solutions of the elements in \mathbf{P} , \mathbf{h}_1 and \mathbf{h}_2 as a function of time.

Substitution of (13.58) into (13.37) yields the steady-state optimal control law (see Figure 13.4)

$$\mathbf{u} = \mathbf{G}_1 \mathbf{x} + \mathbf{G}_2 \mathbf{y}_d + \mathbf{G}_3 \mathbf{w} \quad (13.64)$$

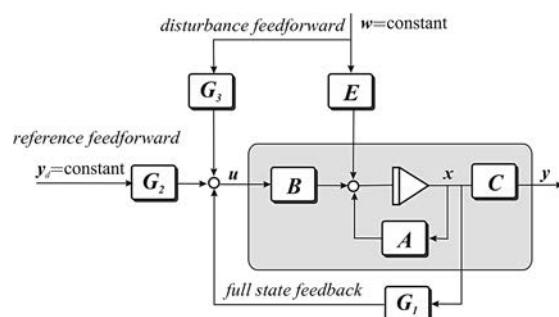


Figure 13.4 Block diagram showing the full state feedback LQ tracker solution with disturbance feedforward.

where $y_d = \text{constant}$ and $w = \text{constant}$, and

$$G_1 = -R^{-1}B^\top P_\infty \quad (13.65)$$

$$G_2 = -R^{-1}B^\top (A + BG_1)^{-\top} C^\top Q \quad (13.66)$$

$$G_3 = R^{-1}B^\top (A + BG_1)^{-\top} P_\infty E \quad (13.67)$$

Matlab

The function `lqtracker.m` is implemented in the MSS toolbox for computation of the matrices G_1 , G_2 and G_3 :

```
function [G1,G2,G3] = lqtracker(A,B,C,Q,R)
[K,P,E] = lqr(A,B,C'*Q*C,R);
G1 = -inv(R)*B'*P;
Temp = inv((A+B*G1)');
G2 = -inv(R)*B'*Temp*C'*Q;
G3 = inv(R)*B'*Temp*P*E;
```

For a mass–damper–spring system the optimal trajectory tracking controller is found using `ExLQtrack.m`:

```
%Design matrices
Q = diag([1]); % tracking error weights
R = diag([1]); % input weights

% System matrices
A = [0 1; -1 -2]; % state matrix
B = [0; 1]; % input matrix
C = [1 0]; % output matrix

% Optimal gain matrices
[G1,G2,G3] = lqtracker(A,B,C,Q,R)
```

SISO Systems

Consider the SISO state-space model

$$\dot{x} = Ax + bu + Ew \quad (13.68)$$

$$y = c^\top x \quad (13.69)$$

where $x \in \mathbb{R}^n$, $u \in \mathbb{R}$ and $y \in \mathbb{R}$. For SISO systems, the performance index (13.34) simplifies to

$$\begin{aligned} J &= \min_u \left\{ \frac{1}{2} \lim_{T \rightarrow \infty} \int_0^T (q e^2 + r u^2) dt \right. \\ &\quad \left. = \frac{q}{2} \lim_{T \rightarrow \infty} \int_0^T \left(e^2 + \frac{r}{q} u^2 \right) dt \right\} \end{aligned} \quad (13.70)$$

where $q \geq 0$ and $r > 0$ are two scalars. By choosing $q = 1$ (without loss of generality) and defining $\lambda := r/q > 0$, the performance index (13.70) changes to

$$J^* = \min_u \left\{ \frac{1}{2} \lim_{T \rightarrow \infty} \int_{t_0}^T (e^2 + \lambda u^2) dt \right\} \quad (13.71)$$

Consequently, the steady-state optimal solution can be approximated as

$$u = \mathbf{g}_1^\top \mathbf{x} + g_2 y_d + \mathbf{g}_3^\top \mathbf{w} \quad (13.72)$$

where

$$\mathbf{g}_1^\top = -\frac{1}{\lambda} \mathbf{b}^\top \mathbf{P}_\infty \quad (13.73)$$

$$g_2 = -\frac{1}{\lambda} \mathbf{b}^\top (\mathbf{A} + \mathbf{b}\mathbf{g}_1^\top)^{-\top} \mathbf{c} \quad (13.74)$$

$$\mathbf{g}_3^\top = \frac{1}{\lambda} \mathbf{b}^\top (\mathbf{A} + \mathbf{b}\mathbf{g}_1^\top)^{-\top} \mathbf{P}_\infty \mathbf{E} \quad (13.75)$$

Here $\mathbf{P}_\infty = \mathbf{P}_\infty^\top > 0$ is the solution of the ARE:

$$\mathbf{P}_\infty \mathbf{A} + \mathbf{A}^\top \mathbf{P}_\infty - \frac{1}{\lambda} \mathbf{P}_\infty \mathbf{b} \mathbf{b}^\top \mathbf{P}_\infty + \mathbf{c} \mathbf{c}^\top = \mathbf{0} \quad (13.76)$$

For a mass–damper–spring system the term $\mathbf{g}_1^\top \mathbf{x}$ can be viewed as a PD controller while $g_2 y_d$ and $\mathbf{g}_3^\top \mathbf{w}$ represent reference and disturbance feedforward, respectively.

13.1.4 Case Study: Optimal Heading Autopilot for Ships and Underwater Vehicles

Autopilots for rudder-controlled ships and underwater vehicles can be designed by considering a linear quadratic optimization problem:

$$J = \min_\delta \left\{ \frac{\alpha}{T} \int_0^T [e^2 + \lambda_1 r^2 + \lambda_2 \delta^2] d\tau \right\} \quad (13.77)$$

where α is a constant to be interpreted later, $e = \psi_d - \psi$ is the heading error, δ is the actual rudder angle and λ_1 and λ_2 are two factors weighting the cost of heading errors e and heading rate r against the control effort δ . This criterion can also be reformulated to describe marine craft that not are turned using a single

rudder by replacing the quadratic term δ^2 with other control inputs. In the forthcoming, we will restrict our analysis to a single input.

For marine craft, operation in restricted waters usually requires accurate control, while the minimization of fuel consumption is more important in open seas. This can be obtained by changing the weights λ_1 and λ_2 . We will discuss three criteria for control weighting that have all been derived by considering a ship. However, the same principles apply for underwater vehicles.

The Steering Criterion of Koyama

The first criterion was derived by Koyama (1967) who observed that the ship's swaying motion y could be approximated by a sinusoid

$$y = \sin(et) \implies \dot{y} = e \cos(et) \quad (13.78)$$

during autopilot control. The length of one arch L_a of the sinusoid is

$$L_a = \int_0^\pi \sqrt{(1 + \dot{y}^2)} d\tau = \int_0^\pi \sqrt{[1 + e^2 \cos^2(e\tau)]} d\tau \approx \pi \left(1 + \frac{e^2}{4}\right) \quad (13.79)$$

Hence, the relative elongation due to a sinusoidal course error is

$$\frac{\Delta L}{L} = \frac{L_a - L}{L} = \frac{\pi(1 + e^2/4) - \pi}{\pi} = \frac{e^2}{4} \quad (13.80)$$

This suggests that the percentage loss of speed during course control can be calculated by using the elongation in distance due to a sinusoidal course error. Consequently, Koyama (1967) proposed minimizing the speed loss term $e^2/4$ against the increased resistance due to steering given by the quadratic term δ^2 . This motivates the following criterion:

$$J = \min_{\delta} \left\{ 100 \left(\frac{\pi}{180} \right)^2 \frac{1}{T} \int_0^T \left[\frac{e^2}{4} + \lambda_2 \delta^2 \right] d\tau \approx \frac{0.0076}{T} \int_0^T [e^2 + \lambda_2 \delta^2] d\tau \right\} \quad (13.81)$$

In this context (13.77) can be interpreted as

$$J = \text{loss of speed (\%)} \quad (13.82)$$

$$\alpha = 0.0076 \quad (13.83)$$

Notice that $\lambda_1 = 0$ for this method. In practice it might be desirable to penalize r^2 by choosing $\lambda_1 > 0$. For ships, Koyama suggested a λ_2 factor of approximately 8–10. Experiments show that such high values for λ_2 avoid large rudder angles, and thus high turning rates. Therefore, $\lambda_2 = 10$ will be a good choice in bad weather, where it is important to suppress high-frequency rudder motions.

Norrbin's Steering Criterion

Another approach for computation of λ_2 was proposed by Norrbin (1972). Consider the surge dynamics of a rudder-controlled marine craft in the form

$$(m - X_u)\dot{u} = X_{|u|u}|u|u + (1 - t)T + T_{\text{loss}} \quad (13.84)$$

where

$$T_{\text{loss}} = (m + X_{vr})vr + X_{cc\delta\delta}c^2 \delta^2 + (X_{rr} + mx_g)r^2 + X_{\text{ext}} \quad (13.85)$$

Norrbin (1972) suggested minimizing the loss term T_{loss} to obtain maximum forward speed u . Consequently, the controller should minimize the centripetal term vr , the square rudder angle δ^2 and the square heading rate r^2 , while the unknown disturbance term X_{ext} is neglected in the analysis. The assumptions in doing this are as follows:

1. The sway velocity v is approximately proportional to r . From Section 7.2 it follows that

$$v(s) = \frac{K_v(T_v s + 1)}{K(T s + 1)} r(s) \approx \frac{K_v}{K} r(s) \quad (13.86)$$

if $T_v \approx T$. Hence, the centripetal term vr will be approximately proportional to the square of the heading rate; that is $vr \approx (K_v/K)r^2$.

2. The ship's yawing motion is periodic under autopilot control such that

$$r_{\max} = \omega_r e_{\max} \quad (13.87)$$

where ω_r is the frequency of the sinusoidal yawing.

These two assumptions suggest that the loss term T_{loss} can be minimized by minimizing e^2 and δ^2 which is the same result obtained in Koyama's analysis. The only difference between the criteria of Norrbin and Koyama is that the λ_2 values arising from Norrbin's approach will be different when computed for the same ship. The performance of the controller also depends on the sea state. This suggests that a trade-off between the λ_2 values proposed by Koyama and Norrbin could be made according to

$\underbrace{\text{(calm sea)}}_{\text{Norrbin}}$	$0.1 \leq \lambda_2 \leq 10$	$\underbrace{\text{(rough sea)}}_{\text{Koyama}}$
---	------------------------------	---

(13.88)

Van Amerongen and Van Nauta Lemke's Steering Criterion

Experiments with the steering criteria of Koyama and Norrbin soon showed that the performance could be further improved by considering the squared yaw rate r^2 , in addition to e^2 and δ^2 (Van Amerongen and Van Nauta Lemke, 1978). Consequently, the following criterion was proposed:

$$J = \min_{\delta} \left\{ \frac{0.0076}{T} \int_0^T (e^2 + \lambda_1 r^2 + \lambda_2 \delta^2) d\tau \right\} \quad (13.89)$$

For a tanker and a cargo ship, Van Amerongen and Van Nauta Lemke (1978, 1980) gave the following values for the weighting factors λ_1 and λ_2 corresponding to the data set of Norrbin (1972):

Tanker: $L_{pp} = 300$ m, $\lambda_1 = 15\ 000$, $\lambda_2 = 8.0$

Cargo ship: $L_{pp} = 200$ m, $\lambda_1 = 1\ 600$, $\lambda_2 = 6.0$

The solution of the optimal steering criteria is found by considering Nomoto's first-order model in the form

$$\dot{\psi}' = r' \quad (13.90)$$

$$T'\dot{r} + (U/L)r = (U/L)^2 K' \delta \quad (13.91)$$

Straightforward application of optimal control theory to the criterion of Van Amerongen and Van Nauta Lempke (1978) yields (see Section 13.1.3)

$$\delta = -K_p(\psi - \psi_d) - K_d r \quad (13.92)$$

where the controller gains are computed using the steady-state solution (13.73) and (13.76). This gives

$$K_p = \sqrt{\frac{1}{\lambda_2}} \quad (13.93)$$

$$K_d = \frac{L}{U} \frac{\sqrt{1 + 2K_p K' T' + K'^2 (U/L)^2 (\lambda_1/\lambda_2)} - 1}{K'} \quad (13.94)$$

Consequently, the solution of the criteria of Koyama and Norrbin is obtained by setting $\lambda_1 = 0$ and $\lambda_2 = \lambda$, which yields

$$K_p = \sqrt{\frac{1}{\lambda}} \quad (13.95)$$

$$K_d = \frac{L}{U} \frac{\sqrt{1 + 2K_p K' T'} - 1}{K'} \quad (13.96)$$

From these expressions it is seen that K_p depends on the weighting factor λ , while K_d depends on K_p as well as the model parameters K' and T' . Hence, accurate steering requires that K' and T' are known with sufficient accuracy.

An extension to Nomoto's second-order model is found by considering the state-space model (see Section 7.2)

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} \quad (13.97)$$

$$\mathbf{y} = \mathbf{C}\mathbf{x} \quad (13.98)$$

where $\mathbf{x} = [v, r, \psi]^\top$, $\mathbf{u} = \delta$, $\mathbf{y} = [r, \psi]^\top$ and

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & 0 \\ a_{21} & a_{22} & 0 \\ 0 & 1 & 0 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} b_1 \\ b_2 \\ 0 \end{bmatrix}, \quad \mathbf{C} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (13.99)$$

Let $\mathbf{y} = [0, \psi_d]^\top = \text{constant}$ and

$$\begin{aligned}\mathbf{e} &= \mathbf{y} - \mathbf{y}_d \\ &= \mathbf{C}(\mathbf{x} - \mathbf{x}_d)\end{aligned}\quad (13.100)$$

The steady-state optimal solution minimizing the quadratic performance index

$$J = \min_{\mathbf{u}} \left\{ \frac{1}{2} \int_0^T (\mathbf{e}^\top \mathbf{Q} \mathbf{e} + \mathbf{u}^\top \mathbf{R} \mathbf{u}) d\tau \right\} \quad (13.101)$$

where $\mathbf{Q} = \text{diag}\{q_{11}, q_{22}\} \geq 0$ and $\mathbf{R} = r_{11} > 0$ are the weights (see Section 13.1.3)

$$\mathbf{u} = \mathbf{G}_1 \mathbf{x} + \mathbf{G}_2 \mathbf{y}_d \quad (13.102)$$

where

$$\mathbf{G}_1 = -\mathbf{R}^{-1} \mathbf{B}^\top \mathbf{P}_\infty \quad (13.103)$$

$$\mathbf{G}_2 = -\mathbf{R}^{-1} \mathbf{B}^\top (\mathbf{A} + \mathbf{B} \mathbf{G}_1)^{-\top} \mathbf{C}^\top \mathbf{Q} \quad (13.104)$$

and \mathbf{P}_∞ is the solution of the matrix Riccati equation:

$$\mathbf{P}_\infty \mathbf{A} + \mathbf{A}^\top \mathbf{P}_\infty - \mathbf{P}_\infty \mathbf{B} \mathbf{R}^{-1} \mathbf{B}^\top \mathbf{P}_\infty + \mathbf{C}^\top \mathbf{Q} \mathbf{C} = \mathbf{0} \quad (13.105)$$

The robustness of optimal autopilots for course-keeping control with a state estimator is analyzed in Holzhüter (1992).

13.1.5 Case Study: Optimal Fin and Rudder-Roll Damping Systems for Ships

The roll motion of ships and underwater vehicles can be damped by using fins alone or in combination with rudders. The main motivation for using roll stabilizing systems on merchant ships is to prevent cargo damage and to increase the effectiveness of the crew by avoiding or reducing seasickness. This is also important from a safety point of view. For naval ships critical marine operations include landing a helicopter, formation control, underway replenishment, or the effectiveness of the crew during combat.

Several *passive* and *active* (feedback control) systems have been proposed to accomplish roll reduction; see Burger and Corbet (1960), Lewis (1967) and Bhattacharyya (1978) for a more detailed discussion. Design methods for rudder-roll damping and fin stabilization systems are found in Perez (2005). Some passive solutions are:

Bilge Keels: Bilge keels are fins in planes approximately perpendicular to the hull or near the turn of the bilge. The longitudinal extent varies from about 25 to 50 % of the length of the ship. Bilge keels are widely used, are inexpensive but increase the hull resistance. In addition to this, they are effective mainly around the natural roll frequency of the ship. This effect significantly decreases with the speed of the ship. Bilge keels were first demonstrated in 1870.

Hull Modifications: The shape and size of the ship hull can be optimized for minimum rolling using hydrostatic and hydrodynamic criteria. This must, however, be done before the ship is built.

Anti-Rolling Tanks: The most common anti-rolling tanks in use are free-surface tanks, U-tube tanks and diversified tanks. These systems provide damping of the roll motion even at small speeds. The disadvantages are the reduction in metacenter height due to free water surface effects and that a large amount of space is required. The earliest versions were installed about the year 1874.

The most widely used systems for *active* roll damping are:

Fin Stabilizers: Fin stabilizers are highly useful devices for roll damping. They provide considerable damping if the speed of the ship is not too low. The disadvantage with additional fins is increased hull resistance and high costs associated with the installation, since at least two new hydraulic systems must be installed. Retractable fins are popular, since they are inside the hull when not in use (no additional drag). It should be noted that fins are not effective at low speed and that they cause underwater noise in addition to drag. Fin stabilizers were patented by *John I. Thornycroft* in 1889.

Rudder-Roll Damping (RRD): Roll damping by means of the rudder is relatively inexpensive compared to fin stabilizers, has approximately the same effectiveness and causes no drag or underwater noise if the system is turned off. However, RRD requires a relatively fast rudder to be effective; typically rudder rates of $\dot{\delta}_{\max} = 5\text{--}20 \text{ deg/s}$ are needed. RRD will not be effective at low ship speeds.

Gyroscopic Roll Stabilizers: Gyroscopic roll stabilizers are typically used for boats and yachts under 100 feet. The ship gyroscopic stabilizer has a spinning rotor that generates a roll stabilizing moment that counteracts the wave-induced roll motions. Unlike stabilizing fins, the ship gyroscopic stabilizer can only produce a limited roll stabilizing moment and effective systems require approximately 3 to 5 % of the craft's displacement.

For a history of ship stabilization, the interested reader is advised to consult Bennett (1991), while a detailed evaluation of different ship roll stabilization systems can be found in Sellars and Martin (1992).

Rudder-roll damping (RRD) was first suggested in the late 1970s; see Cowley and Lambert (1972, 1975), Carley (1975), Lloyd (1975) and Baitis (1980). Research in the early 1980 showed that it was indeed feasible to control the heading of a ship with at least one rudder while simultaneously using the rudder for roll damping. If only one rudder is used, this is an *underactuated control* problem. In the linear case this can be solved by *frequency separation* of the steering and roll modes since heading control can be assumed to be a low-frequency trajectory-tracking control problem while roll damping can be achieved at higher frequencies.

Before designing an RRD system the applicability of the control system in terms of effectiveness should be determined (Roberts, 1993). For a large number of ships it is in fact impossible to obtain a significant roll damping effect due to limitations of the rudder servo and the relatively large inertia of the ship.

Motivated by the results in the 1970s, RRD was tested by the US Navy by Baitis *et al.* (1983, 1989), in Sweden by Källström (1987), Källström *et al.* (1988), Källström and Schultz (1990) and Källström and Theoren (1994), and in the Netherlands by Amerongen and coauthors. Van Amerongen *et al.* (1987), Van Amerongen and Van Nauta Lempe (1987) and Van der Klugt (1987) introduced LQG theory in RRD systems. A similar approach has been proposed by Katebi *et al.* (1987), while adaptive RRD is discussed in Zhou (1990).

Blanke and co-workers have developed an RRD autopilot (Blanke *et al.*, 1989) that has been implemented by the Danish Navy on 14 ships (Munk and Blanke, 1987). Sea trials show that some of the ships had less efficient RRD systems than others. In Blanke and Christensen (1993) it was shown that the cross-couplings between steering and roll were highly sensitive to parametric variations, which again resulted in robustness problems. Different loading conditions and varying rudder shapes have been identified as reasons for this (Blanke, 1996). In Stoustrup *et al.* (1995) it has been shown that a robust RRD controller can be designed by separating the roll and steering specifications and then optimizing the two controllers independently. The coupling effects between the roll and yaw modes have also been measured in model scale and compared with full-scale trial results (Blanke and Jensen, 1997), while a new approach to identification of steering-roll models has been presented by Blanke and Tiano (1997).

More recently H_∞ control has been used to deal with model uncertainties in RRD control systems. This allows the designer to specify frequency-dependent weights for frequency separation between the steering and roll modes; see Yang and Blanke (1997, 1998). Qualitative feedback theory (QFT) has also

been applied to solve the combined RRD heading control problem under model uncertainty; see Hearn and Blanke (1998). Results from sea trials are reported in Blanke *et al.* (2000).

Simulation and full-scale experimental results of RRD systems using a multivariate autoregressive model and the minimum AIC estimate procedure have been reported by Oda *et al.* (1996, 1997). Experimental results with various control strategies are also reported by Sharif *et al.* (1996). A nonlinear RRD control system using sliding-mode control for compensation of modeling errors is reported in Lauvdal and Fossen (1997).

A gain scheduling algorithm for input rate and magnitude saturations in RRD damping systems has been developed by Lauvdal and Fossen (1998). This method is motivated by the automatic gain controller (AGC) by Van der Klugt (1987) and a technique developed for stabilization of integrator chains with input rate saturation.

In this section the focus will be on linear quadratic optimal RRD. The interested reader is recommended to consult the references above and Perez (2005) for other design techniques.

Linear Quadratic Optimal RRD Control System

Consider the 4 DOF maneuvering model (7.138) in Section 7.4:

$$\dot{\mathbf{x}} = \mathbf{Ax} + \mathbf{Bu} \quad (13.106)$$

where $\mathbf{x} = [v, p, r, \phi, \psi]^\top$ and

$$\phi = \mathbf{c}_{\text{roll}}^\top \mathbf{x}, \quad \psi = \mathbf{c}_{\text{yaw}}^\top \mathbf{x} \quad (13.107)$$

The transfer functions corresponding to (13.106) and (13.107) are

$$\frac{\phi}{\delta}(s) = \frac{b_2 s^2 + b_1 s + b_0}{s^4 + a_3 s^3 + a_2 s^2 + a_1 s + a_0} \approx \frac{K_{\text{roll}} \omega_{\text{roll}}^2 (1 + T_5 s)}{(1 + T_4 s)(s^2 + 2\zeta\omega_{\text{roll}} s + \omega_{\text{roll}}^2)} \quad (13.108)$$

$$\frac{\psi}{\delta}(s) = \frac{c_3 s^3 + c_2 s^2 + c_1 s + c_0}{s(s^4 + a_3 s^3 + a_2 s^2 + a_1 s + a_0)} \approx \frac{K_{\text{yaw}} (1 + T_3 s)}{s(1 + T_1 s)(1 + T_2 s)} \quad (13.109)$$

The control objective is a simultaneous heading control $\psi = \psi_d = \text{constant}$ and RRD ($p_d = \phi_d = 0$) using one control input. There will be a trade-off between accurate heading control (minimizing $\tilde{\psi} = \psi - \psi_d$) and control action needed to increase the natural frequency ω_{roll} and damping ratio ζ_{roll} . Also notice that it is impossible to regulate ϕ to a nonzero value while simultaneously controlling the heading angle to a nonzero value by means of a single rudder. This can easily be seen by performing a steady-state analysis of the closed-loop system. This suggests that the output of the controller should be specified as

$$\mathbf{y} = [p, r, \phi, \psi]^\top, \quad \mathbf{y}_d = [0, 0, 0, \psi_d]^\top \quad (13.110)$$

Choosing $\mathbf{y} = \mathbf{Cx}$ implies that

$$\mathbf{C} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (13.111)$$

Application of optimal control theory implies that the control objective should be specified as an optimization problem for course keeping, roll damping and minimum fuel consumption. The trade-off between these quantities can be expressed as

$$J = \min_{\mathbf{u}} \left\{ \frac{1}{2} \int_0^T (\tilde{\mathbf{y}}^\top \mathbf{Q} \tilde{\mathbf{y}} + \mathbf{u}^\top \mathbf{R} \mathbf{u}) \, d\tau \right\} \quad (13.112)$$

where $\tilde{\mathbf{y}} = \mathbf{y} - \mathbf{y}_d$, $\tilde{\mathbf{x}} = \mathbf{x} - \mathbf{x}_d$ and $\mathbf{x}_d = [0, 0, 0, 0, \psi_d]^\top$. Accurate steering is weighted against roll damping by specifying the cost matrix $\mathbf{Q} = \text{diag}\{\mathcal{Q}_p, \mathcal{Q}_r, \mathcal{Q}_\phi, \mathcal{Q}_\psi\} \geq 0$, while $\mathbf{R} = \text{diag}\{R_1, R_{21}, \dots, R_r\} > 0$ weights the use of the different rudder servos.

The solution to the LQ trajectory-tracking problem is (see Section 13.1.3)

$$\mathbf{u} = \mathbf{G}_1 \mathbf{x} + \mathbf{G}_2 \mathbf{y}_d \quad (13.113)$$

where

$$\mathbf{G}_1 = -\mathbf{R}^{-1} \mathbf{B}^\top \mathbf{P}_\infty \quad (13.114)$$

$$\mathbf{G}_2 = -\mathbf{R}^{-1} \mathbf{B}^\top (\mathbf{A} + \mathbf{B}\mathbf{G}_1)^{-\top} \mathbf{C}^\top \mathbf{Q} \quad (13.115)$$

with $\mathbf{P}_\infty = \mathbf{P}_\infty^\top > 0$ given by

$$\mathbf{P}_\infty \mathbf{A} + \mathbf{A}^\top \mathbf{P}_\infty - \mathbf{P}_\infty \mathbf{B} \mathbf{R}^{-1} \mathbf{B}^\top \mathbf{P}_\infty + \mathbf{C}^\top \mathbf{Q} \mathbf{C} = \mathbf{0} \quad (13.116)$$

Frequency Separation and Bandwidth Limitations

Since (\mathbf{A}, \mathbf{B}) is controllable and full-state feedback is applied, it is possible to move all the five poles of the system. The closed-loop system becomes

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} \\ &= \underbrace{(\mathbf{A} + \mathbf{B}\mathbf{G}_1^\top)}_{\mathbf{A}_c} \mathbf{x} + \mathbf{B}\mathbf{G}_2 \underbrace{\mathbf{h}\psi_d}_{\mathbf{y}_d} \end{aligned} \quad (13.117)$$

where

$$\mathbf{h} = [0, 0, 0, 1]^\top \quad (13.118)$$

The closed-loop transfer function in yaw is

$$\psi(s) = \mathbf{c}_{\text{yaw}}^\top (s\mathbf{I} - \mathbf{A}_c)^{-1} \mathbf{B} \mathbf{G}_2 \mathbf{h} \psi_d(s) \quad (13.119)$$

which clearly satisfy

$$\lim_{t \rightarrow \infty} \psi(t) = \psi_d \quad (13.120)$$

Notice that integral action in yaw is needed in a practical implementation of the controller. Similarly, the closed-loop roll dynamics becomes

$$\phi(s) = \mathbf{c}_{\text{roll}}^\top (s\mathbf{I} - \mathbf{A}_c)^{-1} \mathbf{B} \mathbf{G}_2 \mathbf{h} \psi_d(s) \quad (13.121)$$

If one rudder is used to control both ϕ and ψ , frequency separation is necessary to achieve this. Assume that the steering dynamics is slower than the frequency $1/T_l$ and that the natural frequency in roll is higher than $1/T_h$. Hence, the vertical reference unit (VRU) and compass measurements can be low- and high-pass filtered according to

$$\frac{\phi}{\phi_{\text{vrui}}}(s) = h_h(s) = \frac{T_h s}{1 + T_h s} \quad (13.122)$$

$$\frac{\psi}{\psi_{\text{compass}}}(s) = h_l(s) = \frac{1}{1 + T_l s} \quad (13.123)$$

It is also necessary to filter the roll and yaw rate measurements $p(s)$ and $r(s)$. These signals can also be computed by numerical differentiation of $\phi_{\text{vrui}}(s)$ and $\psi_{\text{vrui}}(s)$ using a state estimator.

This suggests that the bandwidth of the yaw angle control system must satisfy (frequency separation)

$$\omega_b \ll \omega_{\text{roll}} \quad (13.124)$$

This again implies that the low- and high-pass filters must satisfy

$$\underbrace{\omega_{\text{yaw}}}_{\substack{\text{cross-over} \\ \text{frequency}}} < \underbrace{\omega_b}_{\substack{\text{bandwidth} \\ \text{in yaw}}} < \underbrace{1/T_l}_{\substack{\text{low-pass filter} \\ \text{frequency}}} < \underbrace{1/T_h}_{\substack{\text{high-pass filter} \\ \text{frequency}}} < \underbrace{\omega_{\text{roll}}}_{\substack{\text{natural} \\ \text{frequency}}}$$

which clearly puts a restriction on the ships that can be stabilized. For many ships this requirement is impossible to satisfy due to limitations of the rudder servos and control forces.

Example 13.3 (RRD Control System Using One Rudder)

Let $\mathbf{G}_1 = [g_{11}, g_{12}, g_{13}, g_{14}, g_{15}]$ and $\mathbf{G}_2 = [0, 0, 0, g_{24}]$ such that the solution (13.113) of the SISO LQ trajectory-tracking problem can be written

$$\delta = [g_{11}, g_{12}, g_{13}, g_{14}, g_{15}] \mathbf{x} + g_{24} \psi_d \quad (13.125)$$

or

$$\delta = \underbrace{-K_v v}_{\substack{\text{sway feedback}}} \underbrace{-K_p(\psi - \psi_d) - K_d r}_{\substack{\text{PD heading controller}}} \underbrace{-K_{r1} p - K_{r2} \phi}_{\substack{\text{roll damper}}} \quad (13.126)$$

where $K_v = -g_{11}$, $K_p = -g_{15} = g_{24}$, $K_d = -g_{13}$, $K_{r1} = -g_{12}$ and $K_{r2} = -g_{14}$. Frequency separation suggests that

$$\delta = h_l(s)\delta_{\text{course}} + h_h(s)\delta_{\text{roll}} \quad (13.127)$$

where

$$\delta_{\text{course}} = -K_v v - K_p(\psi - \psi_d) - K_d r \quad (13.128)$$

$$\delta_{\text{roll}} = -K_{r1} p - K_{r2} \phi \quad (13.129)$$

The controller gains can be found by using the MSS toolbox m-function (see Section 13.1.3):

```
[G1,G2]=lqtracker(A,B,C,Q,R)
```

Alternatively, the gains can be computed by using pole placement. The two subsystems (7.145) and (7.146) with heading autopilot and RRD become (neglecting the interactions between the systems)

$$\begin{bmatrix} \dot{p} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} \underbrace{a_{22} - b_{21}K_{r1}}_{-2\zeta_{\text{roll}}\omega_{\text{roll}}} & \underbrace{(a_{24} - b_{21}K_{r2})}_{-\omega_{\text{roll}}^2} \\ 1 & 0 \end{bmatrix} \begin{bmatrix} p \\ \phi \end{bmatrix} = 0 \quad (13.130)$$

$$\begin{bmatrix} \dot{v} \\ \dot{r} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} a_{11} - b_{11}K_v & a_{13} - b_{11}K_d & -b_{11}K_p \\ a_{31} - b_{31}K_v & a_{33} - b_{31}K_d & -b_{31}K_p \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} v \\ r \\ \psi - \psi_d \end{bmatrix} = 0 \quad (13.131)$$

The poles can be specified directly in Matlab using

```
[Kr1,Kr2]=place(A_phiphi,B_phiphi,[p_phi1,p_phi2])
```

```
[Kv,Kp,Kd]=place(A_psipsi,B_psipsi,[p_psi1,p_psi2,p_psi3])
```

For roll it is seen that

$$-\omega_{\text{roll}}^2 = a_{24} - b_{21}K_{r2}, \quad -2\zeta_{\text{roll}}\omega_{\text{roll}} = a_{22} - b_{21}K_{r1} \quad (13.132)$$

or

$$K_{r1} = \frac{a_{22} + 2\zeta_{\text{roll}}\omega_{\text{roll}}}{b_{21}}, \quad K_{r2} = \frac{a_{24} + \omega_{\text{roll}}^2}{b_{21}} \quad (13.133)$$

where ζ_{roll} and ω_{roll} are pole-placement design parameters that can be used instead of eigenvalues. The model of Son and Nomoto (see ExRRD2.m in the MSS toolbox) has been used to demonstrate how an LQ optimal RRD control system can be designed. The linear state-space model for the container ship is

$$A = \begin{bmatrix} -0.0406 & -1.9614 & 0.2137 & 0.1336 & 0 \\ 0.0011 & -0.1326 & -0.1246 & -0.0331 & 0 \\ -0.0010 & 0.0147 & -0.1163 & -0.0006 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} -0.0600 \\ 0.0035 \\ 0.0026 \\ 0 \\ 0 \end{bmatrix} \quad (13.134)$$

The controller gains were computed using [G1,G2]=lqtracker(A,B,C,Q,R) with the weights

```
Q=diag([10000 1000 10 1]), R=0.5
```

resulting in

```
G1=[0.1631-16.1193-6.7655-1.1644-0.4472], G2=[0 0 0 0.4472]
```

Notice that $g_{15} = -g_{24}$. The open- and closed-loop poles are computed in Matlab by using the commands damp(A) and damp(A+B*G1); see Table 13.1.

Table 13.1 Eigenvalues, damping ratios and frequencies for the RRD control system

Eigenvalues		Damping		Frequencies (rad/s)	
Open loop	Closed loop	Open loop	Closed loop	Open loop	Closed loop
0	-0.061	—	1.00	—	0.016
-0.027	-0.026	1.00	1.00	0.027	0.026
-0.071 + 0.183 <i>i</i>	-0.100 + 0.165 <i>i</i>	0.36	0.52	0.197	0.193
-0.071 - 0.183 <i>i</i>	-0.100 - 0.165 <i>i</i>	0.36	0.52	0.197	0.193
-0.121	-0.131	1.00	1.00	0.121	0.131

It is seen that the natural frequency and relative damping ratio in roll are $\omega_{\text{roll}} = 0.193 \text{ rad/s}$ and $\zeta_{\text{roll}} = 0.36$, respectively. This is improved to $\omega_{\text{roll}} = 0.197 \text{ rad/s}$ and $\zeta_{\text{roll}} = 0.52$ by roll feedback. It is difficult to increase the relative damping ratio further due to limitations of the steering machine ($\dot{\delta}_{\text{max}} = 20 \text{ deg/s}$ and $\delta_{\text{max}} = 20 \text{ deg}$). These values can, however, be changed in RRDcontainer.m.

Since the roll frequency ω_{roll} is 0.193 rad/s and the cross-over frequency in yaw ω_{yaw} is 0.03 rad/s, see Figure 7.7 in Example 7.7, it is approximately one decade between the frequencies ω_{yaw} and ω_{roll} . Therefore, frequency separation can be obtained by choosing the low-pass and high-pass filter frequencies as $1/T_l = 0.1 \text{ rad/s}$ in yaw and $1/T_h = 0.05 \text{ rad/s}$ in roll, respectively. It is seen that the heading controller moves the poles to -0.061, -0.026 and -0.131, resulting in satisfactory course-changing capabilities (see Figure 13.5). It is also seen that the course-keeping performance is degraded during RRD. The additional yawing motion, typically 1–2 degrees in amplitude, is the price paid for adding roll feedback to an autopilot system. Also notice that the right half-plane zero in the transfer function $\phi/\delta_1(s)$ given by (7.147) is unchanged since feedback only moves the poles.

Performance Criterion for RRD

The percentage roll reduction of RRD system can be computed by using the following criterion of Oda *et al.* (1992):

$$\text{Roll reduction} = \frac{\sigma_{\text{AP}} - \sigma_{\text{RRD}}}{\sigma_{\text{AP}}} \times 100 \% \quad (13.135)$$

where

σ_{AP} = standard deviation of roll rate during course-keeping (RRD off)

σ_{RRD} = standard deviation of roll rate during course-keeping (RRD on)

For the case study in Example 13.3, $\sigma_{\text{AP}} = 0.0105$ and $\sigma_{\text{RRD}} = 0.0068$. This resulted in a roll reduction of approximately 35 % during course-keeping. For small high-speed vessels a roll reduction as high as 50–75 % can be obtained. This of course depends on the shape of the hull (hydrodynamic effects) and the capacity of the steering machine. In particular the maximum rudder rate $\dot{\delta}_{\text{max}}$ should be in the magnitude of 15–20 % to obtain good results.

Optimal Fin and RRD Systems

The most effective roll damping systems are those that combine stabilizing fins and rudders; see Källström (1981), Roberts and Braham (1990) and Perez (2005). Warship stabilization using integrated rudder and fins are discussed by Roberts (1992). More recently robust fin stabilizer controller design using the QFT

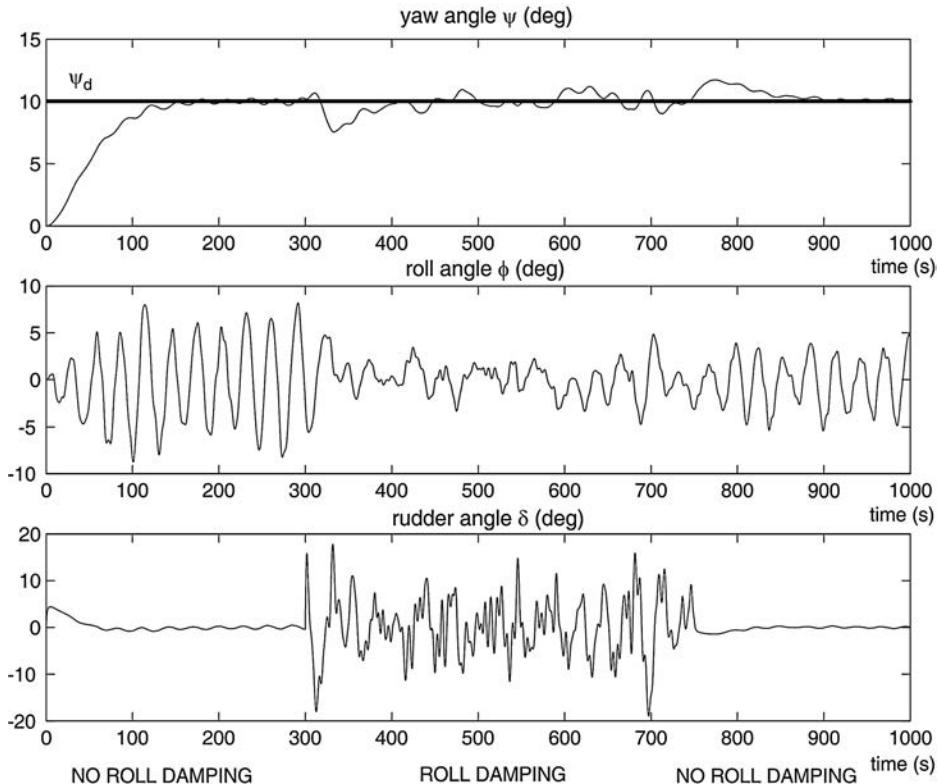


Figure 13.5 Performance of RRD control system during course-keeping and a 10° course-changing maneuver. The RRD system is active between $t = 300\text{--}700\text{ s}$.

and H_∞ design techniques have been presented by Hearns *et al.* (2000), while the performance of classical PID, optimized PID (Hickey *et al.*, 2000) and H_∞ controllers are compared in Katebi *et al.* (2000). Sea trials with the *MV Barfleur* using PID and H_∞ controllers are presented in Hickey *et al.* (1997) and experimental results with a fin and RRD control system onboard a frigate-size Royal Naval warship are reported in Sharif *et al.* (1995, 1996).

Reduction of vertical accelerations of fast ferries using fins and a T-foil is discussed by Esteban *et al.* (2000) and Giron-Sierra *et al.* (2001), while the modeling and identification results are reported in de-la-Cruz *et al.* (1998) and Aranda *et al.* (2000).

Fin stabilizers are useful for roll reduction since they are highly effective, work on a large number of ships and are more easier to control than RRD systems, even for varying load conditions and actuator configurations. Fin stabilizers are effective at high speed, but at the price of additional drag and added noise. The most economical systems are retractable fins, where additional drag is avoided during normal operation, since fin stabilizers are not needed in moderate weather. Another advantageous feature of fin stabilizing systems is that they can be used to control ϕ to a nonzero value (heel control). This is impossible with an RRD control system where the accurate control of ψ has priority.

Notice that a stand alone fin stabilization system can be constructed by simply removing the rudder inputs from the input matrix. When designing an LQ optimal fin and RRD system the following model representations can be used:

$$\mathbf{M}\dot{\mathbf{v}} + \mathbf{D}\mathbf{v} = \boldsymbol{\tau} \quad (13.136)$$

where

$$\boldsymbol{\tau} = \mathbf{T}\mathbf{f}, \quad \mathbf{f} = \mathbf{K}\mathbf{u} \quad (13.137)$$

In this representation, \mathbf{K} is the diagonal matrix of force coefficients and \mathbf{T} is the actuator configuration matrix (see Section 12.3). We can premultiply (13.136) with \mathbf{M}^{-1} to obtain

$$\dot{\mathbf{v}} = \underbrace{-\mathbf{M}^{-1}\mathbf{D}}_{\text{upper left part of } A \text{ in (7.138)}} \mathbf{v} + \underbrace{\mathbf{M}^{-1}\mathbf{T}\mathbf{K}}_{\text{upper part of } \mathbf{B} \text{ in (7.138)}} \mathbf{u} \quad (13.138)$$

In the first representation, the generalized force $\boldsymbol{\tau}$ is used as the control input while the last representation uses \mathbf{u} , that is propeller rpm, rudder angles and fin angles. In practice it is advantageous to use (13.136) instead of (13.138), since actuator failures can be handled independently by the control allocation algorithm without redesigning the control law. Notice that the \mathbf{B} matrix in (13.138) depends on \mathbf{T} and \mathbf{K} while these matrices are not used in (13.136); see Section 12.3.

Energy Optimal Criterion for Combined Fin and RRD

It is possible to derive LQ controllers for both models (13.136) and (13.138). This is demonstrated by considering a ship equipped with r_1 rudders and r_2 fins. The total number of actuators is $r = r_1 + r_2$, implying that $\mathbf{u} \in \mathbb{R}^r$. The DOFs considered are *sway*, *roll* and *yaw*; that is $n = 3$. Consequently, $\mathbf{v} = [v, p, r]^\top \in \mathbb{R}^n$. It is also assumed that the ship is fully actuated such that $r \geq n$. The generalized forces are

$$\begin{aligned} \boldsymbol{\tau} &= \mathbf{T}\mathbf{f} \\ &= \mathbf{T}\mathbf{K}\mathbf{u} \end{aligned} \quad (13.139)$$

It is advantageous to solve for the optimal control force $\boldsymbol{\tau}$ and then use control allocation to compute \mathbf{u} . For most systems the inverse (see `alloc.m` in the Matlab MSS toolbox)

$$\mathbf{u} = \mathbf{K}^{-1}\mathbf{T}_w^\dagger \boldsymbol{\tau} \quad (13.140)$$

exists. An energy optimal criterion weighting f , u or τ against accurate tracking and roll damping is

$$\begin{aligned}
 J &= \min_f \left\{ \frac{1}{2} \int_0^T (\mathbf{e}^\top \mathbf{Q} \mathbf{e} + \mathbf{f}^\top \mathbf{R}_f \mathbf{f}) d\tau \right\} \\
 &= \min_u \left\{ \frac{1}{2} \int_0^T (\mathbf{e}^\top \mathbf{Q} \mathbf{e} + \mathbf{u}^\top \underbrace{\mathbf{K}^\top \mathbf{R}_f \mathbf{K}}_{\mathbf{R}_u} \mathbf{u}) d\tau \right\} \\
 &= \min_\tau \left\{ \frac{1}{2} \int_0^T (\mathbf{e}^\top \mathbf{Q} \mathbf{e} + \boldsymbol{\tau}^\top \underbrace{(\mathbf{T}_w^\dagger)^\top \mathbf{R}_u \mathbf{T}_w}_{\mathbf{R}_\tau} \boldsymbol{\tau}) d\tau \right\}
 \end{aligned} \tag{13.141}$$

where $\mathbf{e} = \mathbf{y} - \mathbf{y}_d$. The elements in $\mathbf{Q} = \text{diag}\{Q_p, Q_r, Q_\phi, Q_\psi\} \geq 0$ are used to weight accurate steering against roll damping. The rudder and fin servos are weighted against each other by specifying the elements in $\mathbf{R}_f = \text{diag}\{R_{\delta 1}, R_{\delta 2}, \dots, R_{\delta r_1}, R_{f1}, R_{f2}, \dots, R_{fr_2}\} > 0$. If $r_1 = 0$ and $R_{\delta 1} = R_{\delta 2} = \dots = R_{\delta r_1} = 0$ only fin stabilization is obtained (no rudders).

The control weights satisfy

$$\mathbf{R}_u = \mathbf{K}^\top \mathbf{R}_f \mathbf{K}, \quad \mathbf{R}_\tau = (\mathbf{T}_w^\dagger)^\top \mathbf{R}_f \mathbf{T}_w^\dagger \tag{13.142}$$

The solution to the LQ problem (13.141) with $\boldsymbol{\tau}$ as the control variable is (see Section 13.1.3)

$$\boldsymbol{\tau} = \mathbf{G}_1 \mathbf{x} + \mathbf{G}_2 \mathbf{y}_d \tag{13.143}$$

$$\mathbf{G}_1 = -[(\mathbf{T}_w^\dagger)^\top \mathbf{R}_f \mathbf{T}_w^\dagger]^{-1} \mathbf{B}^\top \mathbf{P}_\infty \tag{13.144}$$

$$\mathbf{G}_2 = -[(\mathbf{T}_w^\dagger)^\top \mathbf{R}_f \mathbf{T}_w^\dagger]^{-1} \mathbf{B}^\top (\mathbf{A} + \mathbf{B}\mathbf{G}_1)^{-1} \mathbf{C}^\top \mathbf{Q} \tag{13.145}$$

where \mathbf{Q} and \mathbf{R}_f are design matrices while $\mathbf{P}_\infty = \mathbf{P}_\infty^\top > 0$ is given by

$$\mathbf{P}_\infty \mathbf{A} + \mathbf{A}^\top \mathbf{P}_\infty - \mathbf{P}_\infty \mathbf{B} [(\mathbf{T}_w^\dagger)^\top \mathbf{R}_f \mathbf{T}_w^\dagger]^{-1} \mathbf{B}^\top \mathbf{P}_\infty + \mathbf{C}^\top \mathbf{Q} \mathbf{C} = \mathbf{0} \tag{13.146}$$

Operability and Motion Sickness Incidence Criteria

Operability criteria for manual and intellectual work as well as motion sickness are important design criteria for the evaluation of autopilot and roll damping systems. Sea-sickness is especially important in high-speed craft and ships with high vertical accelerations.

Human Operability Limiting Criteria in Roll: Operability limiting criteria with regard to vertical and lateral accelerations, and roll angle for the effectiveness of the crew and the passengers are given in

Table 13.2 Criteria for effectiveness of the crew (Faltinsen, 1990)

Standard deviation (root mean square) criteria			
Vertical acceleration (\dot{w})	Lateral acceleration (\dot{v})	Roll angle (ϕ)	Description of work
0.20 g	0.10 g	6.0 deg	Light manual work
0.15 g	0.07 g	4.0 deg	Heavy manual work
0.10 g	0.05 g	3.0 deg	Intellectual work
0.05 g	0.04 g	2.5 deg	Transit passengers
0.02 g	0.03 g	2.0 deg	Cruise liner

Table 13.2. This gives an indication on what type of work that can be expected to be carried out for different roll angles/sea states.

ISO 2631-3:1985 Criterion for Motion Sickness Incidence: In addition to operability, limiting criteria passenger comfort can be evaluated with respect to motion sickness. The International Organization for Standardization (ISO) motion seasickness incidence criterion is reported in ISO 2631-1 (1997). This report replaces ISO 2631-3 (1985); see <http://www.iso.ch>. The most important factors for seasickness are vertical (heave) accelerations a_z (m/s^2), exposure time t (hours) and encounter frequency ω_e (rad/s). The ISO standard criterion for MSI proposes an MSI of 10 %, which means that 10 % of the passengers become seasick during t hours. The MSI curves as a function of exposure time are shown in Figure 13.6, where

$$a_z(t, \omega_e) = \begin{cases} 0.5\sqrt{2/t} & \text{for } 0.1 \text{ Hz} < \frac{\omega_e}{2\pi} \leq 0.315 \text{ Hz} \\ 0.5\sqrt{2/t} \cdot 6.8837 \left(\frac{\omega_e}{2\pi}\right)^{1.67} & \text{for } 0.315 \text{ Hz} \leq \frac{\omega_e}{2\pi} \leq 0.63 \text{ Hz} \end{cases} \quad (13.147)$$

Matlab

The MSI curves (13.147) as functions of the exposure time are implemented in the Matlab MSS toolbox as

```
[a_z,w_e] = ISOmsi(t)
```

Figure 13.6 is generated by using the example file

```
ExMSI
```

The main limitation of the ISO criterion is that it only predicts the exceedence of the 10 % MSI point. It is also assumed that the accelerations in the CG are representative for the entire ship and that a representative wave period can be used instead of the actual wave. In many cases it is advantageous to use the extended sickness method for more accurate predictions. This method is presented below.

Probability Integral Method for MSI: The O'Hanlon and McCauley (1974) probability integral method is convenient to use since it produces an MSI criterion in percentage for combinations of heave acceleration a_z (m/s^2) and frequency of encounter ω_e (rad/s). The MSI index is defined as the

number of sea sick people in percentage for an exposure time of two hours; see Lloyd (1989) and Lewis (1989). The criterion is as follows:

$$\text{MSI} = 100 \left[0.5 \pm \text{erf} \left(\frac{\pm \log_{10}(a_z/g) \mp \mu_{\text{MSI}}}{0.4} \right) \right] \quad (\%) \quad (13.148)$$

where

$$\mu_{\text{MSI}} = -0.819 + 2.32 (\log_{10} \omega_e)^2 \quad (13.149)$$

and

$$\text{erf}(x) = \text{erf}(-x) = \frac{1}{\sqrt{2\pi}} \int_0^x \exp \left(-\frac{z^2}{2} \right) dz \quad (13.150)$$

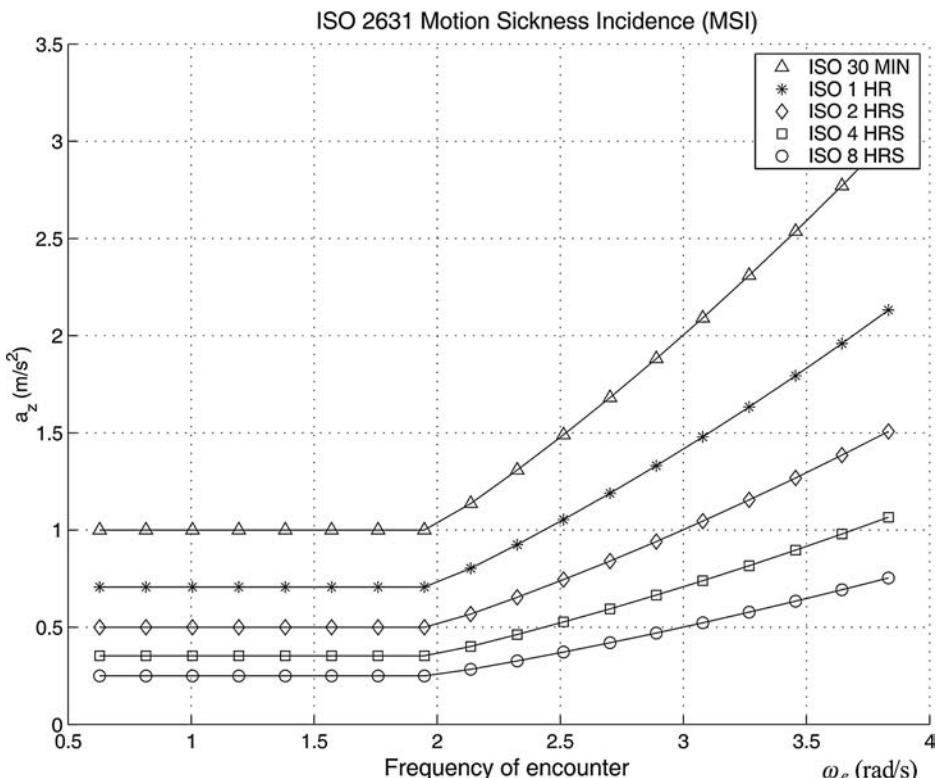


Figure 13.6 Heave acceleration a_z (m/s^2) as a function of frequency of encounter ω_e (rad/s) for different exposure times. The ISO curves represent an MSI of 10 %.

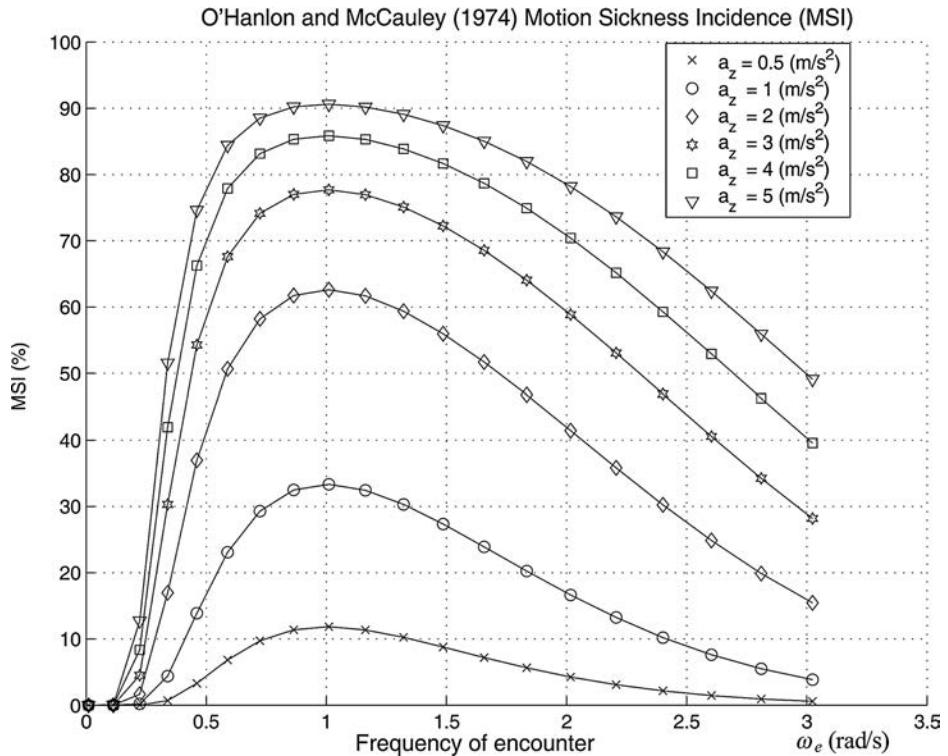


Figure 13.7 MSI is the number of motion sick persons in percentage during a two hour exposure time as a function of encounter frequency ω_e (rad/s) and heave acceleration a_z (m/s^2).

Matlab

The Matlab MSS toolbox function

```
msi = HMmsi(a_z,w_e)
```

can be used for computation of the MSI. Notice that the erf function in `HMmsi.m` is scaled differently from the Matlab function `erf.m`. The MSI curves in Figure 13.7 are plotted for different a_z and ω_e using the example file

`ExMSI`

The major drawback of the O'Hanlon and McCauley method is that it only applies to a two hour exposure time. Another effect to take into account is that the O'Hanlon and McCauley MSI criterion is derived from tests with young men seated separately in insulated cabins. According to ISO 2631-1, the MSI number is about 1.5 higher among women and children, suggesting that the actual MSI number for passengers of average age and sex distribution should be at least 1.25 times higher.

13.1.6 Case Study: Optimal Dynamic Positioning System for Ships and Floating Structures

In Section 12.2.10 a nonlinear PID controller was designed for DP and the equilibrium point was rendered asymptotically stable under the assumption of full-state feedback. Output feedback in terms of a nonlinear passive observer was also discussed and UGAS of the resulting system was relying on a nonlinear separation principle (Loria *et al.*, 2000). An alternative to the nonlinear PID controller is to formulate the problem as a linear optimal control problem using vessel parallel coordinates. The LQ controller will be designed under the assumption that all states can be measured. This assumption can, however, be relaxed by combining the LQ controller with a Kalman filter for optimal state estimation; see Section 11.3.6. The resulting control law is known as the LQG optimal controller, and convergence and stability of the interconnected system can be proven using a linear separation principle (Gelb *et al.*, 1988).

Controller Model: Recall from Section 7.3.2 that

$$\dot{\eta}_p = v \quad (13.151)$$

$$M\dot{v} + Dv = b_p + \tau + \tau_{wind} + \tau_{wave} \quad (13.152)$$

where VP coordinates have been employed (see Section 7.5.3). The North-East positions and heading are related to η_p according to

$$\eta = R(\psi)\eta_p \quad (13.153)$$

In order to incorporate the limitations of the propellers, the model is augmented by actuator dynamics. The simplest way of doing this is to define three time constants in *surge*, *sway* and *yaw* such that

$$\dot{\tau} = A_{thr}(\tau - \tau_{com}) \quad (13.154)$$

where τ_{com} is the commanded thrust and $A_{thr} = -\text{diag}\{1/T_{surge}, 1/T_{sway}, 1/T_{yaw}\}$ is a diagonal matrix containing the time constants. The resulting state-space model becomes

$$\dot{x}_c = Ax_c + B\tau_{com} \quad (13.155)$$

where the controller states are $x_c := [\eta_p^\top, v^\top, \tau^\top]^\top$ and

$$A = \begin{bmatrix} 0 & I & 0 \\ 0 & -M^{-1}D & M^{-1} \\ 0 & 0 & A_{thr} \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ 0 \\ -A_{thr} \end{bmatrix} \quad (13.156)$$

This model is the basis for the LQ controller.

Observer Model: The Kalman filter can be designed using only position and heading measurements. For this purpose the filter states are chosen as $x_f := [\eta_p^\top, b_p^\top, v_p^\top]^\top$. The WF model is omitted for simplicity but in an industrial system six more states should be added following the approach in

Section 11.3.6, for instance. The filter model takes the following form:

$$\dot{x}_f = Fx_f + G\tau + Ew \quad (13.157)$$

$$z = Hx_f + v \quad (13.158)$$

where

$$F = \begin{bmatrix} \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & I_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & M^{-1} & -M^{-1}D \end{bmatrix}, \quad G = \begin{bmatrix} \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} \\ M^{-1} \end{bmatrix}, \quad H = [I_{3 \times 3}, \mathbf{0}_{3 \times 3}, \mathbf{0}_{3 \times 3}] \quad (13.159)$$

Controllability and Observability

It is important to be aware that the controller model must be controllable and the observer model must be observable to guarantee a stable solution for the LQG controller. These conditions can easily be verified in Matlab by considering the following example:

Matlab

The following example demonstrates how observability and controllability can be checked for a ship in surge, sway and yaw.

Example 13.4 (Observability and Controllability of Ships)

Consider a supply vessel with nondimensional system matrices (Fossen et al., 1996):

$$M'' = \begin{bmatrix} 1.1274 & 0 & 0 \\ 0 & 1.8902 & -0.0744 \\ 0 & -0.0744 & 0.1278 \end{bmatrix}, D'' = \begin{bmatrix} 0.0358 & 0 & 0 \\ 0 & 0.1183 & -0.0124 \\ 0 & -0.0041 & 0.0308 \end{bmatrix} \quad (13.160)$$

These values are defined in accordance to the bis system (see Section 7.2.5) such that

$$M = mT^{-2}(TM''T^{-1}), \quad D = m\sqrt{g/L} T^{-2}(TD''T^{-1}) \quad (13.161)$$

where $T = \text{diag}\{1, 1, L\}$. Assume that $A_{\text{thr}} = -1/100 \times I_{3 \times 3}$. The linear state-space model in surge, sway and yaw is computed as

$$A = \begin{bmatrix} \mathbf{0}_{3 \times 3} & I_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & -M^{-1}D & M^{-1} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & A_{\text{thr}} \end{bmatrix}, \quad B = \begin{bmatrix} \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} \\ -A_{\text{thr}} \end{bmatrix} \quad (13.162)$$

$$F = \begin{bmatrix} \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & I_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & M^{-1} & -M^{-1}D \end{bmatrix}, \quad H = [I_{3 \times 3}, \mathbf{0}_{3 \times 3}, \mathbf{0}_{3 \times 3}]$$

Notice that only the positions (N , E) and yaw angle ψ are defined as the outputs for the observer. Observability and controllability can be checked in Matlab using the commands (see `ExObsCtr.m`):

```
n_obs = rank(obsv(F, H))
n_ctr = rank(ctrb(A, B))
```

Since $n_{\text{obs}} = n_{\text{ctr}} = 9$ the supply vessel is both observable and controllable.

Since the supply vessel is controllable, it is straightforward to design an optimal control law with wind feedforward and integral action. In order to do this, it is convenient to split the control input into two parts:

$$\boldsymbol{\tau}_{\text{com}} = \boldsymbol{\tau}_{\text{LQ}} - \hat{\boldsymbol{\tau}}_{\text{wind}} \quad (13.163)$$

where $\boldsymbol{\tau}_{\text{LQ}}$ is the optimal feedback and $\hat{\boldsymbol{\tau}}_{\text{wind}}$ is an estimate of the generalized wind forces that can be implemented using (12.214).

Optimal Feedback Control

The LQ control objective is to obtain $\boldsymbol{x} = \mathbf{0}$ such that $\boldsymbol{\eta}_p = \boldsymbol{v} = \boldsymbol{\tau} = \mathbf{0}$. This is achieved by minimizing the performance index:

$$J = \min_{\boldsymbol{\tau}_{\text{LQ}}} \left\{ \frac{1}{2} \int_0^T (\boldsymbol{x}^\top \boldsymbol{Q} \boldsymbol{x} + \boldsymbol{\tau}_{\text{LQ}}^\top \boldsymbol{R} \boldsymbol{\tau}_{\text{LQ}}) d\tau \right\} \quad (13.164)$$

where $\boldsymbol{R} = \boldsymbol{R}^\top > 0$ and $\boldsymbol{Q} = \boldsymbol{Q}^\top \geq 0$ are two cost matrices to be specified by the user. The \boldsymbol{Q} matrix is defined as $\boldsymbol{Q} := \text{diag}\{\boldsymbol{Q}_1, \boldsymbol{Q}_2, \boldsymbol{Q}_3\}$ where the weights \boldsymbol{Q}_1 , \boldsymbol{Q}_2 and \boldsymbol{Q}_3 put penalty on position and heading $\boldsymbol{\eta}_p$, velocity \boldsymbol{v} and actuator dynamics $\boldsymbol{\tau}$, respectively. The optimal control law minimizing (13.164) is (see Section 13.1.1)

$$\boldsymbol{\tau}_{\text{LQ}} = \underbrace{-\boldsymbol{R}^{-1} \boldsymbol{B}^\top \boldsymbol{P}_\infty}_{G} \boldsymbol{x} \quad (13.165)$$

where \boldsymbol{P}_∞ is the solution of the ARE:

$$\boldsymbol{P}_\infty \boldsymbol{A} + \boldsymbol{A}^\top \boldsymbol{P}_\infty - \boldsymbol{P}_\infty \boldsymbol{B} \boldsymbol{R}^{-1} \boldsymbol{B}^\top \boldsymbol{P}_\infty + \boldsymbol{Q} = \mathbf{0} \quad (13.166)$$

Integral Action

In order to obtain zero steady-state errors in *surge*, *sway* and *yaw*, integral action must be included in the control law. Integral action can be obtained by using state augmentation. Since we want the three outputs (N, E, ψ) to be regulated to zero, no more than three integral states can be augmented to the system. Define a new state variable:

$$z := \int_0^t y(\tau) d\tau \implies \dot{z} = y \quad (13.167)$$

Here y is a subspace of \boldsymbol{x} given by

$$\boldsymbol{y} = \boldsymbol{C} \boldsymbol{x} \quad (13.168)$$

with

$$\boldsymbol{C} = \begin{bmatrix} \boldsymbol{I}_{3 \times 3} & \mathbf{0}_{3 \times 3} & \mathbf{0}_{3 \times 3} \end{bmatrix} \quad (13.169)$$

Next consider an augmented model with state vector $\mathbf{x}_a := [\mathbf{z}^\top, \mathbf{x}^\top]^\top$ such that

$$\dot{\mathbf{x}}_a = \mathbf{A}_a \mathbf{x}_a + \mathbf{B}_a \boldsymbol{\tau}_{\text{com}} \quad (13.170)$$

where

$$\mathbf{A}_a = \begin{bmatrix} \mathbf{0}_{3 \times 3} & \mathbf{C} \\ \mathbf{0}_{9 \times 3} & \mathbf{A} \end{bmatrix}, \quad \mathbf{B}_a = \begin{bmatrix} \mathbf{0}_{3 \times 3} \\ \mathbf{B} \end{bmatrix} \quad (13.171)$$

Matlab

Controllability of the augmented system $(\mathbf{A}_a, \mathbf{B}_b)$ is checked in Matlab by using the command (see ExObsCtr.m):

```
n_ctr=rank(ctrlb(Aa,Ba))
```

which gives $n_ctr = 12$. Hence, the supply vessel with additional states for integral action is controllable.

The performance index for the integral controller becomes

$$J = \min_{\boldsymbol{\tau}_{\text{LQ}}} \left\{ \frac{1}{2} \int_0^T (\mathbf{x}_a^\top \mathbf{Q}_a \mathbf{x}_a + \boldsymbol{\tau}_{\text{LQ}}^\top \mathbf{R} \boldsymbol{\tau}_{\text{LQ}}) d\tau \right\} \quad (13.172)$$

where $\mathbf{R} = \mathbf{R}^\top > 0$ and

$$\mathbf{Q}_a = \begin{bmatrix} \mathbf{Q}_I & \mathbf{0} \\ \mathbf{0} & \mathbf{Q} \end{bmatrix} \geq 0 \quad (13.173)$$

The matrix $\mathbf{Q}_I = \mathbf{Q}_I^\top > 0$ is used to specify the integral times in *surge*, *sway* and *yaw*. The optimal PID controller is (see Section 13.1.1)

$$\boldsymbol{\tau}_{\text{LQ}} = \mathbf{G}_a \mathbf{x}_a = \mathbf{G} \mathbf{x} + \mathbf{G}_I \underbrace{\int_0^t y(\tau) d\tau}_z \quad (13.174)$$

where $\mathbf{G}_a = [\mathbf{G}_I, \mathbf{G}]$ and

$$\mathbf{G}_a = -\mathbf{R}^{-1} \mathbf{B}_a^\top \mathbf{P}_\infty \quad (13.175)$$

$$\mathbf{P}_\infty \mathbf{A}_a + \mathbf{A}_a^\top \mathbf{P}_\infty - \mathbf{P}_\infty \mathbf{B}_a \mathbf{R}^{-1} \mathbf{B}_a^\top \mathbf{P}_\infty + \mathbf{Q}_a = \mathbf{0} \quad (13.176)$$

LQG Control–Linear Separation Principle

In practice only some of the states are measured. A minimum requirement is that the position and heading of the craft is measured such that velocities and bias terms can be estimated by an observer. This is usually done under the assumption that the states \mathbf{x} can be replaced with the estimated states $\hat{\mathbf{x}}$ such that the optimal integral controller (13.174) can be modified as

$$\tau_{LQ} = \mathbf{G}\hat{\mathbf{x}} + \mathbf{G}_I \mathbf{C} \int_0^t \hat{\mathbf{x}}(\tau) d\tau \quad (13.177)$$

where the state estimate $\hat{\mathbf{x}}$ can be computed using

- Kalman filter (Section 11.3.6)
- Nonlinear passive observer (Section 11.4.1)

For the Kalman filter in cascade with the LQ controller there exists a *linear separation principle* guaranteeing that $\hat{\mathbf{x}} \rightarrow \mathbf{x}$ and that $\mathbf{x} \rightarrow \mathbf{0}$ (Athans and Falb, 1966). This is referred to as LQG control and it was first applied to design DP systems by Balchen *et al.* (1976, 1980a, 1980b) and Grimble *et al.* (1980a, 1980b). Optimal DP systems are used to maintain the position of offshore drilling and supply vessels (see Figure 13.8).



Figure 13.8 Oil production using a dynamically positioned semi-submersible. Illustration by Bjarne Stenberg/MARINTEK.

13.2 State Feedback Linearization

The basic idea with feedback linearization is to transform the nonlinear system dynamics into a linear system (Freund, 1973). Feedback linearization is discussed in more detail by Isidori (1989) and Slotine and Li (1991). Conventional control techniques such as pole-placement and linear quadratic optimal control theory can then be applied to the linear system. In robotics, this technique is commonly referred to as *computed torque* control (Sciavicco and Siciliano, 1996).

Feedback linearization is easily applicable to ships and underwater vehicles since these models basically are nonlinear *mass-damper-spring* systems, which can be transformed into a linear system by using a nonlinear mapping. Transformations that can be used for applications both in BODY and NED coordinates will be presented. Trajectory-tracking control in the BODY frame is used for velocity control while NED frame applications are recognized as position and attitude control. Combined position and velocity control systems will also be discussed.

13.2.1 Decoupling in the BODY Frame (Velocity Control)

The control objective is to transform the marine craft dynamics into a linear system:

$$\dot{v} = a^b \quad (13.178)$$

where a^b can be interpreted as a body-fixed *commanded acceleration* vector. The body-fixed vector representation should be used to control the linear and angular velocities. Consider the nonlinear marine craft dynamics in the form

$$M\dot{v} + n(v, \eta) = \tau \quad (13.179)$$

where η and v are assumed to be measured and n is the nonlinear vector

$$n(v, \eta) = C(v)v + D(v)\eta + g(\eta) \quad (13.180)$$

The nonlinearities can be canceled out by simply selecting the control law as (see Figure 13.9)

$$\tau = Ma^b + n(v, \eta) \quad (13.181)$$

where the commanded acceleration vector a^b can be chosen by, for instance, pole placement or linear quadratic optimal control theory. However, note that to investigate optimality of the original system, the optimal control and cost function must be transformed back through the nonlinear mapping.

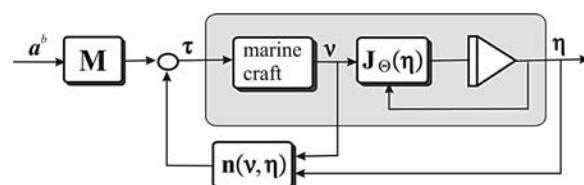


Figure 13.9 Nonlinear decoupling in the BODY frame.

Pole Placement

Let $\Lambda > 0$ be a diagonal design matrix

$$\Lambda = \text{diag}\{\lambda_1, \lambda_2, \dots, \lambda_n\}$$

used to specify the desired control bandwidth, \mathbf{v}_d the desired linear and angular velocity vector and $\tilde{\mathbf{v}} = \mathbf{v} - \mathbf{v}_d$ the velocity tracking error. Then the commanded acceleration vector can be chosen as a PI controller with acceleration feedforward:

$$\mathbf{a}^b = \dot{\mathbf{v}}_d - \mathbf{K}_p \tilde{\mathbf{v}} - \mathbf{K}_i \int_0^t \tilde{\mathbf{v}}(\tau) d\tau \quad (13.182)$$

Choosing the gains as

$$\mathbf{K}_p = 2\Lambda, \quad \mathbf{K}_i = \Lambda^2$$

yields the second-order error dynamics

$$\mathbf{M}(\ddot{\tilde{\mathbf{v}}} - \mathbf{a}^b) = \mathbf{M} \left(\ddot{\tilde{\mathbf{v}}} + 2\Lambda \tilde{\mathbf{v}} + \Lambda^2 \int_0^t \tilde{\mathbf{v}}(\tau) d\tau \right) = \mathbf{0} \quad (13.183)$$

This implies that for each DOF both poles are in $s = -\lambda_i$ ($i = 1, \dots, n$). Consequently,

$$(s + \lambda_i)^2 \int_0^t \tilde{\mathbf{v}}(\tau) d\tau = 0 \quad (i = 1, \dots, n) \quad (13.184)$$

The reference model of Section 10.2.1 can be used to generate a smooth velocity trajectory \mathbf{v}_d for trajectory-tracking control.

13.2.2 Decoupling in the NED Frame (Position and Attitude Control)

For position and attitude control the dynamics are decoupled in the NED reference frame. Consider

$$\ddot{\boldsymbol{\eta}} = \mathbf{a}^n \quad (13.185)$$

where \mathbf{a}^n can be interpreted as the commanded acceleration in NED. Consider the kinematic and kinetic equations in the form

$$\dot{\boldsymbol{\eta}} = \mathbf{J}_{\Theta}(\boldsymbol{\eta})\mathbf{v} \quad (13.186)$$

$$\mathbf{M}\dot{\mathbf{v}} + \mathbf{n}(\mathbf{v}, \boldsymbol{\eta}) = \boldsymbol{\tau} \quad (13.187)$$

where both $\boldsymbol{\eta}$ and \mathbf{v} are assumed measured. Differentiation of the kinematic equation (13.186) with respect to time yields

$$\dot{\mathbf{v}} = \mathbf{J}_{\Theta}^{-1}(\boldsymbol{\eta})[\ddot{\boldsymbol{\eta}} - \dot{\mathbf{J}}_{\Theta}(\boldsymbol{\eta})\mathbf{v}] \quad (13.188)$$

The nonlinear control law

$$\tau = \mathbf{M}\mathbf{a}^b + \mathbf{n}(\mathbf{v}, \boldsymbol{\eta}) \quad (13.189)$$

applied to (13.187) yields

$$\mathbf{M}(\ddot{\mathbf{v}} - \mathbf{a}^b) = \mathbf{M}\mathbf{J}_\Theta^{-1}(\boldsymbol{\eta})[\ddot{\boldsymbol{\eta}} - \dot{\mathbf{J}}_\Theta(\boldsymbol{\eta})\mathbf{v} - \mathbf{J}_\Theta(\boldsymbol{\eta})\mathbf{a}^b] = \mathbf{0} \quad (13.190)$$

Choosing

$$\mathbf{a}^n = \dot{\mathbf{J}}_\Theta(\boldsymbol{\eta})\mathbf{v} + \mathbf{J}_\Theta(\boldsymbol{\eta})\mathbf{a}^b \quad (13.191)$$

yields the linear decoupled system

$$\mathbf{M}^*(\ddot{\boldsymbol{\eta}} - \mathbf{a}^n) = \mathbf{0} \quad (13.192)$$

where $\mathbf{M}^* = \mathbf{J}_\Theta^\top(\boldsymbol{\eta})\mathbf{M}\mathbf{J}_\Theta^{-1}(\boldsymbol{\eta}) > 0$. From (13.191) it is seen that

$$\mathbf{a}^b = \mathbf{J}_\Theta^{-1}(\boldsymbol{\eta})[\mathbf{a}^n - \dot{\mathbf{J}}_\Theta(\boldsymbol{\eta})\mathbf{v}] \quad (13.193)$$

where the commanded acceleration \mathbf{a}^n can be chosen as a PID control law with acceleration feedforward:

$$\mathbf{a}^n = \ddot{\boldsymbol{\eta}}_d - \mathbf{K}_d \dot{\tilde{\boldsymbol{\eta}}} - \mathbf{K}_p \tilde{\boldsymbol{\eta}} - \mathbf{K}_i \int_0^t \tilde{\boldsymbol{\eta}}(\tau) d\tau \quad (13.194)$$

where \mathbf{K}_p , \mathbf{K}_d and \mathbf{K}_i are positive definite matrices chosen such that the error dynamics

$$\ddot{\tilde{\boldsymbol{\eta}}} + \mathbf{K}_d \dot{\tilde{\boldsymbol{\eta}}} + \mathbf{K}_p \tilde{\boldsymbol{\eta}} + \mathbf{K}_i \int_0^t \tilde{\boldsymbol{\eta}}(\tau) d\tau = \mathbf{0} \quad (13.195)$$

is GES. One simple pole-placement algorithm for PID control is

$$(s + \lambda_i)^3 \int_0^t \tilde{\boldsymbol{\eta}}(\tau) d\tau = \mathbf{0} \quad (i = 1, \dots, n) \quad (13.196)$$

which yields

$$\mathbf{K}_d = 3\mathbf{\Lambda} = \text{diag}\{3\lambda_1, 3\lambda_2, \dots, 3\lambda_n\}$$

$$\mathbf{K}_p = 3\mathbf{\Lambda}^2 = \text{diag}\{3\lambda_1^2, 3\lambda_2^2, \dots, 3\lambda_n^2\}$$

$$\mathbf{K}_i = \mathbf{\Lambda}^3 = \text{diag}\{\lambda_1^3, \lambda_2^3, \dots, \lambda_n^3\}$$

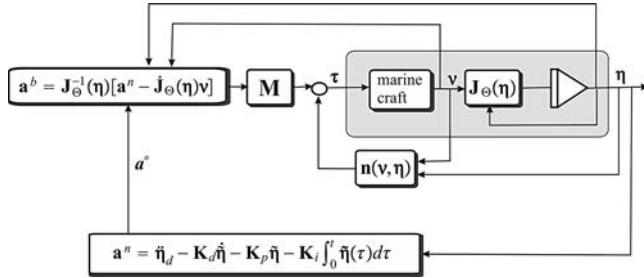


Figure 13.10 Nonlinear decoupling in the NED frame with transformation to the BODY frame.

This is shown in Figure 13.10. When implementing the trajectory-tracking controller a third-order reference model can be used to compute smooth position and attitude trajectories η_d (see Section 10.2.1).

13.2.3 Case Study: Feedback Linearizing Speed Controller for Ships and Underwater Vehicles

Consider the following decoupled model of a ship in surge:

$$m\ddot{u} + d_1 u + d_2 |u|u = \tau \quad (13.197)$$

From Section 13.2.1 it follows that the commanded acceleration can be calculated as

$$a^b = \dot{u}_d - K_p(u - u_d) - K_i \int_0^t (u - u_d) d\tau \quad (13.198)$$

while the speed controller takes the following form:

$$\tau = m[\dot{u}_d - K_p(u - u_d) - K_i \int_0^t (u - u_d) d\tau] + d_1 u + d_2 |u|u \quad (13.199)$$

Hence, the equilibrium point of the linear system

$$\ddot{u} + K_p \ddot{u} + K_i \int_0^t \ddot{u}(\tau) d\tau = 0 \quad (13.200)$$

is GES if the gains are chosen as

$$K_p = 2\lambda \quad (13.201)$$

$$K_i = \lambda^2 \quad (13.202)$$

with $\lambda > 0$. In order to implement the speed controller, the following reference model can be used (see Section 10.2.1):

$$\ddot{u}_d + 2\xi\omega\dot{u}_d + \omega^2 u_d = \omega^2 r^b \quad (13.203)$$

where $\xi > 0$ and $\omega > 0$ are the reference model damping ratio and natural frequency while r^b is the setpoint specifying the desired surge speed.

13.2.4 Case Study: Feedback Linearizing Ship and Underwater Vehicle Autopilot

Consider the nonlinear model (Norrbom, 1963):

$$\dot{\psi} = r \quad (13.204)$$

$$m\ddot{r} + d_1\dot{r} + d_2|r|r = \tau \quad (13.205)$$

where ψ is the yaw angle. Hence, the commanded acceleration can be calculated as (Fossen and Paulsen, 1992)

$$a^n = \dot{r}_d - K_d(r - r_d) - K_p(\psi - \psi_d) - K_i \int_0^t (\psi - \psi_d) d\tau \quad (13.206)$$

where r_d is the desired yaw rate and ψ_d is the desired heading angle. For this particular example, (13.193) implies that $a^n = a^b$. Choosing the decoupling control law as

$$\tau = m \left[\dot{r}_d - K_d(r - r_d) - K_p(\psi - \psi_d) - K_i \int_0^t (\psi - \psi_d) d\tau \right] + d_1\dot{r} + d_2|r|r \quad (13.207)$$

finally gives the error dynamics

$$\dot{\tilde{\psi}} = \tilde{r} \quad (13.208)$$

$$\dot{\tilde{r}} + K_d\tilde{r} + K_p\tilde{\psi} = 0 \quad (13.209)$$

The reference model can be chosen as (see Section 10.2.1)

$$\dot{\psi}_d^{(3)} + (2\zeta + 1)\omega\ddot{\psi}_d + (2\zeta + 1)\omega^2\dot{\psi}_d + \omega^3\psi_d = \omega^3r^n \quad (13.210)$$

Notice that (13.207) depends on the uncertain parameters d_1 and d_2 while m is quite easy to estimate using hydrodynamic programs. Hence, care must be taken when implementing (13.207). For most craft, the control law (13.207) works very well even with $d_1 = d_2 = 0$ so the need for choosing nonzero damping parameters should be seen as a trade-off between robustness and performance.

13.2.5 Case Study: MIMO Adaptive Feedback Linearizing Controller for Ships and Underwater Vehicles

So far only feedback linearization has been discussed under the assumption that all model parameters are *known*. This can be relaxed by using *parameter adaptation*. Consider a marine craft given by the nonlinear system

$$\dot{\eta} = J_{\Theta}(\eta)v \quad (13.211)$$

$$M\dot{v} + n(v, \eta) = \tau \quad (13.212)$$

Taking the control law to be

$$\tau = \hat{M}\hat{a}^b + \hat{n}(v, \eta) \quad (13.213)$$

where the *hat* denotes the adaptive parameter estimates, yields the error dynamics

$$\mathbf{M}[\dot{\mathbf{v}} - \mathbf{a}^b] = [\hat{\mathbf{M}} - \mathbf{M}]\mathbf{a}^b + [\hat{\mathbf{n}}(\mathbf{v}, \boldsymbol{\eta}) - \mathbf{n}(\mathbf{v}, \boldsymbol{\eta})] \quad (13.214)$$

If the equations of motion are linear in a parameter vector $\boldsymbol{\theta}$, the following parametrization can be applied:

$$[\hat{\mathbf{M}} - \mathbf{M}]\mathbf{a}^b + [\hat{\mathbf{n}}(\mathbf{v}, \boldsymbol{\eta}) - \mathbf{n}(\mathbf{v}, \boldsymbol{\eta})] = \Phi(\mathbf{a}^b, \mathbf{v}, \boldsymbol{\eta})\tilde{\boldsymbol{\theta}} \quad (13.215)$$

Here $\tilde{\boldsymbol{\theta}} = \hat{\boldsymbol{\theta}} - \boldsymbol{\theta}$ is the *unknown* parameter error vector and $\Phi(\mathbf{a}^b, \mathbf{v}, \boldsymbol{\eta})$ is a *known* matrix function of measured signals usually referred to as the *regressor matrix*. Using the result $\mathbf{a}^n = \dot{\mathbf{J}}_\Theta(\boldsymbol{\eta})\mathbf{v} + \mathbf{J}_\Theta(\boldsymbol{\eta})\mathbf{a}^b$ from (13.191) gives

$$\mathbf{M}\mathbf{J}_\Theta^{-1}(\boldsymbol{\eta})[\ddot{\boldsymbol{\eta}} - \mathbf{a}^n] = \Phi(\mathbf{a}^b, \mathbf{v}, \boldsymbol{\eta})\tilde{\boldsymbol{\theta}} \quad (13.216)$$

Premultiplying this expression by $\mathbf{J}_\Theta^{-T}(\boldsymbol{\eta})$ and letting $\mathbf{M}^*(\boldsymbol{\eta}) = \mathbf{J}_\Theta^{-T}(\boldsymbol{\eta})\mathbf{M}\mathbf{J}_\Theta^{-1}(\boldsymbol{\eta})$ yields the error dynamics

$$\mathbf{M}^*(\boldsymbol{\eta})[\ddot{\boldsymbol{\eta}} - \mathbf{a}^n] = \mathbf{J}_\Theta^{-T}(\boldsymbol{\eta})\Phi(\mathbf{a}^b, \mathbf{v}, \boldsymbol{\eta})\tilde{\boldsymbol{\theta}} \quad (13.217)$$

Furthermore, let the commanded acceleration be chosen as a PD controller with acceleration feedforward:

$$\mathbf{a}^n = \ddot{\boldsymbol{\eta}}_d - \mathbf{K}_d \dot{\tilde{\boldsymbol{\eta}}} - \mathbf{K}_p \tilde{\boldsymbol{\eta}} \quad (13.218)$$

where $\mathbf{K}_p > 0$ and $\mathbf{K}_d > 0$. Hence, the error dynamics can be expressed according to

$$\mathbf{M}^*(\boldsymbol{\eta})[\ddot{\tilde{\boldsymbol{\eta}}} + \mathbf{K}_d \dot{\tilde{\boldsymbol{\eta}}} + \mathbf{K}_p \tilde{\boldsymbol{\eta}}] = \mathbf{J}_\Theta^{-T}(\boldsymbol{\eta})\Phi(\mathbf{a}^b, \mathbf{v}, \boldsymbol{\eta})\tilde{\boldsymbol{\theta}} \quad (13.219)$$

Writing this expression in state-space form yields

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{J}_\Theta^{-T}(\boldsymbol{\eta})\Phi(\mathbf{a}^b, \mathbf{v}, \boldsymbol{\eta})\tilde{\boldsymbol{\theta}} \quad (13.220)$$

where $\mathbf{x} = [\tilde{\boldsymbol{\eta}}^\top, \dot{\tilde{\boldsymbol{\eta}}}^\top]^\top$ and

$$\mathbf{A} = \begin{bmatrix} \mathbf{0} & \mathbf{I} \\ -\mathbf{K}_p & -\mathbf{K}_d \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} \mathbf{0} \\ \mathbf{M}^*(\boldsymbol{\eta})^{-1} \end{bmatrix} \quad (13.221)$$

The convergence of $\tilde{\boldsymbol{\eta}}$ to zero can be proven by considering

$$V(\mathbf{x}, \tilde{\boldsymbol{\theta}}, t) = \mathbf{x}^\top \mathbf{P} \mathbf{x} + \tilde{\boldsymbol{\theta}}^\top \boldsymbol{\Gamma}^{-1} \tilde{\boldsymbol{\theta}} \quad (13.222)$$

with a time-varying $\mathbf{P} = \mathbf{P}^\top > 0$ and where $\boldsymbol{\Gamma} = \boldsymbol{\Gamma}^\top > 0$ is a positive definite weighting matrix of appropriate dimension. Differentiating V with respect to time and substituting the error dynamics into the expression for \dot{V} yields

$$\dot{V} = \mathbf{x}^\top (\dot{\mathbf{P}} + \mathbf{P}\mathbf{A} + \mathbf{A}^\top \mathbf{P})\mathbf{x} + 2(\mathbf{x}^\top \mathbf{P}\mathbf{B}\mathbf{J}_\Theta^{-T}\Phi + \tilde{\boldsymbol{\theta}}^\top \boldsymbol{\Gamma}^{-1})\tilde{\boldsymbol{\theta}} \quad (13.223)$$

Assume that the parameters are constant such that $\dot{\tilde{\boldsymbol{\theta}}} = \mathbf{0}$ holds. The parameter update law is chosen as

$$\dot{\tilde{\boldsymbol{\theta}}} = -\boldsymbol{\Gamma}\Phi^\top(\mathbf{a}^b, \mathbf{v}, \boldsymbol{\eta})\mathbf{J}_\Theta^{-1}(\boldsymbol{\eta})\mathbf{y} \quad (13.224)$$

where \mathbf{y} is signal vector given by

$$\mathbf{y} = \mathbf{Cx}, \quad \mathbf{C} = \mathbf{B}^\top \mathbf{P} \quad (13.225)$$

In order to prove that $\dot{V} \leq 0$, let

$$\mathbf{C} = [c_0 \mathbf{I} \ c_1 \mathbf{I}] \quad (13.226)$$

where $c_0 > 0$ and $c_1 > 0$ are two scalars to be interpreted later. Furthermore, let

$$\mathbf{PA} + \mathbf{A}^\top \mathbf{P} = -\mathbf{Q}, \quad \mathbf{Q} = \mathbf{Q}^\top > 0 \quad (13.227)$$

where \mathbf{P} and \mathbf{Q} are defined according to Asare and Wilson (1986):

$$\mathbf{P} := \begin{bmatrix} c_0 \mathbf{M}^* \mathbf{K}_d + c_1 \mathbf{M}^* \mathbf{K}_p & c_0 \mathbf{M}^* \\ c_0 \mathbf{M}^* & c_1 \mathbf{M}^* \end{bmatrix} \quad (13.228)$$

$$\mathbf{Q} := \begin{bmatrix} 2c_0 \mathbf{M}^* \mathbf{K}_p & \mathbf{0} \\ \mathbf{0} & 2(c_1 \mathbf{M}^* \mathbf{K}_d - c_0 \mathbf{M}^*) \end{bmatrix} \quad (13.229)$$

Assume that there exists a constant $\beta > 0$ such that

$$\mathbf{x}^\top \dot{\mathbf{P}} \mathbf{x} = \mathbf{x}^\top \begin{bmatrix} c_0 \dot{\mathbf{M}}^* \mathbf{K}_d + c_1 \dot{\mathbf{M}}^* \mathbf{K}_p & c_0 \dot{\mathbf{M}}^* \\ c_0 \dot{\mathbf{M}}^* & c_1 \dot{\mathbf{M}}^* \end{bmatrix} \mathbf{x} \leq \beta \mathbf{x}^\top \begin{bmatrix} \mathbf{M}^* & \mathbf{0} \\ \mathbf{0} & \mathbf{M}^* \end{bmatrix} \mathbf{x} \quad (13.230)$$

Hence, $\mathbf{P} = \mathbf{P}^\top > 0$, $c_0 > 0$, $c_1 > 0$ and $\mathbf{x}^\top \mathbf{Q} \mathbf{x} > \mathbf{x}^\top \dot{\mathbf{P}} \mathbf{x}$ implies that

$$\dot{V} = \mathbf{x}^\top (\dot{\mathbf{P}} - \mathbf{Q}) \mathbf{x} \leq 0 \quad (13.231)$$

if the following requirements are satisfied:

- (i) $(c_0 \mathbf{K}_d + c_1 \mathbf{K}_p)c_1 > c_0^2 \mathbf{I}$
- (ii) $2c_0 \mathbf{K}_p > \beta \mathbf{I}$
- (iii) $2(c_1 \mathbf{K}_d - c_0 \mathbf{I}) > \beta \mathbf{I}$

Here β is usually taken to be a small positive constant while $\mathbf{K}_p > 0$ and $\mathbf{K}_d > 0$ can be chosen as diagonal matrices. Consequently, convergence of $\tilde{\eta}$ and $\dot{\tilde{\eta}}$ to zero is guaranteed by applying *Barbălat's lemma* (Barbălat, 1959); see Appendix A.2. It is also seen that the parameter vector $\tilde{\theta}$ will be bounded but not necessarily convergent.

Adaptive feedback linearization has been applied to the ship autopilot control problem by Fossen and Paulsen (1992). The assumption that $\mathbf{x}^\top \dot{\mathbf{P}} \mathbf{x}$ is bounded by a positive constant β can be relaxed by using adaptive slide-mode control where the skew-symmetric property $\mathbf{x}^\top [\dot{\mathbf{M}} - 2\mathbf{C}(\mathbf{v})] \mathbf{x} = \mathbf{0}$ is exploited (see Slotine and Benedetto, 1990, Fossen, 1993).

13.3 Integrator Backstepping

Backstepping is a design methodology for construction of a feedback control law through a *recursive* construction of a control Lyapunov function. Nonlinear backstepping designs are strongly related to feedback linearization. However, while feedback linearization methods cancel all nonlinearities in the system it will be shown that when applying the backstepping design methodology more design flexibility is obtained. In particular, the designer is given the possibility to exploit “good” nonlinearities while “bad” nonlinearities can be dominated by adding nonlinear damping, for instance. Hence, additional robustness

is obtained, which is important in industrial control systems since cancellation of all nonlinearities requires precise models that are difficult to obtain in practice.

13.3.1 A Brief History of Backstepping

The idea of integrator backstepping seems to have appeared simultaneously, often implicit, in the works of Koditschek (1987), Sonntag and Sussmann (1988), Tsinias (1989) and Byrnes and Isidori (1989). Stabilization through an integrator (Kokotovic and Sussmann, 1989) can be viewed as a special case of stabilization through an SPR transfer function, which is a frequently used technique in the early adaptive designs (see Parks, 1966, Landau, 1979, Narendra and Annaswamy, 1989). Extensions to nonlinear cascades by using passivity arguments have been done by Ortega (1991) and Byrnes *et al.* (1991). Integrator backstepping appeared as a recursive design technique in Saberi *et al.* (1990) and was further developed by Kanellakopoulos *et al.* (1992). The relationship between backstepping and passivity has been established by Lozano *et al.* (1992). For the interested reader, a tutorial overview of backstepping is given in Kokotovic (1991).

Adaptive and nonlinear backstepping designs are described in detail by Krstic *et al.* (1995). This includes methods for parameter adaptation, tuning functions and modular designs for both full-state feedback and output feedback (observer backstepping). Sepulchre *et al.* (1997) make extensions to forwarding, passivity and cascaded designs. Also discussions on stability margins and optimality are included. The concept of vectorial backstepping was first introduced by Fossen and Berge (1997). Vectorial backstepping exploits the structural properties of nonlinear MIMO systems and this simplifies design and analysis significantly.

Krstic and Deng (1998) present stochastic systems with a focus on stochastic stability and regulation.

The focus of this section is practical designs with implementation considerations for mechanical systems. This is done by exploiting the nonlinear system properties of mechanical systems such as dissipativity (good damping), symmetry of the inertia matrix and the skew-symmetric property of the Coriolis and centripetal matrix. In addition, emphasis is placed on control design with integral action. Two techniques for integral action in nonlinear systems using backstepping designs are discussed (see Loria *et al.*, 1999, Fossen *et al.*, 2001).

13.3.2 The Main Idea of Integrator Backstepping

Integrator backstepping is a *recursive* design technique using *control Lyapunov functions (CLF)*. The CLF concept is a generalization of Lyapunov design results by, for instance, Jacobson (1977) and Jurdjevic and Quinn (1978).

Definition 13.2 (Control Lyapunov Function)

A smooth positive definite and radially unbounded function $V : \mathbb{R}^n \rightarrow \mathbb{R}_+$ is called a *control Lyapunov function* for (see Arstein, 1983, Sontag, 1983)

$$\dot{x} = f(x, u) \quad (13.232)$$

where $x \in \mathbb{R}^n$ and $u \in \mathbb{R}^r$ if

$$\inf_{u \in \mathbb{R}^r} \left\{ \frac{\partial V}{\partial x}(x) f(x, u) \right\} < 0, \quad \forall x \neq 0 \quad (13.233)$$

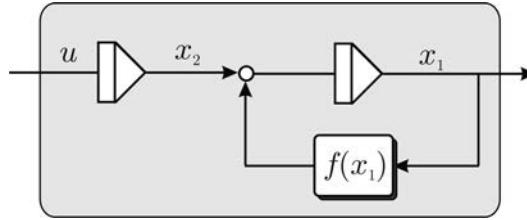


Figure 13.11 Second-order nonlinear system with one single nonlinearity $f(x_1)$ and a pure integrator at the input.

The main idea of integrator backstepping can be demonstrated by considering a simple nonlinear scalar system:

$$\dot{x}_1 = f(x_1) + x_2 \quad (13.234)$$

$$\dot{x}_2 = u \quad (13.235)$$

$$y = x_1 \quad (13.236)$$

where $x_1 \in \mathbb{R}$, $x_2 \in \mathbb{R}$, $y \in \mathbb{R}$ and $u \in \mathbb{R}$. The second equation represents a pure integrator (see Figure 13.11).

Let the design objective be regulation of $y \rightarrow 0$ as $t \rightarrow \infty$. The only equilibrium point with $y = 0$ is $(x_1, x_2) = (0, -f(0))$ corresponding to $\dot{x}_1 = f(0) + x_2 = 0$. The design objective is to render the equilibrium point GAS or GES. Since the nonlinear system (13.234)–(13.235) consists of two states x_1 and x_2 , this will be a recursive design in two steps. Equations (13.234)–(13.235) are therefore treated as two cascaded systems, each with a single input and output. The recursive design starts with the system x_1 and continues with x_2 . A change of coordinates

$$z = \phi(x) \quad (13.237)$$

is introduced during the recursive design process where z is a new state vector and $\phi(x) : \mathbb{R}^n \rightarrow \mathbb{R}^n$ is a transformation to be interpreted later. The backstepping transformation is a *global diffeomorphism*, that is a mapping with smooth functions $\phi(x)$ and $\phi^{-1}(x)$. Hence, the existence of an inverse transformation

$$x = \phi^{-1}(z) \quad (13.238)$$

is guaranteed.

Step 1: For the first system (13.234) the state x_2 is chosen as a *virtual control* input while it is recalled that our design objective is to regulate the output $y = x_1$ to zero. Hence, the first backstepping variable is chosen as

$$z_1 = x_1 \quad (13.239)$$

The virtual control is defined as

$$x_2 := \alpha_1 + z_2 \quad (13.240)$$

where

α_1 = stabilizing function
 z_2 = new state variable

Hence, the z_1 system can be written

$$\dot{z}_1 = f(z_1) + \alpha_1 + z_2 \quad (13.241)$$

The new state variable z_2 will not be used in the first step, but its presence is important since z_2 is needed to couple the z_1 system to the next system, that is the z_2 system to be considered in the next step. Moreover, integrator backstepping implies that the coordinates during the recursive design are changed from $x = [x_1, x_2]^\top$ to $z = [z_1, z_2]^\top$.

A CLF for the z_1 system is

$$V_1 = \frac{1}{2}z_1^2 \quad (13.242)$$

$$\begin{aligned} \dot{V}_1 &= z_1 \dot{z}_1 \\ &= z_1(f(z_1) + \alpha_1) + z_1 z_2 \end{aligned} \quad (13.243)$$

We now turn our attention to the design of the stabilizing function α_1 which will provide the necessary feedback for the z_1 system. For instance, choosing the stabilizing function as a feedback linearizing controller

$$\alpha_1 = -f(z_1) - k_1 z_1 \quad (13.244)$$

where $k_1 > 0$ is the feedback gain, yields

$$\dot{V}_1 = -k_1 z_1^2 + z_1 z_2 \quad (13.245)$$

and

$$\dot{z}_1 = -k_1 z_1 + z_2 \quad (13.246)$$

A block diagram showing the stabilizing function and the new state variable is shown in Figure 13.12. Hence, if $z_2 = 0$ then the z_1 system is stabilized. We now turn our attention to the z_2 system.

Step 2: The z_2 dynamics is computed by time differentiation of (13.240):

$$\begin{aligned} \dot{z}_2 &= \dot{x}_2 - \dot{\alpha}_1 \\ &= u - \dot{\alpha}_1 \end{aligned} \quad (13.247)$$

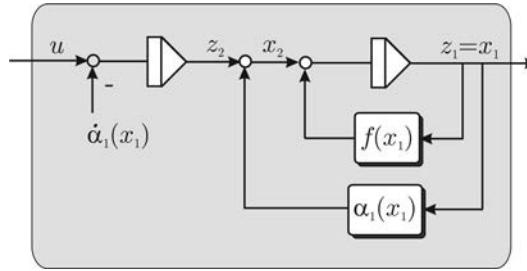


Figure 13.12 Stabilization of the x_1 system by means of the stabilizing function $\alpha_1 = \alpha_1(x_1)$. Note that $-\dot{\alpha}_1(x_1)$ when integrated cancels out the feedback term $\alpha_1(x_1)$.

A CLF for the z_2 system is

$$V_2 = V_1 + \frac{1}{2}z_2^2 \quad (13.248)$$

$$\begin{aligned} \dot{V}_2 &= \dot{V}_1 + \dot{z}_2 z_2 \\ &= (-k_1 z_1^2 + z_1 z_2) + \dot{z}_2 z_2 \\ &= -k_1 z_1^2 + z_2(z_1 + \dot{z}_2) \\ &= -k_1 z_1^2 + z_2(u - \dot{\alpha}_1 + z_1) \end{aligned} \quad (13.249)$$

Since our system has relative degree two, the control input u appears in the second step (see Figure 13.13). Hence, choosing the control law as

$$u = \dot{\alpha}_1 - z_1 - k_2 z_2 \quad (13.250)$$

with $k_2 > 0$ yields

$$\dot{V}_2 = -k_1 z_1^2 - k_2 z_2^2 < 0, \quad \forall z_1 \neq 0, z_2 \neq 0 \quad (13.251)$$

Implementation Aspects

When implementing the control law (13.250) it is important to avoid expressions involving the time derivatives of the states. For this simple system only $\dot{\alpha}_1$ must be evaluated. This can be done by

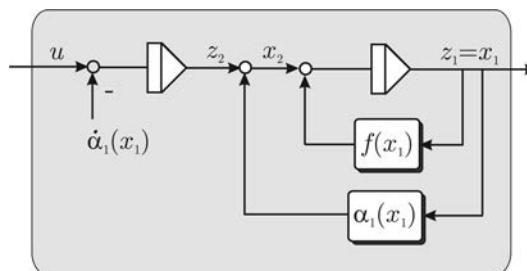


Figure 13.13 Stabilization of the x_2 system by means of the control input $u = u(\dot{\alpha}_1, z_1, z_2)$.

time differentiation of $\alpha_1(x_1)$ along the trajectory of x_1 . Hence, $\dot{\alpha}_1$ can be computed without using the state derivatives:

$$\begin{aligned}\dot{\alpha}_1 &= -\frac{\partial f(x_1)}{\partial x_1} \dot{x}_1 - k_1 \dot{x}_1 \\ &= -\left(\frac{\partial f(x_1)}{\partial x_1} + k_1\right) (f(x_1) + x_2)\end{aligned}\quad (13.252)$$

The final expression for the control law is then

$$u = -\left(\frac{\partial f(x_1)}{\partial x_1} + k_1\right) (f(x_1) + x_2) - x_1 - k_2(x_2 + f(x_1) + k_1 x_1) \quad (13.253)$$

If $f(x_1) = -x_1$ (linear theory), it is seen that

$$\begin{aligned}u &= -(-1 + k_1)(-x_1 + x_2) - x_1 - k_2(x_2 - x_1 + k_1 x_1) \\ &= -\underbrace{(2 + k_1 k_2 - k_1 - k_2)}_{K_p} x_1 - \underbrace{(k_1 + k_2 - 1)}_{K_d} x_2\end{aligned}\quad (13.254)$$

which is a standard PD control law. In general, the expression for u is a nonlinear feedback control law depending on the nonlinear function $f(x_1)$.

Backstepping Coordinate Transformation

The backstepping coordinate transformation $z = \phi(x)$ takes the form

$$\begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 + f(x_1) + k_1 x_1 \end{bmatrix} \quad (13.255)$$

while the inverse transformation $x = \phi^{-1}(z)$ is

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} z_1 \\ z_2 - f(z_1) - k_1 z_1 \end{bmatrix} \quad (13.256)$$

The Final Check

If you have performed the backstepping design procedure correctly the dynamics of the closed-loop system in (z_1, z_2) coordinates can always be written as the sum of a diagonal and skew-symmetric matrix times the state vector. This can be seen by writing the resulting dynamics in the form

$$\begin{bmatrix} \dot{z}_1 \\ \dot{z}_2 \end{bmatrix} = -\underbrace{\begin{bmatrix} k_1 & 0 \\ 0 & k_2 \end{bmatrix}}_{\text{diagonal matrix}} \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} + \underbrace{\begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}}_{\text{skew-symmetrical matrix}} \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} \quad (13.257)$$

or equivalently

$$\dot{z} = -Kz + Sz \quad (13.258)$$

where $\mathbf{z} = [z_1, z_2]^\top$, $\mathbf{K} = \text{diag}\{k_1, k_2\} > 0$ and

$$\mathbf{S} = -\mathbf{S}^\top = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \quad (13.259)$$

where \mathbf{S} satisfies $\mathbf{z}^\top \mathbf{S} \mathbf{z} = 0, \forall \mathbf{z}$. In some cases the diagonal matrix will be a function of the state; that is $\mathbf{K}(\mathbf{z}) > 0$. This is the case when nonlinear damping is added or when some of the nonlinearities not are canceled by the controller.

Investigation of Stability

It is also seen that

$$V_2 = \frac{1}{2} \mathbf{z}^\top \mathbf{z} \quad (13.260)$$

$$\begin{aligned} \dot{V}_2 &= \mathbf{z}^\top (-\mathbf{K} \mathbf{z} + \mathbf{S} \mathbf{z}) \\ &= -\mathbf{z}^\top \mathbf{K} \mathbf{z} \end{aligned} \quad (13.261)$$

Hence, Lyapunov's direct method for autonomous systems ensures that the equilibrium point $(x_1, x_2) = (0, -f(0))$ is GAS. In fact, this system will also be GES since it can be shown that the state vector \mathbf{x} decays exponentially to zero by using Theorem A.3; that is

$$\|\mathbf{z}(t)\|_2 \leq e^{-\beta(t-t_0)} \|\mathbf{z}(t_0)\|_2 \quad (13.262)$$

where $\beta = \lambda_{\min}(\mathbf{K}) > 0$ is the convergence rate.

A generalization to SISO mass–damper–spring systems is done in Section 13.3.3 while extensions to MIMO control are made in Section 13.3.6.

Backstepping versus Feedback Linearization

The backstepping control law of the previous section is in fact equal to a feedback linearizing controller since the nonlinear function $f(x_1)$ is perfectly compensated for by choosing the stabilizing function as

$$\alpha_1 = -f(x_1) - k_1 z_1 \quad (13.263)$$

The disadvantage with this approach is that a perfect model is required. This is impossible in practice. Consequently, an approach of canceling all the nonlinearities may be sensitive for modeling errors.

One of the nice features of backstepping is that the stabilizing functions can be modified to exploit so-called “good” nonlinearities. For instance, assume that

$$f(x_1) = -a_0 x_1 - a_1 x_1^2 - a_2 |x_1| x_1 \quad (13.264)$$

where a_0, a_1 and a_2 are assumed to be *unknown* positive constants. Since both $a_0 x_1$ and $a_2 |x_1| x_1$ tend to damp out the motion these two expressions should be exploited in the control design and therefore not canceled out. On the contrary, the destabilizing term $a_1 x_1^2$ must be perfectly compensated for or dominated by adding a nonlinear damping term proportional to x_1^3 (remember that $z_1 = x_1$).

Nonlinear damping suggests the following candidate for the stabilizing function:

$$\alpha_1 = \underbrace{-k_1 z_1}_{\text{linear damping}} \quad \underbrace{-\kappa_1 z_1^3}_{\text{nonlinear damping}} \quad (13.265)$$

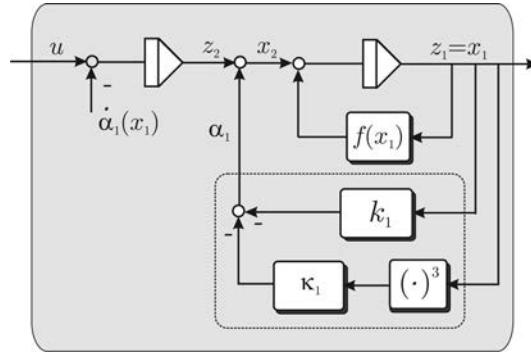


Figure 13.14 Domination of destabilizing terms by adding nonlinear damping.

where $k_1 > 0$ and $\kappa_1 > 0$ (see Figure 13.14). Hence,

$$\begin{aligned}
 \dot{z}_1 &= f(z_1) + (\alpha_1 + z_2) \\
 &= -a_0 z_1 - a_1 z_1^2 - a_2 |z_1| z_1 - (k_1 + \kappa_1 z_1^2) z_1 + z_2 \\
 &= -(\underbrace{a_0 + a_2 |z_1|}_{\text{good damping}} + k_1) z_1 - \underbrace{a_1 z_1^2}_{\text{bad damping}} - \kappa_1 z_1^3 + z_2
 \end{aligned} \tag{13.266}$$

Consider the CLF:

$$V_1 = \frac{1}{2} z_1^2 \tag{13.267}$$

$$\dot{V}_1 = -(a_0 + a_2 |z_1| + k_1) z_1^2 - a_1 z_1^3 - \kappa_1 z_1^4 + z_1 z_2 \tag{13.268}$$

In the next step it is seen that

$$\begin{aligned}
 V_2 &= V_1 + \frac{1}{2} z_2^2 \\
 \dot{V}_2 &= -(\underbrace{a_0 + a_2 |z_1| + k_1}_{\text{energy dissipation}}) z_1^2 - \underbrace{a_1 z_1^3}_{\text{energy dissipation/generation}} - \kappa_1 z_1^4 + z_2(z_1 + u - \dot{\alpha}_1)
 \end{aligned}$$

From this expression it can be concluded that the good damping terms contribute to the energy dissipation. The bad damping term, however, must be dominated by the nonlinear damping term. Choosing

$$u = \dot{\alpha}_1 - k_2 z_2 - z_1 \tag{13.269}$$

finally yields

$$\dot{V}_2 = -(a_0 + a_2 |z_1| + k_1) z_1^2 - a_1 z_1^3 - \kappa_1 z_1^4 - k_2 z_2^2 \tag{13.270}$$

This expression can be rewritten by *completing the squares*. Consider the expression

$$\left(\frac{1}{2\sqrt{\kappa_1}}x + \sqrt{\kappa_1}y \right)^2 = \frac{1}{4\kappa_1}x^2 + xy + \kappa_1y^2 \geq 0 \quad (13.271)$$

⇓

$$-xy - \kappa_1y^2 = - \left(\frac{1}{2\sqrt{\kappa_1}}x + \sqrt{\kappa_1}y \right)^2 + \frac{1}{4\kappa_1}x^2 \quad (13.272)$$

Equation (13.270) with $x = a_1z_1$ and $y = z_1^2$ yields

$$\dot{V}_2 = - \left(\frac{a_1}{2\sqrt{\kappa_1}}z_1 + \sqrt{\kappa_1}z_1^2 \right)^2 + \frac{a_1^2}{4\kappa_1}z_1^2 - (a_0 + a_2|z_1| + k_1)z_1^2 - k_2z_2^2 \quad (13.273)$$

Since

$$\begin{aligned} - \left(\frac{a_1}{2\sqrt{\kappa_1}}z_1 + \sqrt{\kappa_1}z_1^2 \right)^2 &\leq 0 \\ -a_2|z_1| &\leq 0 \end{aligned} \quad (13.274)$$

it then follows that

$$\dot{V}_2 \leq - \left(a_0 + k_1 - \frac{a_1^2}{4\kappa_1} \right) z_1^2 - k_2 z_2^2 \quad (13.275)$$

Hence, by choosing the controller gains according to

$$\kappa_1 > 0 \quad (13.276)$$

$$k_1 > \frac{a_1^2}{4\kappa_1} - a_0 \quad (13.277)$$

$$k_2 > 0 \quad (13.278)$$

our design goal to render $\dot{V}_2 < 0$ is satisfied. Notice that the controller (13.269) with (13.265) is implemented without using the unknown parameters a_0 , a_1 and a_2 . Hence, a robust nonlinear controller is derived by using backstepping. This result differs from feedback linearization, which is based on model cancelation.

13.3.3 Backstepping of SISO Mass–Damper–Spring Systems

The results of Section 13.3.2 can be generalized to the following class of SISO mechanical systems:

$$\dot{x} = v \quad (13.279)$$

$$m\dot{v} + d(v)v + k(x)x = \tau \quad (13.280)$$

$$y = x \quad (13.281)$$

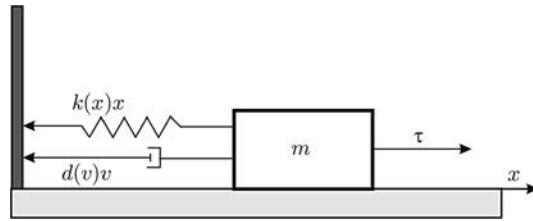


Figure 13.15 Nonlinear mass–damper–spring system.

where \$x\$ is the position, \$v\$ is the velocity and

$$\begin{aligned} m &= \text{mass (positive)} \\ d(v) &= \text{nonlinear damper (non-negative)} \\ k(x) &= \text{nonlinear spring (non-negative)} \end{aligned}$$

The nonlinear mass–damper–spring system is shown in Figure 13.15.

Nonlinear Trajectory-Tracking Control

Backstepping of the mass–damper–spring can be performed by choosing the output

$$e = y - y_d \quad (13.282)$$

where \$e\$ is the tracking error and \$y_d(t) \in C^r\$ is an \$r\$ times differentiable (smooth) and bounded reference trajectory (see Section 10.2.1). Regulation of \$y = x\$ to zero is obtained by choosing \$\dot{y}_d = y_d = 0\$. Time differentiation of \$e\$ yields the following model:

$$\dot{e} = v - \dot{y}_d \quad (13.283)$$

$$m\dot{v} = \tau - d(v)v - k(x)x \quad (13.284)$$

The backstepping control law solving this problem is derived in two recursive steps similar to the integrator backstepping example in Section 13.3.2.

Step 1: Let \$z_1 = e = y - y_d\$, such that

$$\dot{z}_1 = v - \dot{y}_d \quad (13.285)$$

Taking \$v\$ as *virtual control*,

$$v = \alpha_1 + z_2 \quad (13.286)$$

where \$z_2\$ is a new state variable to be interpreted later, yields

$$\dot{z}_1 = \alpha_1 + z_2 - \dot{y}_d \quad (13.287)$$

Next, the stabilizing function \$\alpha_1\$ is chosen as

$$\alpha_1 = \dot{y}_d - [k_1 + n_1(z_1)]z_1 \quad (13.288)$$

where $k_1 > 0$ is a feedback gain and $n_1(z_1) \geq 0$ is a nonlinear damping term, for instance a nonlinear nondecreasing function $n_1(z_1) = \kappa_1 |z_1|^{n_1}$ with $n_1 > 0$ and $\kappa_1 \geq 0$. This yields

$$\dot{z}_1 = -[k_1 + n_1(z_1)]z_1 + z_2 \quad (13.289)$$

A CLF for z_1 is

$$V_1 = \frac{1}{2}z_1^2 \quad (13.290)$$

$$\begin{aligned} \dot{V}_1 &= z_1 \dot{z}_1 \\ &= -[k_1 + n_1(z_1)]z_1^2 + z_1 z_2 \end{aligned} \quad (13.291)$$

Step 2: The second step stabilizes the z_2 dynamics. Moreover, from (13.286) it is seen that

$$\begin{aligned} m\dot{z}_2 &= m\dot{v} - m\dot{\alpha}_1 \\ &= \tau - d(v)v - k(x)x - m\dot{\alpha}_1 \end{aligned} \quad (13.292)$$

Let V_2 be the second CLF, which is chosen to reflect the kinetic energy $\frac{1}{2}mv^2$ of the system. However, it makes sense to replace the velocity v with z_2 in order to solve the trajectory-tracking control problem. This is usually referred to as “pseudo-kinetic energy”. Consider

$$V_2 = V_1 + \frac{1}{2}mz_2^2 \quad (13.293)$$

$$\begin{aligned} \dot{V}_2 &= \dot{V}_1 + m z_2 \dot{z}_2 \\ &= -[k_1 + n_1(z_1)]z_1^2 + z_1 z_2 + z_2[\tau - d(v)v - k(x)x - m\dot{\alpha}_1] \end{aligned} \quad (13.294)$$

Since the input τ appears in \dot{V}_2 , a value for τ can be prescribed such that \dot{V}_2 becomes negative definite. For instance:

$$\tau = m\dot{\alpha}_1 + d(v)v + k(x)x - z_1 - k_2 z_2 - n_2(z_2)z_2 \quad (13.295)$$

where $k_2 > 0$ and $n_2(z_2) = \kappa_2 |z_2|^{n_2} \geq 0$ with $n_2 > 0$ can be specified by the designer. This yields

$$\dot{V}_2 = -[k_1 + n_1(z_1)]z_1^2 - [k_2 + n_2(z_2)]z_2^2 \quad (13.296)$$

When implementing the control law, $\dot{\alpha}_1$ is computed by taking the time derivative of α_1 along the trajectories of y_d and z_1 , see (13.288), to obtain

$$\dot{\alpha}_1 = \frac{\partial \alpha_1}{\partial \ddot{y}_d} \ddot{y}_d - \frac{\partial \alpha_1}{\partial z_1} \dot{z}_1 = \ddot{y}_d - \frac{\partial \alpha_1}{\partial z_1} (v - \dot{y}_d) \quad (13.297)$$

Hence, the state derivatives are avoided in the control law. Notice that the desired state y_d is assumed to be smooth such that \dot{y}_d and \ddot{y}_d exist.

Error Dynamics

The resulting error dynamics is written

$$\begin{aligned} \begin{bmatrix} 1 & 0 \\ 0 & m \end{bmatrix} \begin{bmatrix} \dot{z}_1 \\ \dot{z}_2 \end{bmatrix} &= - \begin{bmatrix} k_1 + n_1(z_1) & 0 \\ 0 & k_2 + n_2(z_2) \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} + \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} \\ &\Downarrow \\ M\dot{z} &= -K(z)z + Sz \end{aligned} \quad (13.298)$$

where $z = [z_1, z_2]^\top$ and

$$M = \text{diag}\{1, m\}$$

$$K(z) = \text{diag}\{k_1 + n_1(z_1), k_2 + n_2(z_2)\}$$

$$S = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$$

Hence, the equilibrium point $(z_1, z_2) = (0, 0)$ is GES. This can be seen from $V_2(z) = \frac{1}{2}z^\top Mz$, which after time differentiation yields $\dot{V}_2(z) = -z^\top Kz$ since $z^\top Sz = 0, \forall z$. Notice that kinetic energy has been applied in the Lyapunov analysis to achieve this.

Setpoint Regulation

Setpoint regulation is obtained by choosing $\dot{y}_d = y_d = 0$. For simplicity let $n_1(z_1) = n_2(z_2) = 0$ such that

$$z_1 = x$$

$$\alpha_1 = -k_1 z_1$$

and

$$\tau = m\dot{\alpha}_1 + d(v)v + k(x)x - z_1 - k_2 z_2 \quad (13.299)$$

Nonlinear PD Control

The backstepping control law (13.299) can also be viewed as a nonlinear PD control law:

$$u = -K_p(x)x - K_d(v)v \quad (13.300)$$

by writing (13.299) as

$$\begin{aligned} u &= [d(v) - mk_1]v + [k(x) - 1]x - k_2(v + k_1x) \\ &= [d(v) - mk_1 - k_2]v + [k(x) - 1 - k_1k_2]x \end{aligned} \quad (13.301)$$

Hence,

$$K_p(x) = k_1 k_2 + 1 - k(x) \quad (13.302)$$

$$K_d(v) = m k_1 + k_2 - d(v) \quad (13.303)$$

Nonlinear PID Control

The nonlinear PD controller (13.300) can be extended to include integral action by using *constant parameter adaptation* or by *augmenting an additional integrator* to the plant. More specifically

1. *Constant parameter adaptation*: An unknown constant (or slowly varying) disturbance is added to the dynamic model. This constant or bias is estimated online by using adaptive control. The resulting system with parameter estimator can be shown to be UGAS for the case of regulation and trajectory-tracking control (Fossen *et al.*, 2001).
2. *Integrator augmentation*: An additional integrator is augmented on the right-hand side of the integrator chain in order to obtain zero steady-state errors. The resulting system is proven to be GES.

The methods are presented in Sections 13.3.4 and 13.3.5.

13.3.4 Integral Action by Constant Parameter Adaptation

The constant parameter adaptation technique is based on Fossen *et al.* (2001). For simplicity a mass–damper–spring system is considered. Hence, adaptive backstepping results in a control law of PID type.

Consider the system:

$$\dot{x} = v \quad (13.304)$$

$$m\dot{v} + d(v)v + k(x)x = \tau + w \quad (13.305)$$

$$\dot{w} = 0 \quad (13.306)$$

The trajectory-tracking control law can be designed by considering the tracking error

$$z_1 = x - x_d \quad (13.307)$$

with

$$\begin{aligned} \dot{z}_1 &= \dot{x} - \dot{x}_d \\ &= v - \dot{x}_d \\ &= (\alpha_1 + z_2) - v_d \end{aligned} \quad (13.308)$$

where z_2 is a new state variable and $v := \alpha_1 + z_2$ is the virtual control for z_1 . Choosing the stabilizing function

$$\alpha_1 = \dot{x}_d - k_1 z_1 \quad (13.309)$$

yields

$$\dot{z}_1 = -k_1 z_1 + z_2 \quad (13.310)$$

The definition $z_2 := v - \alpha_1$ implies that

$$\dot{z}_2 = \dot{v} - \ddot{x}_d + k_1(v - \dot{x}_d) \quad (13.311)$$

$$m\dot{z}_2 = \tau - d(v)v - k(x)x + w - m\ddot{x}_d + mk_1(v - \dot{x}_d) \quad (13.312)$$

Consider the CLF:

$$V_1 = \frac{1}{2}z_1^2 + \frac{1}{2p}\tilde{w}^2, \quad p > 0 \quad (13.313)$$

$$\begin{aligned} \dot{V}_1 &= z_1\dot{z}_1 + \frac{1}{p}\tilde{w}\dot{\tilde{w}} \\ &= z_1z_2 - k_1z_1^2 + \frac{1}{p}\tilde{w}\dot{\tilde{w}} \end{aligned} \quad (13.314)$$

where $\tilde{w} = \hat{w} - w$ is the parameter estimation error. Next, consider the CLF:

$$V_2 = V_1 + \frac{1}{2}mz_2^2 \quad (13.315)$$

$$\begin{aligned} \dot{V}_2 &= \dot{V}_1 + z_2(m\dot{z}_2) \\ &= z_1z_2 - k_1z_1^2 + \frac{1}{p}\tilde{w}\dot{\tilde{w}} \\ &\quad + z_2[\tau - d(v)v - k(x)x + w - m\ddot{x}_d + mk_1(v - \dot{x}_d)] \end{aligned} \quad (13.316)$$

where it is noticed that $\tilde{w} = \dot{\hat{w}}$. Choosing the control law as

$$\tau = d(v)\alpha_1 + k(x)x - \hat{w} + m\ddot{x}_d - mk_1(v - \dot{x}_d) - z_1 - k_2z_2 \quad (13.317)$$

where $\alpha_1 = v - z_2$, yields

$$\dot{V}_2 = -k_1z_1^2 - [k_2 + d(v)]z_2^2 + \tilde{w} \left(\frac{1}{p}\dot{\hat{w}} - z_2 \right) \quad (13.318)$$

Choosing the update law as

$$\dot{w} = pz_2 \quad (13.319)$$

finally yields

$$\dot{V}_2 = -k_1 z_1^2 - [k_2 + d(v)]z_2^2 \quad (13.320)$$

The error dynamics takes the form

$$\begin{bmatrix} \dot{z}_1 \\ \dot{z}_2 \end{bmatrix} = \begin{bmatrix} -k_1 & 1 \\ -1 & -k_2 - d(v) \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} + \begin{bmatrix} 0 \\ -1 \end{bmatrix} \tilde{w} \quad (13.321)$$

$$\dot{\tilde{w}} = -p \begin{bmatrix} 0 & -1 \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} \quad (13.322)$$

\Updownarrow

$$\dot{z} = \mathbf{h}(z, t) + \mathbf{b}\tilde{w} \quad (13.323)$$

$$\dot{\tilde{w}} = -p\mathbf{b}^\top \left(\frac{\partial W(z, t)}{\partial z} \right)^\top \quad (13.324)$$

Notice that the dissipative term $d(v) = d(z_2 + \alpha_1) = d(z_2 - k_1 z_1 + \dot{x}_d(t)) > 0$, $\forall v$ has not been “canceled out” in order to exploit this as good damping in the error dynamics. The price for exploiting the so-called *good nonlinearities* in the design is that the error dynamics becomes *nonautonomous*. Since the feedback gains are assumed to be positive, that is $k_1 > 0$ and $k_2 > 0$, $p > 0$, $\mathbf{b} = [0, -1]^\top$ and $\mathbf{b}^\top \mathbf{b} = 1 > 0$, Theorem A.6 with $W(z) = \frac{1}{2}z^\top z$ guarantees that the nonautonomous systems (13.321)–(13.322) is UGAS.

Notice that if a feedback linearizing controller is applied instead of (13.317), replacing the damping term $d(v)\alpha_1$ with $d(v)v$, the control input becomes

$$\tau = d(v)v + k(x)x - \hat{w} + m\dot{x}_d - mk_1(v - \dot{x}_d) - z_1 - k_2 z_2 \quad (13.325)$$

The error dynamics

$$\begin{bmatrix} \dot{z}_1 \\ \dot{z}_2 \end{bmatrix} = \begin{bmatrix} -k_1 & 1 \\ -1 & -k_2 \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} + \begin{bmatrix} 0 \\ -1 \end{bmatrix} \tilde{w} \quad (13.326)$$

is *autonomous*. In this case, *Krasovskii–LaSalle’s invariant set theorem* (Theorem A.2) can be used to prove GAS.

13.3.5 Integrator Augmentation Technique

Consider the second-order mass–damper–spring system:

$$\dot{x} = v \quad (13.327)$$

$$m\dot{v} + d(v)v + k(x)x = \tau + w \quad (13.328)$$

$$y = x \quad (13.329)$$

where w is a constant *unknown* disturbance. Let e denote the tracking error

$$e = y - y_d \quad (13.330)$$

where y_d is the desired output. Hence,

$$\dot{e} = v - \dot{y}_d \quad (13.331)$$

$$m\dot{v} + d(v)v + k(x)x = \tau + w \quad (13.332)$$

Nonlinear PD Control

If $w = 0$, backstepping results in a nonlinear control law of PD type similar to the result in Section 13.3.3. However, by augmenting the plant with an additional integrator at the right end of the integrator chain, as illustrated in Figure 13.16, nonlinear PID control can be obtained.

Nonlinear PID Control

Augmentation of an additional integrator $\dot{e}_I = e$ to the second-order plant (13.331)–(13.332) yields

$$\dot{e}_I = e \quad (13.333)$$

$$\dot{e} = v - \dot{y}_d \quad (13.334)$$

$$m\dot{v} + d(v)v + k(x)x = \tau + w \quad (13.335)$$

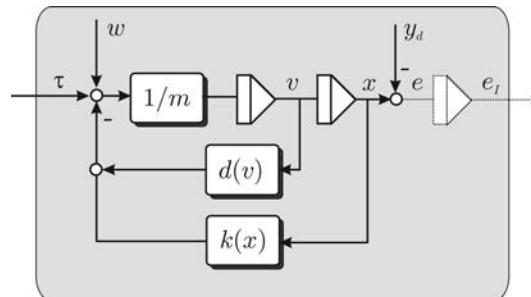


Figure 13.16 Augmentation of an additional integrator.

For simplicity let us first assume that $w = 0$. Hence, backstepping with $z_1 = e_I$ results in three steps:

Step 1:

$$\begin{aligned}\dot{z}_1 &= e \\ &= \alpha_1 + z_2\end{aligned}\tag{13.336}$$

Choosing the stabilizing function $\alpha_1 = -k_1 z_1$ yields

$$\dot{z}_1 = -k_1 z_1 + z_2\tag{13.337}$$

Hence,

$$V_1 = \frac{1}{2} z_1^2\tag{13.338}$$

$$\begin{aligned}\dot{V}_1 &= z_1 \dot{z}_1 \\ &= -k_1 z_1^2 + z_1 z_2\end{aligned}\tag{13.339}$$

Step 2:

$$\begin{aligned}\dot{z}_2 &= \dot{e} - \dot{\alpha}_1 \\ &= v - \dot{y}_d - \dot{\alpha}_1 \\ &= (\alpha_2 + z_3) - \dot{y}_d - \dot{\alpha}_1\end{aligned}\tag{13.340}$$

Hence,

$$V_2 = V_1 + \frac{1}{2} z_2^2\tag{13.341}$$

$$\begin{aligned}\dot{V}_2 &= -k_1 z_1^2 + z_1 z_2 + z_2 \dot{z}_2 \\ &= -k_1 z_1^2 + z_2(z_1 + \alpha_2 + z_3 - \dot{y}_d - \dot{\alpha}_1)\end{aligned}\tag{13.342}$$

Choosing the stabilizing function $\alpha_2 = \dot{\alpha}_1 + \dot{y}_d - k_2 z_2 - z_1$ yields

$$\dot{z}_2 = -z_1 - k_2 z_2 + z_3\tag{13.343}$$

$$\dot{V}_2 = -k_1 z_1^2 - k_2 z_2^2 + z_2 z_3\tag{13.344}$$

Step 3:

$$\begin{aligned}m \dot{z}_3 &= m \dot{v} - m \dot{\alpha}_2 \\ &= \tau + w - d(v)v - k(x)x - m \dot{\alpha}_2 \\ &= \tau - d(v)\alpha_2 - d(v)z_3 - k(x)x - m \dot{\alpha}_2\end{aligned}\tag{13.345}$$

Let

$$V_3 = V_2 + \frac{1}{2}mz_3^2 \quad (13.346)$$

$$\begin{aligned} \dot{V}_3 &= -k_1z_1^2 - k_2z_2^2 + z_3(z_2 + m\dot{z}_3) \\ &= -k_1z_1^2 - k_2z_2^2 + z_3(z_2 + \tau - d(v)\alpha_2 - d(v)z_3 - k(x)x - m\dot{\alpha}_2) \end{aligned} \quad (13.347)$$

Choosing the control law as

$$\tau = m\dot{\alpha}_2 + d(v)\alpha_2 + k(x)x - z_2 - k_3z_3 \quad (13.348)$$

yields

$$\dot{V}_3 = -k_1z_1^2 - k_2z_2^2 - (d(v) + k_3)z_3^2 < 0, \forall z_1 \neq 0, z_2 \neq 0, z_3 \neq 0 \quad (13.349)$$

and

$$m\dot{z}_3 = -[d(v) + k_3]z_3 - z_2 \quad (13.350)$$

Error Dynamics

For the undisturbed case $w = 0$, the error dynamics takes the form

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & m \end{bmatrix} \begin{bmatrix} \dot{z}_1 \\ \dot{z}_2 \\ \dot{z}_3 \end{bmatrix} = - \begin{bmatrix} k_1 & 0 & 0 \\ 0 & k_2 & 0 \\ 0 & 0 & d(v) + k_3 \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix} + \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 1 \\ 0 & -1 & 0 \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix} \quad (13.351)$$

Hence, the equilibrium point $(z_1, z_2, z_3) = (0, 0, 0)$ is GES and therefore the tracking error e converges to zero. If $w = \text{constant}$, the error dynamics takes the form

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & m \end{bmatrix} \begin{bmatrix} \dot{z}_1 \\ \dot{z}_2 \\ \dot{z}_3 \end{bmatrix} = - \begin{bmatrix} k_1 & 0 & 0 \\ 0 & k_2 & 0 \\ 0 & 0 & d(v) + k_3 \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix} + \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 1 \\ 0 & -1 & 0 \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} w$$

Hence, in the steady state ($\dot{z} = \mathbf{0}$ and $d(v) = 0$)

$$z_2 = k_1z_1 = e - \alpha_1 = e + k_1z_1 \Rightarrow e = 0 \quad (13.352)$$

The equilibrium point for $w = \text{constant}$ is

$$\begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix} = \begin{bmatrix} k_1 & -1 & 0 \\ 1 & k_2 & -1 \\ 0 & 1 & k_3 \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} w = \frac{1}{k_1k_2k_3 + k_1 + k_3} \begin{bmatrix} 1 \\ k_1 \\ 1 + k_1k_2 \end{bmatrix} w \quad (13.353)$$

Therefore it can be concluded that for the case $w = \text{constant}$ the equilibrium point (z_1, z_2, z_3) is GES but (z_1, z_2, z_3) will converge to the constant nonzero values given by (13.353), even though $e = 0$. This shows that augmentation of an additional integrator when performing backstepping leads to zero steady-state errors in the case of regulation under the assumption of a constant disturbance w .

Implementation Considerations

The integrator augmentation technique is particularly interesting for implementation in mechanical systems since the integral term is computed by integrating $z_1 = y - y_d$ which for a mechanical system is the position tracking error. This corresponds to applying a PID controller on a second-order system. On the contrary, when using *constant parameter adaptation* the integral term will be the integral of a linear combination of the state tracking errors; see (13.322). For a mechanical system this implies that both the position and velocity tracking errors are used to provide integral action. In many cases it is difficult to measure the velocity with the same accuracy as the position. This implies that the adaptive method will be more sensitive to measurement noise than the *integrator augmentation technique*. A comparative study of the different backstepping integral techniques is found in Skjetne and Fossen (2004).

13.3.6 Case Study: Backstepping of MIMO Mass–Damper–Spring Systems

The concept of vectorial backstepping was first introduced by Fossen and Berge (1997) and Fossen and Grøvlen (1998). Consider a MIMO nonlinear mass–damper–spring system in the form

$$\dot{\mathbf{x}} = \mathbf{v} \quad (13.354)$$

$$\mathbf{M}\dot{\mathbf{v}} + \mathbf{D}(\mathbf{v})\mathbf{v} + \mathbf{K}(\mathbf{x})\mathbf{x} = \mathbf{B}\mathbf{u} \quad (13.355)$$

where $\mathbf{x} \in \mathbb{R}^n$ is the position vector, $\mathbf{v} \in \mathbb{R}^n$ is the velocity vector, $\mathbf{u} \in \mathbb{R}^r$ ($r \geq n$) is the control input vector, $\mathbf{D}(\mathbf{v}) \in \mathbb{R}^{n \times n}$ represents a matrix of damping coefficients, $\mathbf{K}(\mathbf{x}) \in \mathbb{R}^{n \times n}$ is a matrix of spring coefficients, $\mathbf{M} \in \mathbb{R}^{n \times n}$ is the inertia matrix and $\mathbf{B} \in \mathbb{R}^{n \times r}$ is the input matrix. Hence, backstepping can be performed in *two vectorial steps*.

Step 1: For the first system (13.354) consider \mathbf{v} as the control and let

$$\mathbf{v} = \mathbf{s} + \boldsymbol{\alpha}_1 \quad (13.356)$$

where

$\mathbf{s} = \tilde{\mathbf{v}} + \boldsymbol{\Lambda} \tilde{\mathbf{x}}$ New state vector used for tracking control

$\boldsymbol{\alpha}_1$ Stabilizing vector field to be defined later

Here $\tilde{\mathbf{v}} = \mathbf{v} - \mathbf{v}_d$ and $\tilde{\mathbf{x}} = \mathbf{x} - \mathbf{x}_d$ are the velocity and position tracking errors, respectively, and $\boldsymbol{\Lambda} > 0$ is a diagonal matrix of positive elements. The definition of the \mathbf{s} vector is motivated by Slotine and Li (1987), who introduced \mathbf{s} as a measure of tracking when designing their adaptive robot controller. It turns out that this transformation has the nice property of transforming the nonlinear state-space model (13.354)–(13.355) to the form

$$\begin{aligned} \mathbf{M}\dot{\mathbf{s}} + \mathbf{D}(\mathbf{v})\mathbf{s} &= \mathbf{M}\dot{\mathbf{v}} + \mathbf{D}(\mathbf{v})\mathbf{v} - \mathbf{M}\dot{\mathbf{v}}_r - \mathbf{D}(\mathbf{v})\mathbf{v}_r \\ &= \mathbf{B}\mathbf{u} - \mathbf{M}\dot{\mathbf{v}}_r - \mathbf{D}(\mathbf{v})\mathbf{v}_r - \mathbf{K}(\mathbf{x})\mathbf{x} \end{aligned} \quad (13.357)$$

where \mathbf{v}_r can be interpreted as a “virtual” reference trajectory:

$$\begin{aligned}\mathbf{v}_r &= \mathbf{v} - \mathbf{s} \\ &= \mathbf{v}_d - \mathbf{\Lambda} \tilde{\mathbf{x}}\end{aligned}\quad (13.358)$$

The position error dynamics of Step 1 can therefore be written

$$\begin{aligned}\dot{\tilde{\mathbf{x}}} &= \mathbf{v} - \mathbf{v}_d \\ &= s + \boldsymbol{\alpha}_1 - \mathbf{v}_d \quad (\boldsymbol{\alpha}_1 = \mathbf{v}_r = \mathbf{v} - \mathbf{s}) \\ &= -\mathbf{\Lambda} \tilde{\mathbf{x}} + s\end{aligned}\quad (13.359)$$

Hence,

$$V_1 = \frac{1}{2} \tilde{\mathbf{x}}^\top \mathbf{K}_p \tilde{\mathbf{x}}, \quad \mathbf{K}_p = \mathbf{K}_p^\top > 0 \quad (13.360)$$

and

$$\begin{aligned}\dot{V}_1 &= \tilde{\mathbf{x}}^\top \mathbf{K}_p \dot{\tilde{\mathbf{x}}} \\ &= \tilde{\mathbf{x}}^\top \mathbf{K}_p (-\mathbf{\Lambda} \tilde{\mathbf{x}} + s) \\ &= -\tilde{\mathbf{x}}^\top \mathbf{K}_p \mathbf{\Lambda} \tilde{\mathbf{x}} + s^\top \mathbf{K}_p \tilde{\mathbf{x}}\end{aligned}\quad (13.361)$$

Step 2: In the second step, a CLF motivated by *pseudo-kinetic energy* is chosen according to

$$V_2 = \frac{1}{2} \mathbf{s}^\top \mathbf{M} \mathbf{s} + V_1, \quad \mathbf{M} = \mathbf{M}^\top > 0 \quad (13.362)$$

$$\begin{aligned}\dot{V}_2 &= \mathbf{s}^\top \mathbf{M} \dot{\mathbf{s}} + \dot{V}_1 \\ &= \mathbf{s}^\top (\mathbf{B} \mathbf{u} - \mathbf{M} \dot{\mathbf{v}}_r - \mathbf{D}(\mathbf{v}) \mathbf{v}_r - \mathbf{K}(\mathbf{x}) \mathbf{x} - \mathbf{D}(\mathbf{v}) \mathbf{s}) - \tilde{\mathbf{x}}^\top \mathbf{K}_p \mathbf{\Lambda} \tilde{\mathbf{x}} + s^\top \mathbf{K}_p \tilde{\mathbf{x}} \\ &= \mathbf{s}^\top (\mathbf{B} \mathbf{u} - \mathbf{M} \dot{\mathbf{v}}_r - \mathbf{D}(\mathbf{v}) \mathbf{v}_r - \mathbf{K}(\mathbf{x}) \mathbf{x} - \mathbf{D}(\mathbf{v}) \mathbf{s} + \mathbf{K}_p \tilde{\mathbf{x}}) - \tilde{\mathbf{x}}^\top \mathbf{K}_p \mathbf{\Lambda} \tilde{\mathbf{x}}\end{aligned}\quad (13.363)$$

This suggests that the control law is chosen as

$$\mathbf{B} \mathbf{u} = \mathbf{M} \dot{\mathbf{v}}_r + \mathbf{D}(\mathbf{v}) \mathbf{v}_r + \mathbf{K}(\mathbf{x}) \mathbf{x} - \mathbf{K}_p \tilde{\mathbf{x}} - \mathbf{K}_d \mathbf{s}, \quad \mathbf{K}_d > 0 \quad (13.364)$$

which results in

$$\dot{V}_2 = -\mathbf{s}^\top (\mathbf{D}(\mathbf{v}) + \mathbf{K}_d) \mathbf{s} - \tilde{\mathbf{x}}^\top \mathbf{K}_p \mathbf{\Lambda} \tilde{\mathbf{x}}$$

Since V_2 is positive definite and \dot{V}_2 is negative definite it follows from Theorem A.3 that the equilibrium point $(\tilde{\mathbf{x}}, \mathbf{s}) = (\mathbf{0}, \mathbf{0})$ is GES. Moreover, convergence of $\mathbf{s} \rightarrow \mathbf{0}$ and $\tilde{\mathbf{x}} \rightarrow \mathbf{0}$ implies that $\tilde{\mathbf{v}} \rightarrow \mathbf{0}$. When implementing the control law (13.364) it is assumed that \mathbf{B} has an inverse:

$$\mathbf{B}^\dagger = \mathbf{B}^\top (\mathbf{B} \mathbf{B}^\top)^{-1} \quad (13.365)$$

or simply \mathbf{B}^{-1} for the square case $r = n$.

Nonlinear Mass–Damper–Spring System with Actuator Dynamics

Consider the mass–damper–spring system of the previous section with actuator dynamics:

$$\dot{\mathbf{x}} = \mathbf{v} \quad (13.366)$$

$$\mathbf{M}\dot{\mathbf{v}} + \mathbf{D}(\mathbf{v})\mathbf{v} + \mathbf{K}(\mathbf{x})\mathbf{x} = \mathbf{B}\mathbf{u} \quad (13.367)$$

$$\mathbf{T}\dot{\mathbf{u}} + \mathbf{u} = \mathbf{u}_c \quad (13.368)$$

where $\mathbf{T} \in \mathbb{R}^{r \times r}$ is a diagonal matrix of actuator time constants and $\mathbf{u}_c \in \mathbb{R}^r$ is a vector of actuator commands. Instead of choosing the controller \mathbf{u} in Step 2, \mathbf{u}_c is treated as the control input to be specified in Step 3. Recall that

$$\dot{V}_2 = \mathbf{s}^\top (\mathbf{B}\mathbf{u} - \mathbf{M}\dot{\mathbf{v}}_r - \mathbf{D}(\mathbf{v})\mathbf{v}_r - \mathbf{K}(\mathbf{x})\mathbf{x} - \mathbf{D}(\mathbf{v})\mathbf{s} + \mathbf{K}_p\tilde{\mathbf{x}}) - \tilde{\mathbf{x}}^\top \mathbf{K}_p \mathbf{\Lambda} \tilde{\mathbf{x}} \quad (13.369)$$

Step 3: Let \mathbf{Bu} be the virtual control vector of Step 3. Hence,

$$\mathbf{Bu} = \mathbf{z} + \boldsymbol{\alpha}_2 \quad (13.370)$$

$$\boldsymbol{\alpha}_2 = \mathbf{M}\dot{\mathbf{v}}_r + \mathbf{D}(\mathbf{v})\mathbf{v}_r + \mathbf{K}(\mathbf{x})\mathbf{x} - \mathbf{K}_p\tilde{\mathbf{x}} - \mathbf{K}_d\mathbf{s} \quad (13.371)$$

where \mathbf{z} is a new state variable. This results in

$$\dot{V}_2 = \mathbf{s}^\top \mathbf{z} - \mathbf{s}^\top (\mathbf{D}(\mathbf{v}) + \mathbf{K}_d)\mathbf{s} - \tilde{\mathbf{x}}^\top \mathbf{K}_p \mathbf{\Lambda} \tilde{\mathbf{x}} \quad (13.372)$$

Choose

$$V_3 = \frac{1}{2}\mathbf{z}^\top \mathbf{z} + V_2 \quad (13.373)$$

$$\begin{aligned} \dot{V}_3 &= \mathbf{z}^\top \mathbf{K}\dot{\mathbf{z}} + \dot{V}_2 \\ &= \mathbf{z}^\top (\mathbf{B}\dot{\mathbf{u}} - \dot{\boldsymbol{\alpha}}_2) + \mathbf{s}^\top \mathbf{z} - \mathbf{s}^\top (\mathbf{D}(\mathbf{v}) + \mathbf{K}_d)\mathbf{s} - \tilde{\mathbf{x}}^\top \mathbf{K}_p \mathbf{\Lambda} \tilde{\mathbf{x}} \\ &= \mathbf{z}^\top (\mathbf{B}\mathbf{T}^{-1}(\mathbf{u}_c - \mathbf{u}) - \dot{\boldsymbol{\alpha}}_2 + \mathbf{s}) - \mathbf{s}^\top (\mathbf{D}(\mathbf{v}) + \mathbf{K}_d)\mathbf{s} - \tilde{\mathbf{x}}^\top \mathbf{K}_p \mathbf{\Lambda} \tilde{\mathbf{x}} \end{aligned} \quad (13.374)$$

The control law

$$\mathbf{u}_c = \mathbf{u} + \mathbf{T}\mathbf{B}^\dagger(\dot{\boldsymbol{\alpha}}_2 - \mathbf{s} - \mathbf{K}_z\mathbf{z}) \quad (13.375)$$

yields

$$\dot{V}_3 = -\mathbf{z}^\top \mathbf{K}_z \mathbf{z} - \mathbf{s}^\top (\mathbf{D}(\mathbf{v}) + \mathbf{K}_d)\mathbf{s} - \tilde{\mathbf{x}}^\top \mathbf{K}_p \mathbf{\Lambda} \tilde{\mathbf{x}} \quad (13.376)$$

Again GES is guaranteed. The main drawback of including the actuator dynamic is that $\dot{\alpha}_2$ must be computed. The expression for $\dot{\alpha}_2$ will not depend of the state derivatives since

$$\dot{\alpha}_2 = \sum_{i=1}^n \frac{\partial \alpha_2}{\partial (\text{state})_i} (\text{state})_i$$

$(\text{state})_i$ = system equation depending on the states only

Example 13.5 (MIMO Backstepping of Robots)

This example is based on the results of Fossen and Berge (1997). Consider the nonlinear robot model (Sciavicco and Siciliano, 1996):

$$\dot{q} = v \quad (13.377)$$

$$M(q)\dot{v} + C(q, v)v + g(q) = \tau \quad (13.378)$$

where $M(q) = M^\top(q) > 0$ is the inertia matrix, $C(q, v)$ is a matrix of Coriolis and centripetal forces defined in terms of the Christoffel symbols and $g(q)$ is a vector of gravitational forces and moments. $q \in \mathbb{R}^n$ is a vector of joint angles, $v \in \mathbb{R}^n$ is a vector of joint angular rates and $\tau \in \mathbb{R}^n$ is a vector of control torques. Vectorial backstepping of a robot manipulator (see Figure 13.17) can be done in two steps:

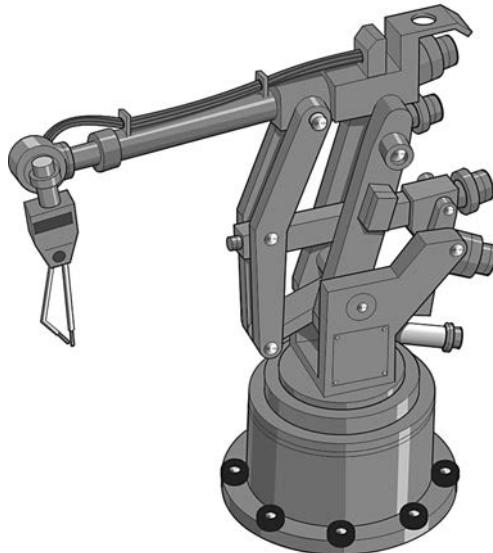


Figure 13.17 Robot manipulator.

Step 1: Define the virtual control vector

$$\dot{\mathbf{q}} = \mathbf{v} := \mathbf{s} + \boldsymbol{\alpha}_1 \quad (13.379)$$

where \mathbf{s} is a new state variable and $\boldsymbol{\alpha}_1$ is stabilizing vector field, which can be chosen as

$$\boldsymbol{\alpha}_1 = \mathbf{v}_r, \quad \mathbf{v}_r = \mathbf{v}_d - \mathbf{\Lambda} \tilde{\mathbf{q}} \quad (13.380)$$

where $\mathbf{\Lambda} > 0$ is a diagonal design matrix and $\tilde{\mathbf{q}} = \mathbf{q} - \mathbf{q}_d$ is the tracking error. Combining (13.379) and (13.380) yields

$$\tilde{\mathbf{v}} = -\mathbf{\Lambda} \tilde{\mathbf{q}} + \mathbf{s} \quad (13.381)$$

where $\dot{\tilde{\mathbf{q}}} = \tilde{\mathbf{v}}$.

Step 2: Consider the CLF:

$$V = \frac{1}{2} (\mathbf{s}^\top \mathbf{M}(\mathbf{q}) \mathbf{s} + \tilde{\mathbf{q}}^\top \mathbf{K}_q \tilde{\mathbf{q}}) > 0, \quad \forall \mathbf{s} \neq \mathbf{0}, \tilde{\mathbf{q}} \neq \mathbf{0} \quad (13.382)$$

$$\begin{aligned} \dot{V} &= \mathbf{s}^\top \mathbf{M}(\mathbf{q}) \dot{\mathbf{s}} + \frac{1}{2} \mathbf{s}^\top \dot{\mathbf{M}}(\mathbf{q}) \mathbf{s} + \tilde{\mathbf{q}}^\top \mathbf{K}_q \tilde{\mathbf{v}} \\ &= \mathbf{s}^\top \mathbf{M}(\mathbf{q}) \dot{\mathbf{s}} + \frac{1}{2} \mathbf{s}^\top \dot{\mathbf{M}}(\mathbf{q}) \mathbf{s} - \tilde{\mathbf{q}}^\top \mathbf{K}_q \mathbf{\Lambda} \tilde{\mathbf{q}} + \tilde{\mathbf{q}}^\top \mathbf{K}_q \mathbf{s} \end{aligned} \quad (13.383)$$

Equations (13.379) and (13.380) can be combined to give

$$\begin{aligned} \mathbf{M}(\mathbf{q}) \dot{\mathbf{s}} &= \mathbf{M}(\mathbf{q}) \dot{\mathbf{v}} - \mathbf{M}(\mathbf{q}) \dot{\boldsymbol{\alpha}} \\ &= \boldsymbol{\tau} - \mathbf{M}(\mathbf{q}) \dot{\mathbf{v}}_r - \mathbf{C}(\mathbf{q}, \mathbf{v}) \mathbf{v}_r - \mathbf{g}(\mathbf{q}) - \mathbf{C}(\mathbf{q}, \mathbf{v}) \mathbf{s} \end{aligned} \quad (13.384)$$

Substituting (13.384) into (13.383) yields

$$\begin{aligned} \dot{V} &= \mathbf{s}^\top (\boldsymbol{\tau} - \mathbf{M}(\mathbf{q}) \dot{\mathbf{v}}_r - \mathbf{C}(\mathbf{q}, \mathbf{v}) \mathbf{v}_r - \mathbf{g}(\mathbf{q}) + \mathbf{K}_q \tilde{\mathbf{q}}) \\ &\quad + \mathbf{s}^\top \left(\frac{1}{2} \dot{\mathbf{M}}(\mathbf{q}) - \mathbf{C}(\mathbf{q}, \mathbf{v}) \right) \mathbf{s} - \tilde{\mathbf{q}}^\top \mathbf{K}_q \mathbf{\Lambda} \tilde{\mathbf{q}} \\ &= \mathbf{s}^\top (\boldsymbol{\tau} - \mathbf{M}(\mathbf{q}) \dot{\mathbf{v}}_r - \mathbf{C}(\mathbf{q}, \mathbf{v}) \mathbf{v}_r - \mathbf{g}(\mathbf{q}) + \mathbf{K}_q \tilde{\mathbf{q}}) - \tilde{\mathbf{q}}^\top \mathbf{K}_q \mathbf{\Lambda} \tilde{\mathbf{q}} \end{aligned} \quad (13.385)$$

Here the skew-symmetric property $\mathbf{s}^\top (\frac{1}{2} \dot{\mathbf{M}}(\mathbf{q}) - \mathbf{C}(\mathbf{q}, \mathbf{v})) \mathbf{s} = \mathbf{0}$, $\forall \mathbf{s}$ has been applied. The backstepping control law is chosen as

$$\boldsymbol{\tau} = \mathbf{M}(\mathbf{q}) \dot{\mathbf{v}}_r + \mathbf{C}(\mathbf{q}, \mathbf{v}) \mathbf{v}_r + \mathbf{g}(\mathbf{q}) - \mathbf{K}_d \mathbf{s} - \mathbf{K}_q \tilde{\mathbf{q}} \quad (13.386)$$

where $\mathbf{K}_d = \mathbf{K}_d^\top > 0$ and $\mathbf{K}_q = \mathbf{K}_q^\top > 0$ are design matrices. This finally yields

$$\dot{V} = -\mathbf{s}^\top \mathbf{K}_d \mathbf{s} - \tilde{\mathbf{q}}^\top \mathbf{K}_q \mathbf{\Lambda} \tilde{\mathbf{q}} < 0, \quad \forall \mathbf{s} \neq \mathbf{0}, \tilde{\mathbf{q}} \neq \mathbf{0} \quad (13.387)$$

and GES follows. The control law (13.386) is equivalent to the control law of Slotine and Li (1987) with perfectly known parameters (nonadaptive case) except for the additional feedback term $\mathbf{K}_q \tilde{\mathbf{q}}$ which is necessary to obtain GES.

13.3.7 Case Study: MIMO Backstepping for Fully Actuated Ships

Conventional ship control systems are designed under the assumption that the kinematic and kinetic equations can be linearized such that gain-scheduling techniques and optimal control theory can be applied (see Fossen, 1994). This is not a good assumption for tracking applications where the surge and sway positions (N, E) and yaw angle ψ must be controlled simultaneously. The main reason for this is that the rotation matrix in yaw must be linearized. In addition to this, assumptions such as linear damping and negligible Coriolis and centripetal forces are only good for low-speed maneuvering and stationkeeping. These limitations clearly motivate a nonlinear design. MIMO nonlinear backstepping designs can be used for this purpose by exploiting nonlinear system properties such as symmetry of the inertia matrix, dissipative damping and skew-symmetry of the Coriolis and centripetal matrix (see Fossen and Fjellstad, 1995).

A MIMO nonlinear backstepping technique for marine craft where the nonlinear system properties are exploited is presented below (Fossen and Strand, 1998). An alternative reference is Fossen and Strand (1999a).

Vectorial Backstepping of Marine Craft in 6 DOF

Consider a marine craft described by the following equations of motion:

$$\dot{\eta} = J_{\Theta}(\eta)v \quad (13.388)$$

$$M\dot{v} + C(v)v + D(v)v + g(\eta) = \tau \quad (13.389)$$

$$\tau = Bu \quad (13.390)$$

This model describes the motion of a craft in 6 DOF. It is assumed that the craft is fully actuated such that BB^T is invertible. The system (13.388)–(13.390) satisfies the following properties:

- (i) $M = M^T > 0$, $\dot{M} = \mathbf{0}$
- (ii) $C(v) = -C^T(v)$
- (iii) $D(v) > 0$
- (iv) BB^T is nonsingular
- (v) $J_{\Theta}(\eta)$ = Euler angle transformation matrix (not defined for $\theta = \pm 90^\circ$)

New State Variables

Assume that the reference trajectories given by $\eta_d^{(3)}, \ddot{\eta}_d, \dot{\eta}_d$ and η_d are smooth and bounded. The virtual reference trajectories in BODY and NED coordinates are defined as

$$\dot{\eta}_r := \dot{\eta}_d - \Lambda \tilde{\eta} \quad (13.391)$$

$$v_r := J_{\Theta}^{-1}(\eta)\dot{\eta}_r, \quad \theta \neq \pm 90^\circ \quad (13.392)$$

where $\tilde{\eta} = \eta - \eta_d$ is the tracking error and $\Lambda > 0$ is a diagonal design matrix. Furthermore, let

$$s = \dot{\eta} - \dot{\eta}_r = \dot{\tilde{\eta}} + \Lambda \tilde{\eta} \quad (13.393)$$

The marine craft dynamics (13.388)–(13.389) can be written (Fossen, 1993)

$$M^*(\eta)\ddot{\eta} + C^*(v, \eta)\dot{\eta} + D^*(v, \eta)\dot{\eta} + g^*(\eta) = J_{\Theta}^{-T}(\eta)\tau \quad (13.394)$$

where

$$\begin{aligned}\mathbf{M}^*(\eta) &= \mathbf{J}_\Theta^{-\top}(\eta) \mathbf{M} \mathbf{J}_\Theta^{-1}(\eta) \\ \mathbf{C}^*(\mathbf{v}, \eta) &= \mathbf{J}_\Theta^{-\top}(\eta) [\mathbf{C}(\mathbf{v}) - \mathbf{M} \mathbf{J}_\Theta^{-1}(\eta) \dot{\mathbf{J}}_\Theta(\eta)] \mathbf{J}_\Theta^{-1}(\eta) \\ \mathbf{D}^*(\mathbf{v}, \eta) &= \mathbf{J}_\Theta^{-\top}(\eta) \mathbf{D}(\mathbf{v}) \mathbf{J}_\Theta^{-1}(\eta) \\ \mathbf{g}^*(\eta) &= \mathbf{J}_\Theta^{-\top}(\eta) \mathbf{g}(\eta)\end{aligned}$$

Hence,

$$\begin{aligned}\mathbf{M}^*(\eta) \dot{s} &= -\mathbf{C}^*(\mathbf{v}, \eta) s - \mathbf{D}^*(\mathbf{v}, \eta) s + \mathbf{J}_\Theta^{-\top}(\eta) \mathbf{B} u \\ &\quad - \mathbf{M}^*(\eta) \ddot{\eta}_r - \mathbf{C}^*(\mathbf{v}, \eta) \dot{\eta}_r - \mathbf{D}^*(\mathbf{v}, \eta) \dot{\eta}_r - \mathbf{g}^*(\eta)\end{aligned}\quad (13.395)$$

or equivalently

$$\begin{aligned}\mathbf{M}^*(\eta) \dot{s} &= -\mathbf{C}^*(\mathbf{v}, \eta) s - \mathbf{D}^*(\mathbf{v}, \eta) s \\ &\quad + \mathbf{J}_\Theta^{-\top}(\eta) [\mathbf{B} u - \mathbf{M} \dot{\mathbf{v}}_r - \mathbf{C}(\mathbf{v}) \mathbf{v}_r - \mathbf{D}(\mathbf{v}) \mathbf{v}_r - \mathbf{g}(\eta)]\end{aligned}\quad (13.396)$$

Step 1: Consider the error dynamics

$$\dot{\eta} - \dot{\eta}_d = \mathbf{J}_\Theta(\eta)(\mathbf{v} - \mathbf{v}_d) \quad (13.397)$$

Let \mathbf{v} be the virtual control vector

$$\mathbf{J}_\Theta(\eta) \mathbf{v} := s + \boldsymbol{\alpha}_1 \quad (13.398)$$

The position error dynamics can therefore be written

$$\begin{aligned}\dot{\tilde{\eta}} &= \mathbf{J}_\Theta(\eta)(\mathbf{v} - \mathbf{v}_d) \\ &= s + \boldsymbol{\alpha}_1 - \mathbf{J}_\Theta(\eta) \mathbf{v}_d \quad \{\boldsymbol{\alpha}_1 = \dot{\eta}_r = \dot{\eta}_d - \boldsymbol{\Lambda} \tilde{\eta}, \quad \dot{\eta}_d = \mathbf{J}_\Theta(\eta) \mathbf{v}_d\} \\ &= -\boldsymbol{\Lambda} \tilde{\eta} + s\end{aligned}\quad (13.399)$$

Hence, a CLF is

$$V_1 = \frac{1}{2} \tilde{\eta}^\top \mathbf{K}_p \tilde{\eta}, \quad \mathbf{K}_p = \mathbf{K}_p^\top > 0 \quad (13.400)$$

resulting in

$$\begin{aligned}\dot{V}_1 &= \tilde{\eta}^\top \mathbf{K}_p \dot{\tilde{\eta}} \\ &= \tilde{\eta}^\top \mathbf{K}_p (-\boldsymbol{\Lambda} \tilde{\eta} + s) \\ &= -\tilde{\eta}^\top \mathbf{K}_p \boldsymbol{\Lambda} \tilde{\eta} + s^\top \mathbf{K}_p \tilde{\eta}\end{aligned}\quad (13.401)$$

Step 2: In the second step a CLF motivated by the *pseudo-kinetic energy* is chosen:

$$V_2 = \frac{1}{2} s^\top \mathbf{M}^*(\eta) s + V_1, \quad \mathbf{M}^* = (\mathbf{M}^*)^\top > 0 \quad (13.402)$$

$$\begin{aligned}
\dot{V}_2 &= s^\top \mathbf{M}^*(\eta) \dot{s} + \frac{1}{2} s^\top \dot{\mathbf{M}}^*(\eta) s + \dot{V}_1 \\
&= -s^\top [C^*(v, \eta) + D^*(v, \eta)] s \\
&\quad + s^\top J_{\Theta}^{-\top}(\eta) [\mathbf{B}u - \mathbf{M}\dot{v}_r - \mathbf{C}(v)v_r - \mathbf{D}(v)v_r - g(\eta)] \\
&\quad + \frac{1}{2} s^\top \dot{\mathbf{M}}^*(\eta) s - \tilde{\eta}^\top \mathbf{K}_p \Lambda \tilde{\eta} + s^\top \mathbf{K}_p \tilde{\eta}
\end{aligned} \tag{13.403}$$

Using the skew-symmetric property

$$s^\top (\dot{\mathbf{M}}^*(\eta) - 2C^*(v, \eta)) s = 0, \quad \forall v, \eta, s \tag{13.404}$$

yields

$$\begin{aligned}
\dot{V}_2 &= s^\top J_{\Theta}^{-\top}(\eta) [\mathbf{B}u - \mathbf{M}\dot{v}_r - \mathbf{C}(v)v_r - \mathbf{D}(v)v_r - g(\eta) + J_{\Theta}^\top(\eta) \mathbf{K}_p \tilde{\eta}] \\
&\quad - s^\top D^*(v, \eta) s - \tilde{\eta}^\top \mathbf{K}_p \Lambda \tilde{\eta}
\end{aligned} \tag{13.405}$$

Hence, the control law can be chosen as (see Figure 13.18)

$$\tau = \mathbf{M}\dot{v}_r + \mathbf{C}(v)v_r + \mathbf{D}(v)v_r + g(\eta) - J_{\Theta}^\top(\eta) \mathbf{K}_p \tilde{\eta} - J_{\Theta}^\top(\eta) \mathbf{K}_d s \tag{13.406}$$

$$u = B^\dagger \tau \tag{13.407}$$

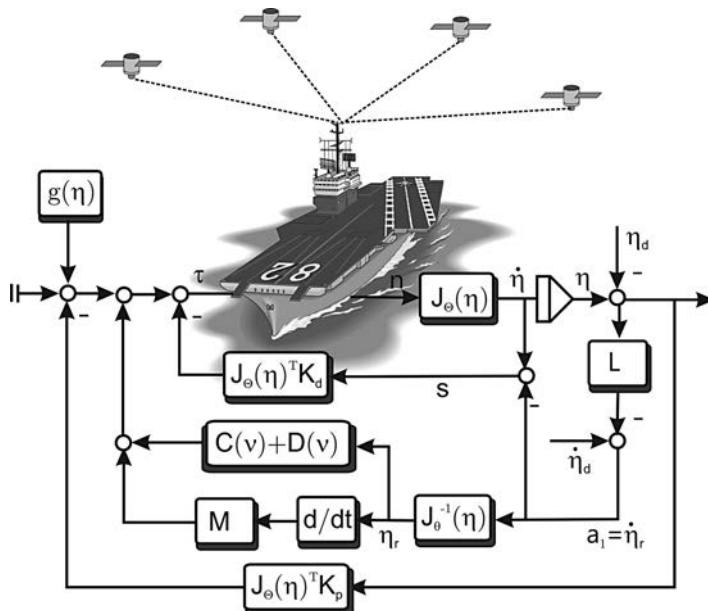


Figure 13.18 Nonlinear MIMO backstepping controller for 6 DOF trajectory-tracking control.

where $\mathbf{K}_d > 0$. This results in

$$\dot{V}_2 = -\mathbf{s}^\top (\mathbf{D}^*(\mathbf{v}, \boldsymbol{\eta}) + \mathbf{K}_d)\mathbf{s} - \tilde{\boldsymbol{\eta}}^\top \mathbf{K}_p \mathbf{\Lambda} \tilde{\boldsymbol{\eta}}$$

Since V_2 is positive definite and \dot{V}_2 is negative definite it follows from Theorem A.3 that the equilibrium point $(\tilde{\boldsymbol{\eta}}, \mathbf{s}) = (\mathbf{0}, \mathbf{0})$ is GES. In addition, it follows from convergence of $\mathbf{s} \rightarrow \mathbf{0}$ and $\tilde{\boldsymbol{\eta}} \rightarrow \mathbf{0}$ that $\dot{\tilde{\boldsymbol{\eta}}} \rightarrow \mathbf{0}$.

Vectorial Backstepping in 3 DOF

Vectorial backstepping in 3 DOF (surge, sway and yaw) is a special case of the general 6 DOF solution which can be applied for surface vessels. Typical applications are stationkeeping and low-speed maneuvering of ships, semi-submersibles and high-speed craft (see Figure 13.19).

In this case the Euler angle transformation matrix $\mathbf{J}_\Theta(\boldsymbol{\eta})$ reduces to (see (2.40) in Section 2.2):

$$\mathbf{J}_\Theta(\boldsymbol{\eta}) \in \mathbb{R}^{6 \times 6} \rightarrow \mathbf{R}(\psi) \in SO(3) \quad (13.408)$$

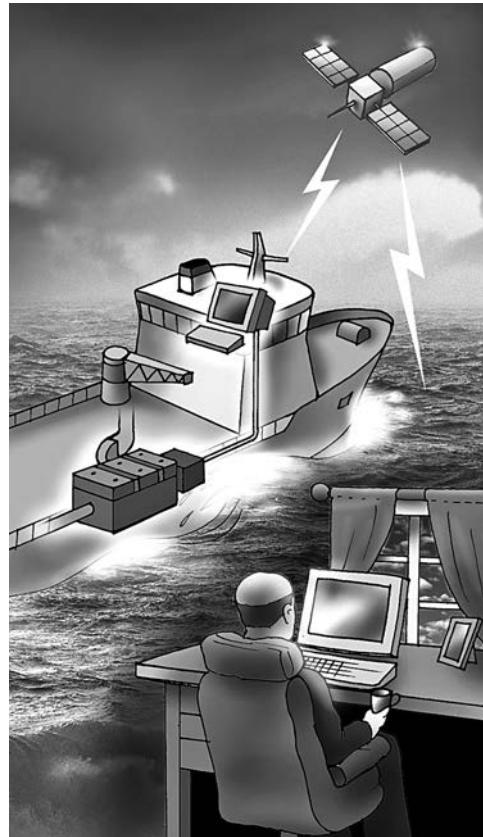


Figure 13.19 Dynamic positioning of a supply vessel using measurements from a global navigation satellite system. Illustration by Bjarne Stenberg/Department of Marine Technology, NTNU.

which is the rotation matrix in yaw. This implies that

$$\mathbf{R}^{-1}(\psi) = \mathbf{R}^\top(\psi) \quad (13.409)$$

The equations of motion (13.394) therefore become

$$\mathbf{M}^*(\psi)\ddot{\eta} + \mathbf{C}^*(v, \psi)\dot{\eta} + \mathbf{D}^*(v, \psi)\dot{\eta} = \mathbf{R}(\psi)\tau \quad (13.410)$$

where the gravitational and buoyancy forces are assumed to outbalance each other such that $\mathbf{g}(\eta) = \mathbf{0}$, and

$$\begin{aligned}\mathbf{M}^*(\psi) &= \mathbf{R}(\psi)\mathbf{M}\mathbf{R}^\top(\psi) \\ \mathbf{C}^*(v, \psi) &= \mathbf{R}(\psi)[\mathbf{C}(v) - \mathbf{M}\mathbf{R}^\top(\psi)\dot{\mathbf{R}}(\psi)]\mathbf{R}^\top(\psi) \\ \mathbf{D}^*(v, \psi) &= \mathbf{R}(\psi)\mathbf{D}(v)\mathbf{R}^\top(\psi)\end{aligned}$$

13.3.8 Case Study: MIMO Backstepping Design with Acceleration Feedback for Fully Actuated Ships

The results of the previous section can be extended to include acceleration feedback. A surface vessel in surge, sway and yaw will be used to illustrate the design procedure. For simplicity a PD control law will be designed. Integral action can easily be included by using adaptive backstepping or integral augmentation techniques as explained in Sections 13.3.4 and 13.3.5.

Consider the 3 DOF maneuvering model:

$$\dot{\eta} = \mathbf{R}(\psi)v \quad (13.411)$$

$$\mathbf{M}\dot{v} + \mathbf{C}(v)v + \mathbf{D}(v)v = \tau \quad (13.412)$$

where

$$\mathbf{M} = \begin{bmatrix} m - X_{\dot{u}} & 0 & 0 \\ 0 & m - Y_{\dot{v}} & mx_g - Y_{\dot{r}} \\ 0 & mx_g - N_{\dot{v}} & I_z - N_{\dot{r}} \end{bmatrix} \quad (13.413)$$

Conventional accelerometers measure linear accelerations along the body axes. Hence, the signals \dot{u} and \dot{v} can be fed back using the control law

$$\tau = \tau_{PD} - \mathbf{K}_m\dot{v} - \mathbf{C}_m(v)v \quad (13.414)$$

where

$$\mathbf{K}_m = \begin{bmatrix} K_{11} & K_{12} & 0 \\ K_{21} & K_{22} & 0 \\ K_{31} & K_{32} & 0 \end{bmatrix} \quad (13.415)$$

$$\mathbf{C}_m(v) = \begin{bmatrix} 0 & 0 & -K_{21}u - K_{22}v \\ 0 & 0 & K_{11}u + K_{12}v \\ K_{21}u + K_{22}v & -K_{11}u - K_{12}v & 0 \end{bmatrix} \quad (13.416)$$

The expression for $\mathbf{C}_m(\mathbf{v})$ in (13.414) is based on (6.43). Consequently, the system inertia matrix after acceleration feedback becomes

$$\mathbf{H} = \mathbf{M} + \mathbf{K}_m = \begin{bmatrix} m - X_{\dot{u}} + K_{11} & K_{12} & 0 \\ K_{21} & m - Y_{\dot{v}} + K_{22} & mx_g - Y_{\dot{r}} \\ K_{31} & mx_g - N_{\dot{v}} + K_{32} & I_z - N_{\dot{r}} \end{bmatrix} \quad (13.417)$$

and

$$\mathbf{C}_H(\mathbf{v}) = \mathbf{C}(\mathbf{v}) + \mathbf{C}_m(\mathbf{v}) \quad (13.418)$$

The feedback term $\mathbf{C}_m(\mathbf{v})\mathbf{v}$ is necessary to ensure that

$$\mathbf{s}^\top [\dot{\mathbf{H}}^*(\psi) - 2\mathbf{C}_H^*(\mathbf{v}, \psi)]\mathbf{s} = 0, \quad \mathbf{s} \neq \mathbf{0} \quad (13.419)$$

where

$$\mathbf{H}^*(\psi) = \mathbf{R}(\psi)\mathbf{H}\mathbf{R}^\top(\psi) \quad (13.420)$$

$$\mathbf{C}_H^*(\mathbf{v}, \psi) = \mathbf{R}(\psi)[\mathbf{C}_H(\mathbf{v}) - \mathbf{H}\mathbf{R}^\top(\psi)\dot{\mathbf{R}}(\psi)]\mathbf{R}^\top(\psi) \quad (13.421)$$

The control law (13.414) gives us some flexibility since the acceleration feedback terms K_{11} , K_{12} , K_{21} , K_{22} , K_{31} and K_{32} can be chosen such that $\mathbf{H} = \mathbf{H}^\top > 0$. A symmetric inertia matrix is obtained by requiring that

$$\mathbf{K}_m = \begin{bmatrix} K_{11} & K_{12} & 0 \\ K_{21} & K_{22} & 0 \\ K_{31} & K_{32} & 0 \end{bmatrix} := \begin{bmatrix} X_{\dot{u}} + \Delta K_{11} & 0 & 0 \\ 0 & Y_{\dot{v}} + \Delta K_{22} & 0 \\ 0 & N_{\dot{v}} - Y_{\dot{r}} & 0 \end{bmatrix} \quad (13.422)$$

where ΔK_{11} and ΔK_{22} can be treated as additional design parameters for the mass in the x and y directions. The resulting expression is

$$\mathbf{H} = \begin{bmatrix} m + \Delta K_{11} & 0 & 0 \\ 0 & m + \Delta K_{22} & mx_g - Y_{\dot{r}} \\ 0 & mx_g - Y_{\dot{r}} & I_z - N_{\dot{r}} \end{bmatrix} \quad (13.423)$$

If $\Delta K_{11} = \Delta K_{22}$, the mass in the x and y directions is equal. Hence, the PID controller will be independent of the heading angle, which is advantageous when tuning dynamic positioning systems, for instance.

The resulting dynamics after acceleration feedback is

$$\mathbf{H}\dot{\mathbf{v}} + \mathbf{C}_H(\mathbf{v})\mathbf{v} + \mathbf{D}(\mathbf{v})\mathbf{v} = \boldsymbol{\tau}_{\text{PD}} \quad (13.424)$$

Consider the CLF:

$$V_1 = \frac{1}{2}\mathbf{z}_1^\top \mathbf{K}_p \mathbf{z}_1, \quad \mathbf{z}_1 = \boldsymbol{\eta}_d - \boldsymbol{\eta} \quad (13.425)$$

$$V_2 = V_1 + \frac{1}{2}\mathbf{v}^\top \mathbf{H}\mathbf{v} \quad (13.426)$$

where V_1 and V_2 represent the potential and kinetic energy, respectively.

New State Variables

Assume that the reference trajectories $\eta_d^{(3)}$, $\ddot{\eta}_d$, $\dot{\eta}_d$ and η_d are smooth and bounded. A virtual reference trajectory is defined as

$$\dot{\eta}_r := \dot{\eta}_d - \Lambda \tilde{\eta}, \quad v_r := R^\top(\psi) \dot{\eta}_r \quad (13.427)$$

where $\tilde{\eta} = \eta - \eta_d$ is the tracking error and $\Lambda > 0$ is a diagonal design matrix. Furthermore, let

$$s = \dot{\eta} - \dot{\eta}_r = \dot{\tilde{\eta}} + \Lambda \tilde{\eta} \quad (13.428)$$

The marine craft dynamics (13.411)–(13.424) can be transformed to

$$H^*(\psi) \ddot{\eta} + C_H^*(v, \psi) \dot{\eta} + D^*(v, \psi) \eta = R(\psi) \tau_{PD} \quad (13.429)$$

Hence,

$$\begin{aligned} H^*(\psi) \dot{s} &= -C_H^*(v, \psi) s - D^*(v, \psi) s + R(\psi) \tau_{PD} \\ &\quad - H^*(\psi) \ddot{\eta}_r - C_H^*(v, \psi) \dot{\eta}_r - D^*(v, \psi) \eta_r \end{aligned} \quad (13.430)$$

or equivalently

$$H^*(\psi) \dot{s} + C_H^*(v, \psi) s + D^*(v, \psi) s = R(\psi) [\tau_{PD} - H \dot{v}_r - C_H(v) v_r - D(v) v_r] \quad (13.431)$$

Step 1: Consider the error dynamics:

$$\dot{\eta} - \dot{\eta}_d = R(\psi) [v - v_d] \quad (13.432)$$

Let $R(\psi)v$ be the virtual control vector $R(\psi)v := s + \alpha_1$. The position error dynamics can therefore be written

$$\begin{aligned} \dot{\tilde{\eta}} &= R(\psi) [v - v_d] \\ &= s + \alpha_1 - R(\psi) v_d, \quad \{\alpha_1 = \dot{\eta}_r = \dot{\eta}_d - \Lambda \tilde{\eta}, \quad \dot{\eta}_d = R(\psi) v_d\} \\ &= -\Lambda \tilde{\eta} + s \end{aligned} \quad (13.433)$$

Hence,

$$V_1 = \frac{1}{2} \tilde{\eta}^\top K_p \tilde{\eta}, \quad K_p = K_p^\top > 0 \quad (13.434)$$

and

$$\begin{aligned} \dot{V}_1 &= \tilde{\eta}^\top K_p \dot{\tilde{\eta}} \\ &= \tilde{\eta}^\top K_p (-\Lambda \tilde{\eta} + s) \\ &= -\tilde{\eta}^\top K_p \Lambda \tilde{\eta} + s^\top K_p \tilde{\eta} \end{aligned} \quad (13.435)$$

Step 2: In the second step, a CLF is motivated by *pseudo-kinetic energy* is chosen according to

$$V_2 = \frac{1}{2} s^\top H^*(\psi) s + V_1 \quad (13.436)$$

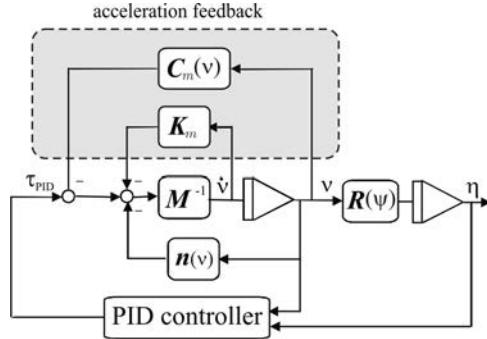


Figure 13.20 Acceleration feedback and PID controller.

$$\begin{aligned}
 \dot{V}_2 &= s^\top H^*(\psi) \dot{s} + \frac{1}{2} s^\top \dot{H}^*(\psi) s + \dot{V}_1 \\
 &= s^\top [-C_H^*(v, \psi) s - D^*(v, \psi) s + R(\psi) [\tau_{PD} - H \dot{v}_r - C_H(v) v_r - D(v) v_r]] \\
 &\quad + \frac{1}{2} s^\top \dot{H}^*(\psi) s - \tilde{\eta}^\top K_p \Lambda \tilde{\eta} + s^\top K_p \tilde{\eta}
 \end{aligned} \tag{13.437}$$

Using the skew-symmetric property $s^\top [\dot{H}^*(\psi) - 2C_H^*(v, \psi)]s = 0$ yields

$$\begin{aligned}
 \dot{V}_2 &= s^\top R(\psi) [\tau_{PD} - H \dot{v}_r - C_H(v) v_r - D(v) v_r + R^\top(\psi) K_p \tilde{\eta}] \\
 &\quad - s^\top D^*(v, \psi) s - \tilde{\eta}^\top K_p \Lambda \tilde{\eta}
 \end{aligned} \tag{13.438}$$

Consequently, the 3 DOF control law

$$\tau_{PD} = H \dot{v}_r + C_H(v) v_r + D(v) v_r - R^\top(\psi) [K_p \tilde{\eta} + K_d s] \tag{13.439}$$

results in

$$\dot{V}_2 = -s^\top (D^*(v, \psi) + K_d) s - \tilde{\eta}^\top K_p \Lambda \tilde{\eta}$$

Since V_2 is positive definite and \dot{V}_2 is negative definite it follows that the equilibrium point $(\tilde{\eta}, s) = (\mathbf{0}, \mathbf{0})$ is GES. Moreover, convergence of $s \rightarrow \mathbf{0}$ and $\tilde{\eta} \rightarrow \mathbf{0}$ implies that $\dot{\tilde{\eta}} \rightarrow \mathbf{0}$. The PD controller can easily be replaced by a PID controller (see Sections 13.3.4 and 13.3.5). In this case only UGAS is guaranteed.

13.3.9 Case Study: Nonlinear Separation Principle for PD Controller-Observer Design

For the motion control systems presented so far, slowly varying environmental forces have been compensated for by adding integral action in the controller. In this section it is demonstrated how a globally converging observer and a PD control law plus a nonlinear term of observer bias estimates can be combined to compensate for slowly varying environmental disturbances (Loria *et al.*, 2000). Moreover, the

integral term is removed in the controller and replaced by a bias estimate. The stability proof is based on a *separation principle*, which holds for nonlinear systems. The separation principle is theoretically supported by results on cascaded nonlinear systems and standard Lyapunov theory, and it is validated in practice by experimentation with a model ship.

The controller–observer is designed in three steps:

1. Design a UGAS state estimator.
2. Design the control law as if the whole state \mathbf{x} and bias term \mathbf{b} were known (measured) and free of noise.
3. Implement the control law with the observer estimates $\hat{\mathbf{x}}$ and $\hat{\mathbf{b}}$ and show that the observer–controller error dynamics is GAS.

The stability proof of this approach requires that the separation principle hold for nonlinear systems. The method in this section relies on Lyapunov theorems for stability of cascaded time-varying systems to prove UGAS (Panteley and Loria, 1998).

Cascaded Systems

Consider the time-varying systems Σ_1 and Σ_2 (Loria *et al.*, 2000):

$$\Sigma_1 : \dot{\mathbf{x}}_1 = \mathbf{f}_1(t, \mathbf{x}_1) + \mathbf{G}(t, \mathbf{x})\mathbf{x}_2 \quad (13.440)$$

$$\Sigma_2 : \dot{\mathbf{x}}_2 = \mathbf{f}_2(t, \mathbf{x}_2) \quad (13.441)$$

where $\mathbf{x}_1 \in \mathbb{R}^n$, $\mathbf{x}_2 \in \mathbb{R}^m$ and $\mathbf{x} = [\mathbf{x}_1^\top, \mathbf{x}_2^\top]^\top$. The function $\mathbf{f}_1(t, \mathbf{x}_1)$ is continuously differentiable in (t, \mathbf{x}_1) , while $\mathbf{f}_2(t, \mathbf{x}_2)$ and $\mathbf{G}(t, \mathbf{x})$ are continuous in their arguments, and locally Lipschitz. The two subsystems Σ_1 and Σ_2 will represent the controller and observer error dynamics, respectively, while $\mathbf{G}(t, \mathbf{x})\mathbf{x}_2$ is the interaction term coupling these two subsystems together. A growth rate condition on $\mathbf{G}(t, \mathbf{x})$ is needed in order to prevent the controller error dynamics Σ_1 from becoming unstable when the true states are replaced with observer estimates.

The cascaded system (13.440)–(13.441) can be proven to be UGAS by reformulating Theorems 1 and 2 in Panteley and Loria (1998) according to:

Theorem 13.1 (UGAS for Cascaded Systems)

The cascaded system (13.440)–(13.441) is UGAS if Assumptions A1–A3 are satisfied:

A1. The system

$$\dot{\mathbf{x}}_1 = \mathbf{f}_1(t, \mathbf{x}_1) \quad (13.442)$$

is UGAS with a Lyapunov function $V(t, \mathbf{x}_1)$, $V : \mathbb{R}_{\geq 0} \times \mathbb{R}^n \rightarrow \mathbb{R}_{\geq 0}$, positive definite, that is $V(t, \mathbf{0}) = \mathbf{0}$ and $V(t, \mathbf{x}_1) > 0$ for all $\mathbf{x}_1 \neq \mathbf{0}$, and proper (radially unbounded), which satisfies

$$\left\| \frac{\partial V}{\partial \mathbf{x}_1} \right\| \|\mathbf{x}_1\| \leq c_1 V(t, \mathbf{x}_1), \quad \forall \|\mathbf{x}_1\| \geq \mu \quad (13.443)$$

where $c_1, \mu > 0$. It is also assumed that $(\partial V / \partial \mathbf{x}_1)(t, \mathbf{x}_1)$ is bounded uniformly in t for all $\|\mathbf{x}_1\| < \mu$; that is there exists a constant $c_2 > 0$ such that for all $t \geq t_0 \geq 0$:

$$\left\| \frac{\partial V}{\partial \mathbf{x}_1} \right\| \leq c_2, \quad \forall \|\mathbf{x}_1\| \leq \mu \quad (13.444)$$

A2. The function $\mathbf{G}(t, \mathbf{x})$ satisfies

$$\|\mathbf{G}(t, \mathbf{x})\| \leq \theta_1(\|\mathbf{x}_2\|) + \theta_2(\|\mathbf{x}_2\|) \|\mathbf{x}_1\| \quad (13.445)$$

where $\theta_1, \theta_2 : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}_{\geq 0}$ are continuous.

A3. Equation $\dot{\mathbf{x}}_2 = \mathbf{f}_2(t, \mathbf{x}_2)$ is UGAS, and for all $t_0 \geq 0$:

$$\int_{t_0}^{\infty} \|\mathbf{x}_2(t)\| dt \leq \phi(\|\mathbf{x}_2(t_0)\|) \quad (13.446)$$

where the function $\phi(\cdot)$ is a class \mathcal{K} function.

DP Control System

Consider the nonlinear DP model:

$$\dot{\boldsymbol{\eta}} = \mathbf{R}(\psi)\mathbf{v} \quad (13.447)$$

$$\mathbf{M}\dot{\mathbf{v}} + \mathbf{D}\mathbf{v} = \boldsymbol{\tau} + \mathbf{R}^\top(\psi)\mathbf{b} \quad (13.448)$$

$$\dot{\mathbf{b}} = \mathbf{0} \quad (13.449)$$

$$\mathbf{y} = \boldsymbol{\eta} + \boldsymbol{\eta}_w \quad (13.450)$$

where $\mathbf{b} \in \mathbb{R}^3$ is a bias term representing slowly varying environmental forces and $\mathbf{y} \in \mathbb{R}^3$ represent the measurements. Instead of using integral action to compensate for \mathbf{b} , a PD controller

$$\boldsymbol{\tau} = -\mathbf{R}^\top(\psi)\mathbf{K}_p\mathbf{e} - \mathbf{K}_d\mathbf{v} - \mathbf{R}^\top(\psi)\mathbf{b}, \quad \mathbf{e} = \boldsymbol{\eta} - \boldsymbol{\eta}_d \quad (13.451)$$

can be implemented under the assumption that \mathbf{b} is known (perfect compensation) and $\dot{\boldsymbol{\eta}}_d = \mathbf{0}$. However, it is impossible to measure \mathbf{b} so a state observer is needed. For this purpose the passive observer (11.128)–(11.132) in Section 11.4.1 can be used to generate estimates of $\boldsymbol{\eta}$, \mathbf{v} and \mathbf{b} , and at the same time provide wave filtering. Application of a nonlinear separation principle implies that the controller can be implemented using the estimated states $\hat{\boldsymbol{\eta}}$, $\hat{\mathbf{v}}$ and $\hat{\mathbf{b}}$; that is

$$\boldsymbol{\tau} = -\mathbf{R}^\top(\psi)\mathbf{K}_p\hat{\mathbf{e}} - \mathbf{K}_d\hat{\mathbf{v}} - \mathbf{R}^\top(\psi)\hat{\mathbf{b}}, \quad \hat{\mathbf{e}} = \hat{\boldsymbol{\eta}} - \boldsymbol{\eta}_d \quad (13.452)$$

The proof needed to show that the passive observer with the controller (13.452) is UGAS is done in three steps corresponding to Assumptions A1–A3 in Theorem 13.1.

Step 1: Observer Error Dynamics

Since the observer error dynamics

$$\Sigma_2 : \dot{\mathbf{x}}_2 = \mathbf{f}_2(t, \mathbf{x}_2) \quad (13.453)$$

is UGES and consequently UGAS when considering the passive observer in Section 11.4.1, there exist positive constants λ_1 and λ_2 such that

$$\|\mathbf{x}_2(t)\| \leq \lambda_1 \|\mathbf{x}_2(t_0)\| e^{-\lambda_2(t-t_0)} \quad (13.454)$$

Therefore Assumption A3 in Theorem 13.1 is satisfied with $\phi(\|\mathbf{x}_2(t_0)\|) = (\lambda_1/\lambda_2) \|\mathbf{x}_2(t_0)\|$.

Step 2: Regulator Error Dynamics

The *full-state* feedback controller (13.451) when applied to (13.447)–(13.448) results in

$$\dot{\mathbf{e}} = \mathbf{R}(\psi)\mathbf{v} \quad (13.455)$$

$$\mathbf{M}\dot{\mathbf{v}} + (\mathbf{D} + \mathbf{K}_d)\mathbf{v} + \mathbf{R}^\top(\psi)\mathbf{K}_p\mathbf{e} = \mathbf{0} \quad (13.456)$$

This system is GAS according to LaSalle–Krasovskii's theorem since

$$V = \frac{1}{2} (\mathbf{v}^\top \mathbf{M}\mathbf{v} + \mathbf{e}^\top \mathbf{K}_p\mathbf{e}) > 0, \quad \forall \mathbf{v} \neq \mathbf{0}, \mathbf{e} \neq \mathbf{0} \quad (13.457)$$

and

$$\dot{V} = -\mathbf{v}^\top (\mathbf{D} + \mathbf{K}_d)\mathbf{v} \leq 0 \quad (13.458)$$

This implies that the first condition on the system $\dot{\mathbf{x}}_1 = f_1(t, \mathbf{x}_1)$, Assumption A1 in Theorem 13.1, is satisfied. Next, a constant c_1 is easily found by considering

$$\left\| \frac{\partial V}{\partial \mathbf{x}_1} \right\| \|\mathbf{x}_1\| \leq \max\{m_M, k_M, 1\} \|\mathbf{x}_1\|^2, \quad \forall \|\mathbf{x}_1\| \geq \mu \quad (13.459)$$

where $m_M = \lambda_{\max}(\mathbf{M})$ and $k_M = \lambda_{\max}(\mathbf{K}_p)$. Hence, (13.443) is satisfied with

$$c_1 = \frac{\max\{m_M, k_M, 1\}}{\min\{m_m, k_m, 1\}} \quad (13.460)$$

where $m_m = \lambda_{\min}(\mathbf{M})$ and $k_m = \lambda_{\min}(\mathbf{K}_p)$. Also from (13.459) it is clear that (13.444) is satisfied by

$$c_2 = \max\{m_M, k_M, 1\}\mu \quad (13.461)$$

Step 3: Growth Rate Condition

Finally, it can be shown that the growth rate condition (13.445) on \mathbf{x}_1 , Assumption A2 in Theorem 13.1, is satisfied by choosing $\theta_1 = \text{constant}$ and $\theta_2 = 0$ such that

$$\|\mathbf{G}(t, \mathbf{x})\| \leq \theta_1 (\|\mathbf{x}_2\|) \quad (13.462)$$

The details in this analysis is found in Loria *et al.* (2000).

Experimental Results

The nonlinear controller (13.452) and passive observer of Section 11.4.1 have been tested experimentally using a model ship. In this experiment wind and wave forces were generated using a fan and a wave maker. More details regarding the experiment are found in Loria *et al.* (2000).

In the experiments the desired position and heading of the ship during DP were chosen as

$$x_d = 208 \text{ m} \quad (13.463)$$

$$y_d = 334 \text{ m} \quad (13.464)$$

$$\psi_d = -150 \text{ deg} \quad (13.465)$$

The experiment was carried out for a ship scale 1:70, but the results have been transformed to full scale. The development of the experiment is as follows:

1. During the first 350 seconds there are no environmental forces perturbing the ship.

Comments: From Figure 13.22 it is seen that the bias estimate \hat{b} and the WF estimate $\hat{\eta}_w$ are both approximately zero, as expected, in the first 350 seconds. The nonzero values of \hat{b} are due to the water motion generated by the propellers. It is also seen that the regulation and estimation errors are very small during this phase; see upper plots in Figures 13.21 and 13.23.

2. After 350 seconds wind forces are generated by using a ducted fan directed approximately 30 degrees off the port side bow of the ship.

Comments: When turned on, the fan produces a step input disturbance to the system; notice the peaks in Figures 13.21 and 13.23. This step is an unrealistic situation (in full-scale applications, no abrupt changes in the bias occur). However, it can be generated in the laboratory to show the performance of the observer-based controller. The bias estimates \hat{b} from the observer are used in the output feedback control law to obtain perfect regulation, which verifies the separation principle (see Figure 13.23). Most of the wind force is compensated by the control input, and therefore the regulation errors converge to zero in 100–150 seconds; see the first three plots of Figure 13.21. However, since the wind force is a step, the observer needs some time for the bias estimate to converge to its true value, after which the controller compensates for the bias, hence keeping the ship almost still.

3. After 800 seconds the wave generator is turned on.

Comments: This results in an oscillatory WF motion η_w which builds up over time. The estimated wave frequency motion $\hat{\eta}_w$ is shown in the upper plots of Figure 13.22. Their effect in the position measurements is shown in the upper plots of Figure 13.21. In order to avoid $\hat{\eta}_w$ entering the feedback loop, this signal is filtered out from the position measurement. This results in smooth controls; see the bottom plots of Figure 13.22. The LF estimates are clearly shown in the upper plots of Figure 13.21.

4. After 1700 seconds both the wind and wave generators are turned off.

Comments: Turning off the fan produces a second step input disturbance while the wave-induced motion decays more slowly. It is seen from Figure 13.22 that the bias estimates drop to approximately their initial values in 100–150 seconds while the amplitudes of the WF motion estimates drop quite slowly. Again, almost perfect regulation to zero is obtained as soon as the bias estimates have converged to their true values. This clearly demonstrates the separation principle. In a full-scale implementation the wind force will build up quite slowly. Hence the step inputs do not constitute a problem.

13.3.10 Case Study: Weather Optimal Position Control for Ships and Floating Structures

Conventional DP systems for ships and free-floating rigs are usually designed for stationkeeping by specifying a desired constant position (N_d, E_d) and a desired constant heading angle ψ_d . In order to minimize the ship fuel consumption, the desired heading ψ_d should in many operations be chosen such that the yaw moment is zero. For vessels with port/starboard symmetry, this means that the mean environmental forces due to wind, waves and ocean currents act through the centerline of the vessel. Then the ship must be rotated until the yaw moment is zero.

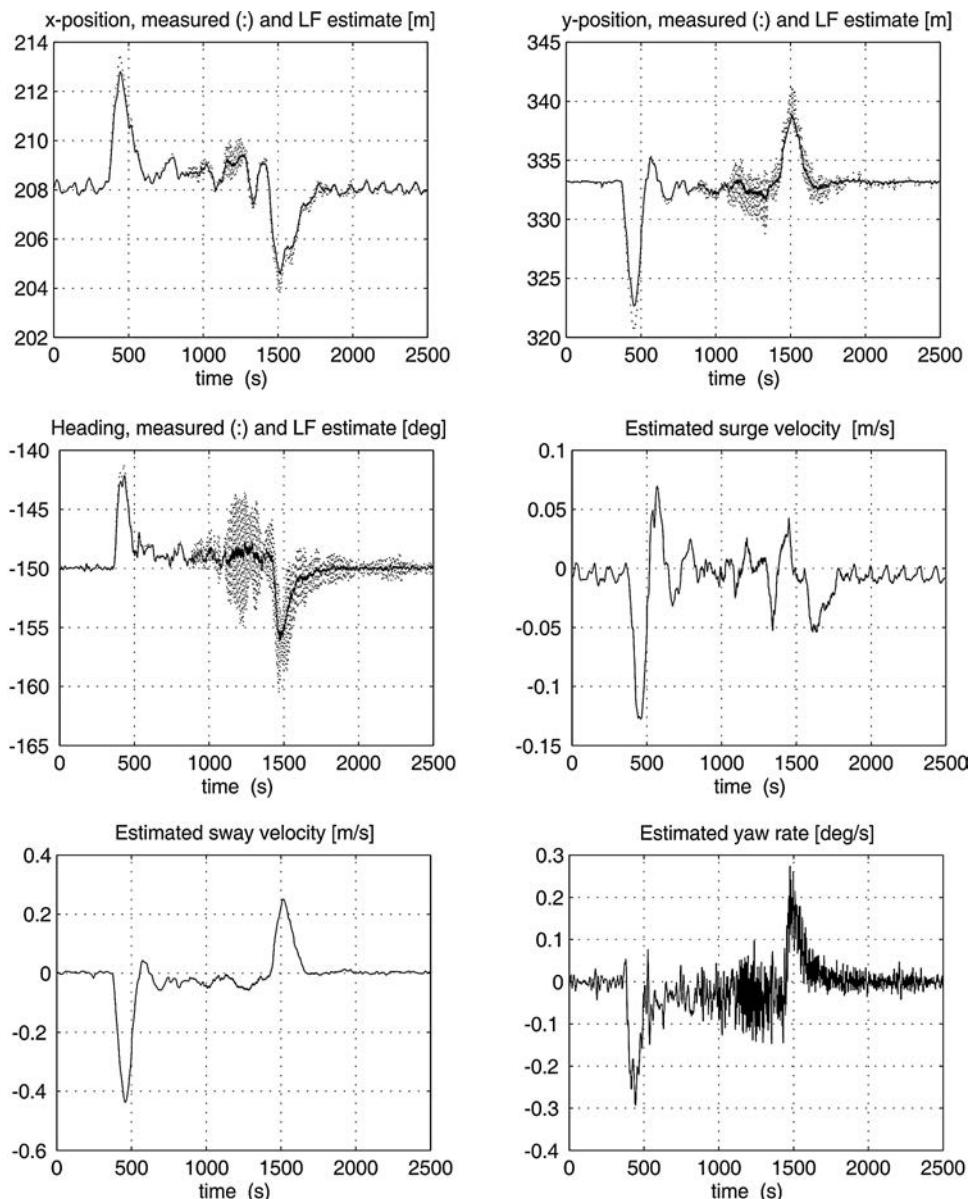


Figure 13.21 Plots 1–3 show the three components of the measurement vector $\mathbf{y} = [x + x_w, y + y_w, \psi + \psi_w]$ and the LF estimates. Plots 4–6 show the estimated LF velocity components $\hat{\mathbf{v}} = [\hat{u}, \hat{v}, \hat{r}]^\top$ versus time.

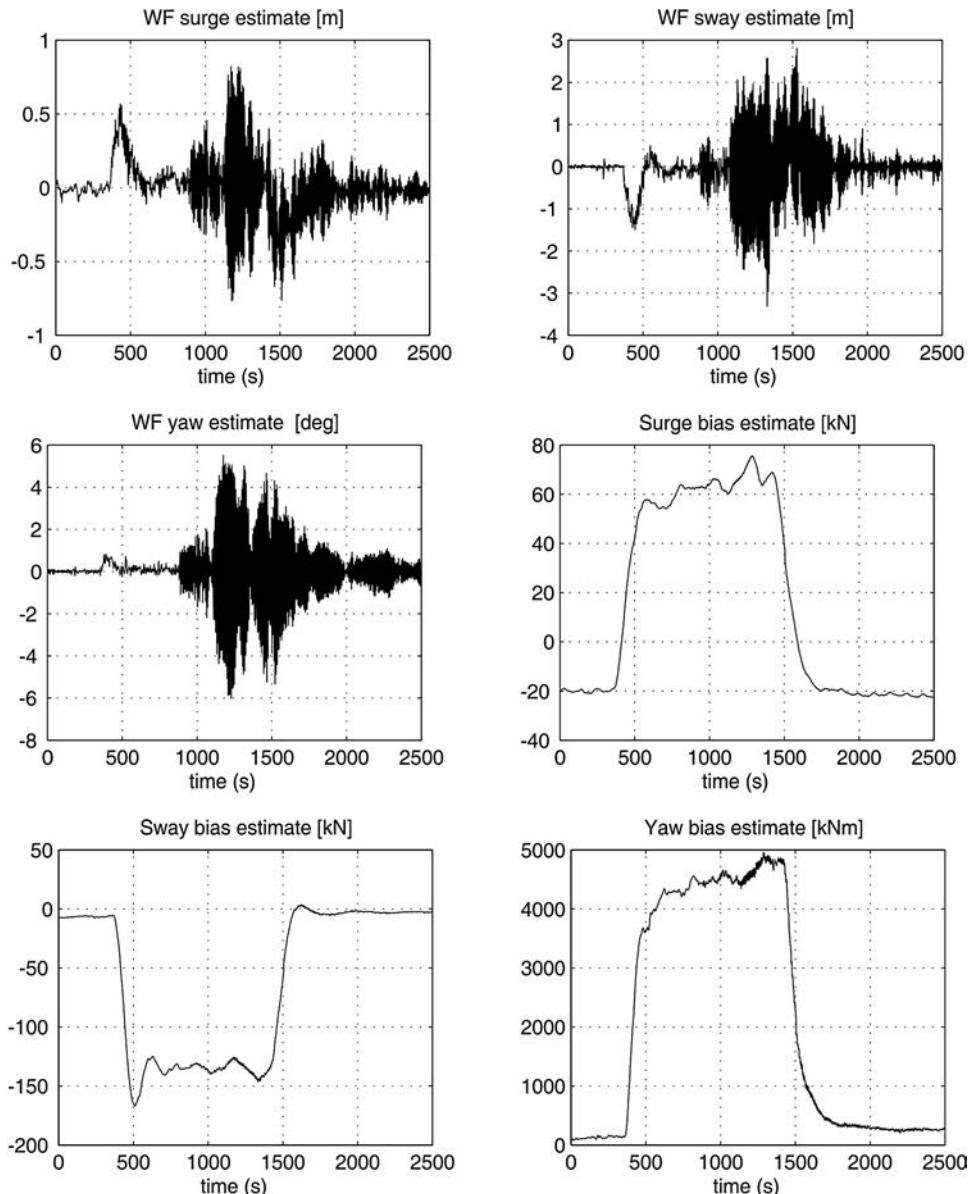


Figure 13.22 Plots 1–3 show the estimated WF motion components $\hat{\eta}_w = [\hat{x}_w, \hat{y}_w, \hat{\psi}_w]^\top$ while plots 4–6 show the bias estimates $\hat{b} = [\hat{b}_1, \hat{b}_2, \hat{b}_3]^\top$ versus time.

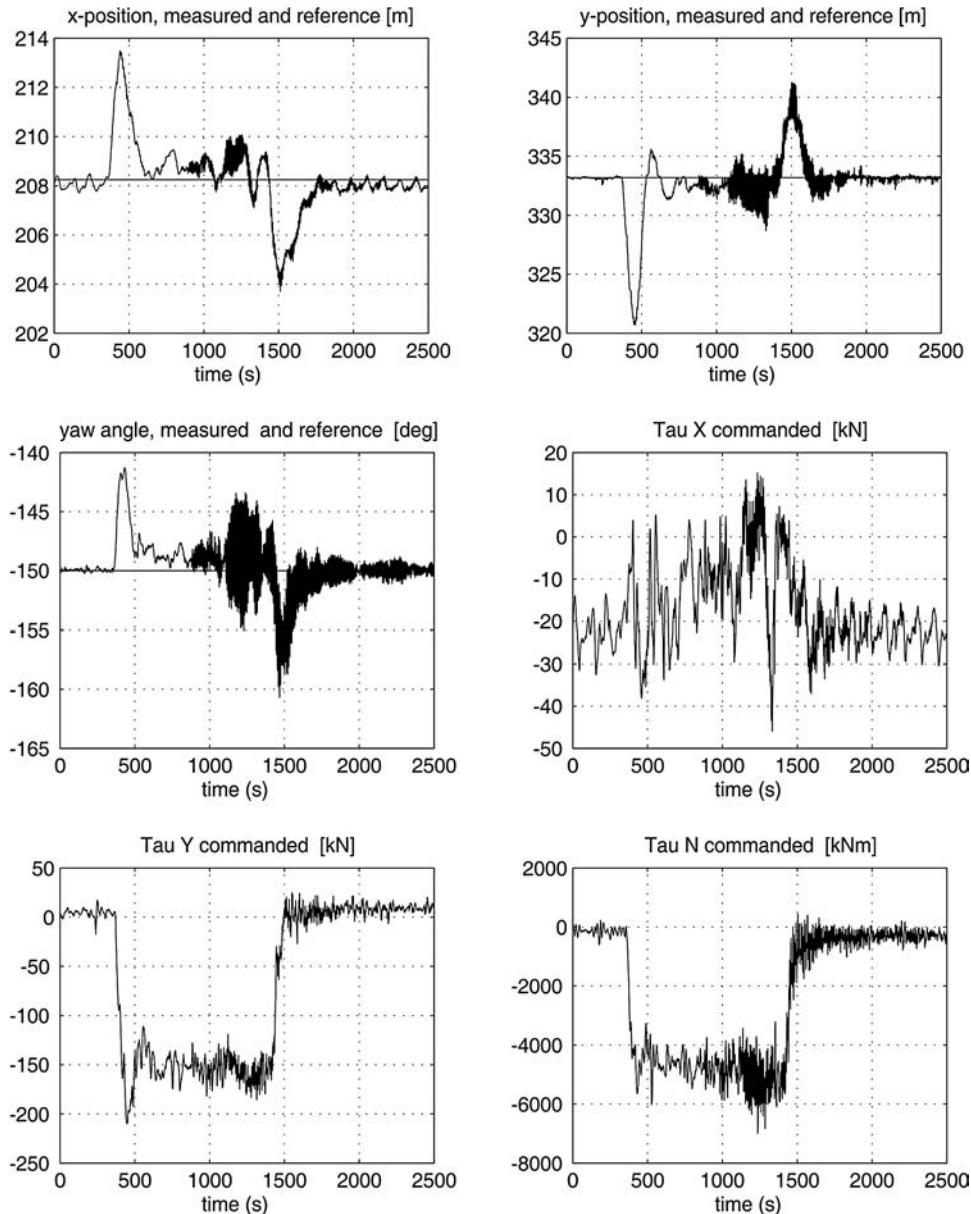


Figure 13.23 Plots 1–3 show the three components of the measured position $y = [x + x_w, y + y_w, \psi + \psi_w]^\top$ together with the desired position $\eta_d = [x_d, y_d, \psi_d]^\top$ while plots 4–6 are the control inputs $\tau = [\tau_1, \tau_2, \tau_3]^\top$ versus time.

Unfortunately, it is impossible to measure or compute the direction of the mean environmental force with sufficient accuracy. Hence, the desired heading ψ_d is usually taken to be the measurement of the mean wind direction, which can be easily measured. In practice, however, this can result in large offsets from the true mean direction of the total environmental force. The main reason for this is the unmeasured ocean current force component and waves that do not coincide with the wind direction. Hence, the DP system can be operated under highly nonoptimal conditions if fuel saving is the issue. A small offset in the optimal heading angle will result in a large use of thrust.

One popular method for computing the weather optimal heading ψ_d is to monitor the resulting thruster forces in the x and y directions. Hence, the bow of the ship can be turned in one direction until the thruster force in the y direction approaches zero. This method is appealing but the main catch in doing this is that the total resulting thruster forces in the x and y directions have to be computed since there are no sensors doing this job directly. The sensors only measure the angular speed and pitch angle of the propellers. Hence, the thrust for each propeller must be computed by using a model of the thruster characteristic, resulting in a fairly rough estimate of the total thruster force in each direction.

Another principle, proposed by Pinkster (1971) and Pinkster and Nienhuis (1986), is to control the x and y positions using a PID feedback controller, in addition to feedback from the yaw velocity, such that the vessel tends toward the optimal heading. This principle, however, requires that the rotation point of the vessel is located a certain distance forward of the center of gravity, or even fore of the bow, and it also puts restrictions on the thruster configuration and the number of thrusters installed.

This section describes the weather optimal position controller (WOPC) by Fossen and Strand (2001). The control objective is that the vessel heading should adjust automatically to the mean environmental forces (wind, waves and ocean currents) such that a minimum amount of energy is used in order to save fuel and reduce NO_x/CO_x emissions without using any environmental sensors. This is particularly useful for shuttle tankers and FPSOs, which can be located at the same position for a long time. Also DP-operated supply vessels that must keep their position for days in loading/off-loading operations have a great WOPC fuel-saving potential.

The ship can be exponentially stabilized on a circle arc with constant radius by letting the bow of the ship point toward the origin of the circle. In order to maintain a fixed position at the same time, a translatory circle center control law is designed. The circle center is translated such that the Cartesian position is constant, while the bow of the ship is automatically turned up against the mean environmental force to obtain weathervaning. This approach is motivated by a pendulum in the gravity field where gravity is the unmeasured quantity. The circular motion of the controlled ship, where the mean environmental force can be interpreted as an unknown force field, copies the dynamics of a pendulum in the gravity field (see Figure 13.24).

3 DOF Equations of Motion using Polar Coordinates

Consider a marine craft in 3 DOF:

$$\dot{\eta} = \mathbf{R}(\psi)\mathbf{v} \quad (13.466)$$

$$\mathbf{M}\dot{\mathbf{v}} + \mathbf{C}(\mathbf{v})\mathbf{v} + \mathbf{D}(\mathbf{v})\mathbf{v} = \boldsymbol{\tau} + \mathbf{w} \quad (13.467)$$

where the North-East positions (N, E) and heading ψ are represented by $\eta = [N, E, \psi]^\top$ and the body-fixed velocities are represented by $\mathbf{v} = [u, v, r]^\top$. It is assumed that $\mathbf{M} = \mathbf{M}^\top > 0$, $\dot{\mathbf{M}} = \mathbf{0}$ and $\mathbf{D}(\mathbf{v}) > 0$. Unmodeled external forces and moment due to wind, ocean currents and waves are lumped together into a body-fixed disturbance vector $\mathbf{w} \in \mathbb{R}^3$ to be interpreted later.

The Cartesian coordinates (N, E) are related to the *polar coordinates* by

$$N = N_0 + \rho \cos(\gamma), \quad E = E_0 + \rho \sin(\gamma) \quad (13.468)$$

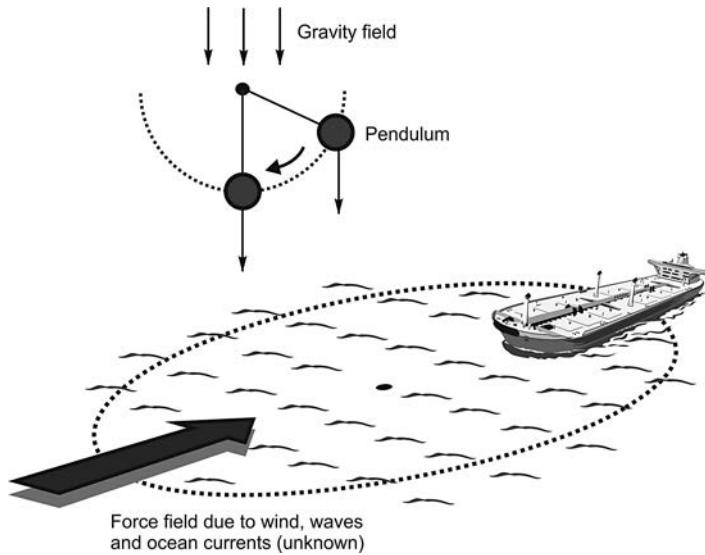


Figure 13.24 The principle of WOPC using the equivalence to a pendulum in the gravity field where gravity is the unmeasured quantity.

where (N_0, E_0) is the origin of a circle with radius ρ and polar angle γ given by

$$\rho = \sqrt{(N - N_0)^2 + (E - E_0)^2} \quad \gamma = \text{atan2}((E - E_0), (N - N_0)) \quad (13.469)$$

Time differentiation of (13.468) yields

$$\dot{N} = \dot{N}_0 + \dot{\rho} \cos(\gamma) - \rho \sin(\gamma)\dot{\gamma} \quad (13.470)$$

$$\dot{E} = \dot{E}_0 + \dot{\rho} \sin(\gamma) + \rho \cos(\gamma)\dot{\gamma} \quad (13.471)$$

Define the state vectors:

$$\mathbf{p}_0 := [N_0, E_0]^\top, \quad \mathbf{x} := [\rho, \gamma, \psi]^\top \quad (13.472)$$

From (13.470) and (13.471) a new kinematic relationship can be derived in terms of the vectors \mathbf{p}_0 and \mathbf{x} as

$$\dot{\eta} = \mathbf{R}(\gamma)\mathbf{H}(\rho)\dot{\mathbf{x}} + \mathbf{L}\dot{\mathbf{p}}_0 \quad (13.473)$$

$$\mathbf{H}(\rho) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \rho & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{L} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} \quad (13.474)$$

From (13.473) the Cartesian kinematics (13.466) can be replaced by a differential equation for the polar coordinates:

$$\dot{\mathbf{x}} = \mathbf{T}(\mathbf{x})\mathbf{v} - \mathbf{T}(\mathbf{x})\mathbf{R}^\top(\psi)\mathbf{L}\dot{\mathbf{p}}_0 \quad (13.475)$$

$$\mathbf{T}(\mathbf{x}) = \mathbf{H}^{-1}(\rho) \underbrace{\mathbf{R}^\top(\gamma)\mathbf{R}(\psi)}_{\mathbf{R}^\top(\gamma-\psi)} \quad (13.476)$$

Note that the conversion between Cartesian and polar coordinates is only a local diffeomorphism, since the radius must be kept larger than a minimum value, that is $\rho > \rho_{\min} > 0$, in order to avoid the singular point $\rho = 0$.

Marine Craft Model Transformation

The marine craft model (13.467) can be expressed in polar coordinates by using (13.475) and substituting

$$\mathbf{v} = \mathbf{T}^{-1}(\mathbf{x})\dot{\mathbf{x}} + \mathbf{R}^\top\mathbf{L}\dot{\mathbf{p}}_0 \quad (13.477)$$

$$\dot{\mathbf{v}} = \mathbf{T}^{-1}(\mathbf{x})\ddot{\mathbf{x}} + \dot{\mathbf{T}}^{-1}(\mathbf{x})\dot{\mathbf{x}} + \mathbf{R}^\top\mathbf{L}\ddot{\mathbf{p}}_0 + \dot{\mathbf{R}}^\top\mathbf{L}\dot{\mathbf{p}}_0 \quad (13.478)$$

such that

$$\begin{aligned} \mathbf{M}\dot{\mathbf{v}} + \mathbf{C}(\mathbf{v})\mathbf{v} + \mathbf{D}(\mathbf{v})\mathbf{v} &= \boldsymbol{\tau} + \mathbf{w} \\ \Updownarrow \rho > 0 \\ \mathbf{M}_x(\mathbf{x})\ddot{\mathbf{x}} + \mathbf{C}_x(\mathbf{v}, \mathbf{x})\dot{\mathbf{x}} + \mathbf{D}_x(\mathbf{v}, \mathbf{x})\dot{\mathbf{x}} &= \mathbf{T}^{-\top}[\mathbf{q}(\mathbf{v}, \mathbf{x}, \dot{\mathbf{p}}_0, \ddot{\mathbf{p}}_0) + \boldsymbol{\tau} + \mathbf{w}] \end{aligned} \quad (13.479)$$

where

$$\mathbf{M}_x(\mathbf{x}) = \mathbf{T}^{-\top}(\mathbf{x})\mathbf{M}\mathbf{T}^{-1}(\mathbf{x})$$

$$\mathbf{C}_x(\mathbf{v}, \mathbf{x}) = \mathbf{T}^{-\top}(\mathbf{x}) (\mathbf{C}(\mathbf{v}) - \mathbf{M}\mathbf{T}^{-1}(\mathbf{x})\dot{\mathbf{T}}(\mathbf{x})) \mathbf{T}^{-1}(\mathbf{x})$$

$$\mathbf{D}_x(\mathbf{v}, \mathbf{x}) = \mathbf{T}^{-\top}(\mathbf{x})\mathbf{D}(\mathbf{v})\mathbf{T}^{-1}(\mathbf{x})$$

$$\mathbf{q}(\mathbf{v}, \mathbf{x}, \dot{\mathbf{p}}_0, \ddot{\mathbf{p}}_0) = -\mathbf{M}\mathbf{R}^\top(\psi)\mathbf{L}\ddot{\mathbf{p}}_0 - \mathbf{M}\dot{\mathbf{R}}^\top(\psi)\mathbf{L}\dot{\mathbf{p}}_0 - [\mathbf{C}(\mathbf{v}) + \mathbf{D}(\mathbf{v})]\mathbf{R}^\top(\psi)\mathbf{L}\dot{\mathbf{p}}_0$$

Here $\mathbf{M}_x(\mathbf{x})$, $\mathbf{C}_x(\mathbf{v}, \mathbf{x})$ and $\mathbf{D}_x(\mathbf{v}, \mathbf{x})$ can be shown to satisfy

$$\mathbf{M}_x(\mathbf{x}) = \mathbf{M}_x^\top(\mathbf{x}) > 0, \quad \mathbf{D}_x(\mathbf{v}, \mathbf{x}) > 0, \quad \forall \mathbf{x}$$

The marine craft dynamics also satisfy the skew-symmetric property:

$$\mathbf{z}^\top (\dot{\mathbf{M}}_x - 2\mathbf{C}_x) \mathbf{z} = 0, \quad \forall \mathbf{z}, \mathbf{x} \quad (13.480)$$

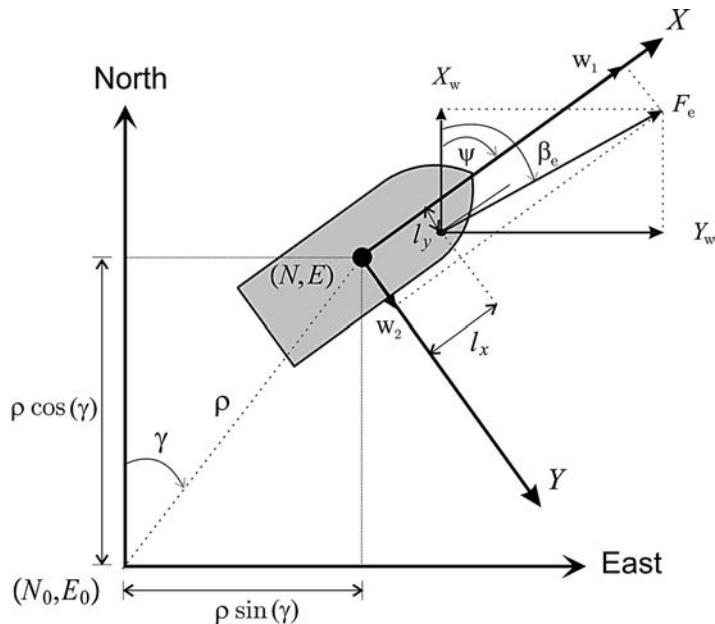


Figure 13.25 Environmental force F_e decomposed into components w_1 and w_2 .

Weather Optimal Control Objectives

The steady-state LF motion of the craft and also the craft's equilibrium position depend on the *unknown* environmental forces acting on the hull. Let the environmental forces due to *wind*, *waves* and *ocean currents* be represented by:

- A slowly varying mean force F_e that attacks the craft at a point (l_x, l_y) in body-fixed coordinates.
- A slowly varying mean direction β_e relative to the Earth-fixed frame (see Figure 13.25).

The WF motion is assumed to be filtered out of the measurements by using a wave filter (see Chapter 11). Since there are no sensors that can be used to measure (F_e, β_e) and (l_x, l_y) with sufficient accuracy, it is impossible to use feedforward from the environmental forces. This leads to the following assumptions:

A1: *The unknown mean environmental force F_e and its direction β_e are assumed to be constant or at least slowly varying.*

A2: *The unknown attack point (l_x, l_y) is constant for each constant F_e .*

Discussion: *These are good assumptions since the motion control system is only supposed to counteract the slowly varying motion components of the environmental forces.*

From Figure 13.25 the body-fixed environmental force vector $\mathbf{w} \in \mathbb{R}^3$ can be expressed as

$$\mathbf{w} = \begin{bmatrix} w_1(\psi) \\ w_2(\psi) \\ w_3(\psi) \end{bmatrix} = \begin{bmatrix} F_e \cos(\beta_e - \psi) \\ F_e \sin(\beta_e - \psi) \\ l_x F_e \sin(\beta_e - \psi) - l_y F_e \cos(\beta_e - \psi) \end{bmatrix} \quad (13.481)$$

Notice that the environmental forces vary with the heading angle ψ of the craft. Consequently,

$$F_e = \sqrt{w_1^2 + w_2^2}, \quad \beta_e = \psi + \tan^{-1}(w_2/w_1) \quad (13.482)$$

The environmental forces X_w and Y_w with attack point $(l_x(\psi), l_y(\psi))$ are shown in Figure 13.25. Note that the attack point will change with the yaw angle ψ . This relationship will be a complicated function of hull and superstructure geometries. However, the weather optimal control objectives can be satisfied by using the following definitions (Fossen and Strand, 2001):

Definition 13.3 (Weather Optimal Heading)

The weather optimal heading angle ψ_{opt} is given by the equilibrium state where the yaw moment $w_3(\psi_{\text{opt}}) = 0$ at the same time as the bow of the craft is turned up against weather (mean environmental forces); that is $w_2(\psi_{\text{opt}}) = 0$. This implies that $\psi_{\text{opt}} = \beta_e$, $l_x(\psi_{\text{opt}}) = \text{constant}$ and $l_y(\psi_{\text{opt}}) = 0$ such that

$$\mathbf{w}(\psi_{\text{opt}}) = \begin{bmatrix} w_1(\psi_{\text{opt}}) \\ w_2(\psi_{\text{opt}}) \\ w_3(\psi_{\text{opt}}) \end{bmatrix} = \begin{bmatrix} -F_e \\ 0 \\ 0 \end{bmatrix}$$

Hence, the mean environmental force attacks the craft in the bow, which has the minimum drag coefficient for water and wind loads.

Definition 13.4 (Weather Optimal Positioning)

Weather optimal positioning (stationkeeping) is defined as the equilibrium state where ψ_{opt} satisfies

$$w_1(\psi_{\text{opt}}) = -F_e, \quad w_2(\psi_{\text{opt}}) = w_3(\psi_{\text{opt}}) = l_y(\psi_{\text{opt}}) = 0 \quad (13.483)$$

and the position $(N, E) = (N_d, E_d)$ is kept constant.

These definitions motivate the following two control objectives:

- **Weather Optimal Heading Control (WOHC):** This is obtained by restricting the craft's movement to a circle with constant radius $\rho = \rho_d$ and at the same time force the craft's bow to point towards the center of the circle until the weather optimal heading angle $\psi = \psi_{\text{opt}}$ is reached (see Figure 13.26). An analogy to this is a pendulum in a gravity field (see Figure 13.24). The position $(N, E) = (N_0 + \rho \cos(\gamma), E_0 + \rho \sin(\gamma))$ will vary until the weather optimal heading angle is reached. This is obtained by specifying the control objective in polar coordinates according to

$$\rho_d = \text{constant}, \quad \dot{\gamma}_d = 0, \quad \psi_d = \pi + \gamma \quad (13.484)$$

Discussion: The requirement $\rho_d = \text{constant}$ implies that the craft follows a circular arc with a constant radius. The second requirement $\dot{\gamma}_d = 0$ implies that the tangential speed $\rho\dot{\gamma}$ is kept small while the last requirement $\psi_d = \pi + \gamma$ ensures that the craft's bow points toward the center of the circle.

- **Weather Optimal Positioning Control (WOPC):** In order to maintain a fixed Earth-fixed position $(N, E) = (N_d, E_d)$, the circle center $\mathbf{p}_0 = [N_0, E_0]^\top$ must be moved simultaneously as control objective O1 is satisfied. This is referred to as *translatory circle center control*.

Nonlinear and Adaptive Control Design

The WOPC positioning controller is derived by using the polar coordinate representation. The backstepping design methodology (Krstic *et al.*, 1995) with extension to integral control (Fossen *et al.*, 2001) is

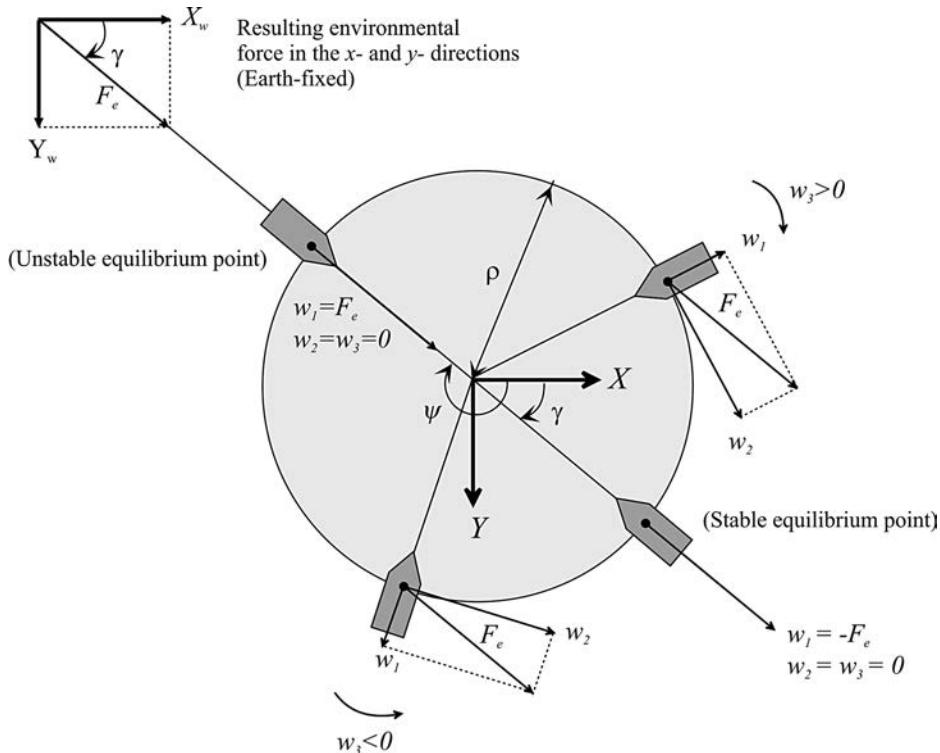


Figure 13.26 Stable and unstable equilibrium points for WOPC.

used to derive the feedback controller (see Section 13.3). Notice that conventional PID control can be used as well. It is assumed that all states can be measured.

The WOPC controller will be derived in three successive steps:

1. *Nonlinear backstepping (PD control):* The ship is forced to move along a circular arc with desired radius ρ_d , with minimum tangential velocity $\rho\dot{\gamma}$ and desired heading ψ_d .
2. *Adaptive backstepping (PID control):* This is necessary to compensate for the unknown environmental force F_e .
3. *Translational control of the circle center:* The circle center (N_0, E_0) is translated such that the ship maintains a constant position (N_d, E_d) even though it is moving along a virtual circular arc. Hence, the captain of the ship will only notice that the ship is rotating a yaw angle ψ about a constant position (N_d, E_d) until the weather optimal heading ψ_{opt} is reached.

Nonlinear Backstepping (PD Control)

A general positioning controller is derived by using vectorial backstepping (Fossen and Grøvlen, 1998). The tracking objective is specified in polar coordinates using a smooth reference trajectory $\mathbf{x}_d = [\rho_d, \gamma_d, \psi_d]^\top \in C^3$ where

$$\mathbf{x}_d, \dot{\mathbf{x}}_d, \ddot{\mathbf{x}}_d \in \mathcal{L}_\infty$$

Since the transformed system (13.479) is of order 2, backstepping is performed in two *vectorial steps*, resulting in a nonlinear PD control law. First, a *virtual reference trajectory* is defined as:

$$\dot{\mathbf{x}}_r := \dot{\mathbf{x}}_d - \boldsymbol{\Lambda} z_1 \quad (13.485)$$

where $z_1 = \mathbf{x} - \mathbf{x}_d$ is the tracking error and $\boldsymbol{\Lambda} > 0$ is a diagonal design matrix. Furthermore, let z_2 denote a measure of tracking defined according to

$$z_2 := \dot{\mathbf{x}} - \dot{\mathbf{x}}_r = \dot{z}_1 + \boldsymbol{\Lambda} z_1 \quad (13.486)$$

From (13.486), the following expressions are obtained:

$$\dot{\mathbf{x}} = z_2 + \dot{z}_r, \quad \ddot{\mathbf{x}} = \dot{z}_2 + \ddot{z}_r \quad (13.487)$$

This implies that the marine craft model (13.479) can be expressed in terms of z_2 , $\dot{\mathbf{x}}_r$ and $\ddot{\mathbf{x}}_r$ as

$$\mathbf{M}_x \dot{z}_2 + \mathbf{C}_x z_2 + \mathbf{D}_x z_2 = \mathbf{T}^{-\top} \boldsymbol{\tau} + \mathbf{T}^{-\top} \mathbf{q}(\cdot) - \mathbf{M}_x \ddot{\mathbf{x}}_r - \mathbf{C}_x \dot{\mathbf{x}}_r - \mathbf{D}_x \dot{\mathbf{x}}_r + \mathbf{T}^{-\top} \mathbf{w} \quad (13.488)$$

Step 1: Let z_1 be the first error variable, which from (13.486) has the dynamics

$$\dot{z}_1 = -\boldsymbol{\Lambda} z_1 + z_2 \quad (13.489)$$

A CLF for the first step is

$$V_1 = \frac{1}{2} z_1^\top \mathbf{K}_p z_1 \quad (13.490)$$

$$\dot{V}_1 = -z_1^\top \mathbf{K}_p \boldsymbol{\Lambda} z_1 + z_1^\top \mathbf{K}_p z_2 \quad (13.491)$$

where $\mathbf{K}_p = \mathbf{K}_p^\top > 0$ is a constant design matrix.

Step 2: In the second step the CLF is motivated by the “pseudo-kinetic energy”:

$$V_2 = V_1 + \frac{1}{2} z_2^\top \mathbf{M}_x z_2, \quad \mathbf{M}_x = \mathbf{M}_x^\top > 0 \quad (13.492)$$

Time differentiation of V_2 along the trajectories of z_1 and z_2 gives

$$\dot{V}_2 = \dot{V}_1 + z_2^\top \mathbf{M}_x \dot{z}_2 + \frac{1}{2} z_2^\top \dot{\mathbf{M}}_x z_2 \quad (13.493)$$

which by substitution of (13.491) and (13.488) gives

$$\begin{aligned} \dot{V}_2 = & -z_1^\top \mathbf{K}_p \boldsymbol{\Lambda} z_1 + \frac{1}{2} z_2^\top (\dot{\mathbf{M}}_x - 2\mathbf{C}_x) z_2 - z_2^\top \mathbf{D}_x z_2 + z_2^\top \mathbf{T}^{-\top} \mathbf{w} \\ & + z_2^\top (\mathbf{K}_p z_1 + \mathbf{T}^{-\top} \boldsymbol{\tau} + \mathbf{T}^{-\top} \mathbf{q}(\cdot) - \mathbf{M}_x \ddot{\mathbf{x}}_r - \mathbf{C}_x \dot{\mathbf{x}}_r - \mathbf{D}_x \dot{\mathbf{x}}_r) \end{aligned} \quad (13.494)$$

By using the property (13.480) and choosing the nonlinear PD control law as

$$\boldsymbol{\tau} = \mathbf{T}^\top (\mathbf{M}_x \ddot{\mathbf{x}}_r + \mathbf{C}_x \dot{\mathbf{x}}_r + \mathbf{D}_x \dot{\mathbf{x}}_r - \mathbf{K}_p z_1 - \mathbf{K}_d z_2) - \mathbf{q}(\cdot) \quad (13.495)$$

where $\mathbf{K}_d > 0$ is a strictly positive design matrix, it is seen that

$$\dot{V}_2 = -z_1^\top \mathbf{K}_p \boldsymbol{\Lambda} z_1 - z_2^\top (\mathbf{K}_d + \mathbf{D}_x) z_2 + z_2^\top \mathbf{T}^{-\top} \mathbf{w} \quad (13.496)$$

Notice that the dissipative term $\mathbf{z}_2^\top \mathbf{D}_x \mathbf{z}_2 > 0, \forall \mathbf{z}_2 \neq \mathbf{0}$ is exploited in the design as it appears in the expression for \dot{V}_2 . With the control law (13.495) the closed-loop dynamics becomes

$$\mathbf{M}_x \dot{\mathbf{z}}_2 + (\mathbf{C}_x + \mathbf{D}_x + \mathbf{K}_d) \mathbf{z}_2 + \mathbf{K}_p \mathbf{z}_1 = \mathbf{T}^{-\top} \mathbf{w} \quad (13.497)$$

Error dynamics: The error dynamics of the resulting system becomes *nonautonomous* since

$$\begin{aligned} \begin{bmatrix} \mathbf{K}_p & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{M}_x \end{bmatrix} \begin{bmatrix} \dot{\mathbf{z}}_1 \\ \dot{\mathbf{z}}_2 \end{bmatrix} &= - \begin{bmatrix} \mathbf{K}_p \mathbf{\Lambda} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{C}_x + \mathbf{D}_x + \mathbf{K}_d \end{bmatrix} \begin{bmatrix} \mathbf{z}_1 \\ \mathbf{z}_2 \end{bmatrix} \\ &\quad + \begin{bmatrix} \mathbf{0}_{3 \times 3} & \mathbf{K}_p \\ -\mathbf{K}_p & \mathbf{0}_{3 \times 3} \end{bmatrix} \begin{bmatrix} \mathbf{z}_1 \\ \mathbf{z}_2 \end{bmatrix} + \begin{bmatrix} \mathbf{0}_{3 \times 1} \\ \mathbf{T}^{-\top} \end{bmatrix} \mathbf{w} \\ &\Downarrow \\ \mathcal{M}(\mathbf{x}) \dot{\mathbf{z}} &= -\mathcal{K}(\mathbf{x}, \mathbf{v}) \mathbf{z} + \mathcal{S} \mathbf{z} + \bar{\mathcal{B}}(\mathbf{x}) \mathbf{w} \end{aligned} \quad (13.498)$$

where the different matrices are defined as

$$\begin{aligned} \mathcal{M}(\mathbf{x}) = \mathcal{M}^T(\mathbf{x}) &= \begin{bmatrix} \mathbf{K}_p & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{M}_x(\mathbf{x}) \end{bmatrix} \\ \mathcal{K}(\mathbf{x}, \mathbf{v}) &= \begin{bmatrix} \mathbf{K}_p \mathbf{\Lambda} & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \mathbf{C}_x(\mathbf{x}, \mathbf{v}) + \mathbf{D}_x(\mathbf{x}, \mathbf{v}) + \mathbf{K}_d \end{bmatrix} > 0 \\ \mathcal{S} = -\mathcal{S}^T &= \begin{bmatrix} \mathbf{0}_{3 \times 3} & \mathbf{K}_p \\ -\mathbf{K}_p & \mathbf{0}_{3 \times 3} \end{bmatrix}, \quad \bar{\mathcal{B}}(\mathbf{x}) = \begin{bmatrix} \mathbf{0}_{3 \times 1} \\ \mathbf{T}^{-\top}(\mathbf{x}) \end{bmatrix} \end{aligned}$$

In the absence of disturbances, $\mathbf{w} \equiv \mathbf{0}$, the origin $\mathbf{z} = \mathbf{0}$ is uniformly locally exponentially stable (ULES) according to Lyapunov. Global results cannot be achieved due to the local diffeomorphism between the Cartesian and polar coordinates; that is the transformation matrix $\mathbf{T}(\mathbf{x})$ is singular for $\rho = 0$.

With disturbances $\mathbf{w} \neq \mathbf{0}$, the closed-loop system is input-to-state stable (ISS). In the next section, it is shown how adaptive backstepping (backstepping with integral action) can be used to obtain ULES for the case of a nonzero disturbance vector $\mathbf{w} \neq \mathbf{0}$.

Adaptive Backstepping (PID Control)

Since the mean disturbance \mathbf{w} is nonzero this will result in a steady-state offset when using the PD controller from the previous section. The craft is, however, restricted to move along a circular arc with \mathbf{w} as a force field. Therefore there will be a stable and an unstable equilibrium point on the circle arc (similar to a pendulum in the gravity field); see Figure 13.24. The stable equilibrium point is given by

$$\mathbf{w} = \phi F_e = [-1, 0, 0]^\top F_e \quad (13.499)$$

Since the disturbance F_e is assumed to be slowly varying, adaptive backstepping can be applied to obtain an integral effect in the system. Thus, in the analysis it will be assumed that $\dot{F}_e = 0$. Let the estimate of F_e be denoted as \hat{F}_e and $\tilde{F}_e = \hat{F}_e - F_e$. An additional step in the derivation of the backstepping control law must be performed in order to obtain an adaptive update law for \hat{F}_e .

Step 3: The adaptive update law is found by adding the square parameter estimation error to V_2 . Consequently,

$$V_3 = V_2 + \frac{1}{2\sigma} \tilde{F}_e^2, \quad \sigma > 0 \quad (13.500)$$

$$\dot{V}_3 = \dot{V}_2 + \frac{1}{\sigma} \dot{\tilde{F}}_e \tilde{F}_e \quad (13.501)$$

The nonlinear control law (13.495) is modified to

$$\tau = \mathbf{T}^\top (\mathbf{M}_x \ddot{\mathbf{x}}_r + \mathbf{C}_x \dot{\mathbf{x}}_r + \mathbf{D}_x \dot{\mathbf{x}}_r - \mathbf{K}_p z_1 - \mathbf{K}_d z_2) - \mathbf{q}(\cdot) - \boldsymbol{\phi} \hat{F}_e \quad (13.502)$$

where the last term $\boldsymbol{\phi} \hat{F}_e$ provides integral action. Hence, the z_2 dynamics becomes

$$\mathbf{M}_x \dot{z}_2 + (\mathbf{C}_x + \mathbf{D}_x + \mathbf{K}_d) z_2 + \mathbf{K}_p z_1 = -\mathbf{T}^{-\top} \boldsymbol{\phi} \hat{F}_e \quad (13.503)$$

This implies that

$$\begin{aligned} \dot{V}_3 &= -z_1^\top \mathbf{K}_p \mathbf{A} z_1 - z_2^\top (\mathbf{K}_d + \mathbf{D}_x) z_2 - z_2^\top \mathbf{T}^{-\top} \boldsymbol{\phi} \hat{F}_e + \frac{1}{\sigma} \dot{\tilde{F}}_e \tilde{F}_e \\ &= -z_1^\top \mathbf{K}_p \mathbf{A} z_1 - z_2^\top (\mathbf{K}_d + \mathbf{D}_x) z_2 + \hat{F}_e (-\boldsymbol{\phi}^\top \mathbf{T}^{-1} z_2 + \frac{1}{\sigma} \dot{\tilde{F}}_e) \end{aligned} \quad (13.504)$$

The adaptive law $\dot{\hat{F}}_e = \dot{\tilde{F}}_e$ is chosen as

$$\dot{\hat{F}}_e = \sigma \boldsymbol{\phi}^\top \mathbf{T}^{-1} z_2, \quad \sigma > 0 \quad (13.505)$$

such that

$$\dot{V}_3 = -z_1^\top \mathbf{K}_p \mathbf{A} z_1 - z_2^\top (\mathbf{K}_d + \mathbf{D}_x) z_2 \leq 0 \quad (13.506)$$

Error Dynamics

The nonautonomous error dynamics for the adaptive backstepping controller can be written

$$\mathcal{M}(\mathbf{x}) \dot{\mathbf{z}} = [-\mathcal{K}(\mathbf{x}, \mathbf{v}) + \mathcal{S}] \mathbf{z} + \mathcal{B}(\mathbf{x}) \tilde{F}_e \quad (13.507)$$

$$\dot{\tilde{F}}_e = -\sigma \mathcal{B}^\top(\mathbf{x}) \mathbf{z} \quad (13.508)$$

where

$$\mathcal{B}(\mathbf{x}) = \begin{bmatrix} \mathbf{0}_{3 \times 1} \\ -\mathbf{T}^{-\top}(\mathbf{x}) \boldsymbol{\phi} \end{bmatrix} \quad (13.509)$$

In order to satisfy *control objective O1*, the controller gains must be chosen according to

$$\mathbf{K}_p = \begin{bmatrix} k_{p1} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & k_{p3} \end{bmatrix}, \quad \mathbf{K}_d = \begin{bmatrix} k_{d1} & 0 & 0 \\ 0 & k_{d2} & 0 \\ 0 & 0 & k_{d3} \end{bmatrix}, \quad \boldsymbol{\Lambda} = \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \lambda_3 \end{bmatrix} \quad (13.510)$$

Notice that $k_{p2} = \lambda_2 = 0$. This implies that the craft is free to move along the circular arc with tangential velocity $\rho\dot{\psi}$. The gain $k_{d2} > 0$ is used to increase the tangential damping (D control) while the radius ρ and heading ψ are stabilized by using PID control.

Semi-Definite Matrices

Since the controller gains k_{p2} and λ_2 are chosen to be zero, the matrices

$$\mathbf{K}_p \geq 0, \quad \boldsymbol{\Lambda} \geq 0 \quad (13.511)$$

are only positive semi-definite. Hence, V_3 is positive semi-definite. Uniform local asymptotic stability (ULAS) of the equilibrium $(z, \tilde{F}_e) = (\mathbf{0}, 0)$ can, however, be proven since the error dynamics (z_1, z_2) is ISS. Consider the reduced order system (z_{1r}, z_2) given by

$$z_{1r} = \mathbf{E}z_1, \quad \mathbf{E} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (13.512)$$

This implies that

$$\begin{aligned} \dot{z}_{1r} &= -\mathbf{E}\boldsymbol{\Lambda}z_1 + \mathbf{E}z_2 \\ &= -(\mathbf{E}\boldsymbol{\Lambda}\mathbf{E}^\top)z_{1r} + \mathbf{E}z_2 \end{aligned} \quad (13.513)$$

Notice that the last step is possible since the diagonal matrices $\boldsymbol{\Lambda} = \text{diag}\{\lambda_1, 0, \lambda_3\}$ satisfy

$$\boldsymbol{\Lambda}\mathbf{E}^\top z_{1r} = \boldsymbol{\Lambda}z_1 \quad (13.514)$$

Hence, the error dynamics (13.507)–(13.508) can be transformed to

$$\mathcal{M}_r(\mathbf{x})\dot{z}_r = [-\mathcal{K}_r(\mathbf{x}, \mathbf{v}) + \mathcal{S}_r]z_r + \mathcal{B}_r(\mathbf{x})\tilde{F}_e \quad (13.515)$$

$$\dot{\tilde{F}}_e = -\sigma\mathcal{B}_r^\top(\mathbf{x})z_r \quad (13.516)$$

where $z_r = [z_{1r}^\top, z_2^\top]^\top$ and

$$\begin{aligned} \mathcal{M}_r(\mathbf{x}) = \mathcal{M}_r^\top(\mathbf{x}) &= \begin{bmatrix} \mathbf{E}\mathbf{K}_p\mathbf{E}^\top & \mathbf{0}_{2 \times 3} \\ \mathbf{0}_{3 \times 2} & \mathbf{M}_x(\mathbf{x}) \end{bmatrix} \\ \mathcal{K}_r(\mathbf{x}, \mathbf{v}) &= \begin{bmatrix} (\mathbf{E}\mathbf{K}_p\mathbf{E}^\top)(\mathbf{E}\boldsymbol{\Lambda}\mathbf{E}^\top) & \mathbf{0}_{2 \times 3} \\ \mathbf{0}_{3 \times 2} & \mathbf{C}_x(\mathbf{x}, \mathbf{v}) + \mathbf{D}_x(\mathbf{x}, \mathbf{v}) + \mathbf{K}_d \end{bmatrix} > 0 \\ \mathcal{S}_r &= -\mathcal{S}_r^\top = \begin{bmatrix} \mathbf{0}_{2 \times 2} & \mathbf{E}\mathbf{K}_p \\ -\mathbf{K}_p\mathbf{E}^\top & \mathbf{0}_{3 \times 3} \end{bmatrix}, \quad \mathcal{B}_r(\mathbf{x}) = \begin{bmatrix} \mathbf{0}_{2 \times 1} \\ \mathbf{T}^{-\top}(\mathbf{x})\boldsymbol{\phi} \end{bmatrix} \end{aligned}$$

where the fact that $\mathbf{K}_p\mathbf{E}^\top z_{1r} = \mathbf{K}_p z_1$ for $\mathbf{K}_p = \text{diag}\{k_{p1}, 0, k_{p3}\}$ has been applied.

Nonautonomous Lyapunov Analysis

Even though the Lyapunov function V_3 corresponding to the states (z_1, z_2) is only positive semi-definite (since \mathbf{K}_p is positive semi-definite) the Lyapunov function V_{3r} corresponding to the new output (z_{1r}, z_2) is positive definite. Using the fact that the closed-loop system governed by (z_1, z_2) is ISS, asymptotic tracking is guaranteed by

$$V_{3r} = \frac{1}{2} \left[z_{1r}^\top (\mathbf{E} \mathbf{K}_p \mathbf{E}^\top) z_{1r} + z_2^\top \mathbf{M}_x z_2 + \frac{1}{\sigma} \tilde{F}_e^2 \right] > 0 \quad (13.517)$$

$$\dot{V}_{3r} = -z_{1r}^\top (\mathbf{E} \mathbf{K}_p \mathbf{E}^\top) (\mathbf{E} \Lambda \mathbf{E}^\top) z_{1r} - z_2^\top (\mathbf{K}_d + \mathbf{D}_x) z_2 \leq 0 \quad (13.518)$$

where $\mathbf{E} \mathbf{K}_p \mathbf{E}^\top > 0$ and $\mathbf{E} \Lambda \mathbf{E}^\top > 0$. Hence, $z_{1r}, z_2, \tilde{F}_e \in \mathcal{L}_\infty$. Notice that \dot{V}_3 is only negative semi-definite since a negative term proportional to $-\tilde{F}_e^2$ is missing in the expression for \dot{V}_3 . ULES of the equilibrium point $(z_{1r}, z_2, \tilde{F}_e) = (\mathbf{0}, \mathbf{0}, 0)$ follows by using the stability theorem of Fossen *et al.* (2001) for nonlinear *nonautonomous* systems (see Appendix A.2.4). Since, the closed-loop system (z_1, z_2) is ISS it is sufficient to consider the reduced order system (z_{1r}, z_2) with output $z_{1r} = \mathbf{E} z_1$ in the stability analysis. According to Appendix A.2.4, we can choose $\mathbf{x}_1 = [z_{1r}^\top, z_2^\top]^\top$, $\mathbf{x}_2 = \tilde{F}_e$, $\mathbf{P} = \sigma$ and $W(\mathbf{x}_1, t) = \frac{1}{2} \mathbf{x}_1^\top \mathbf{x}_1$. Then the equilibrium point $(z_{1r}, z_2, \tilde{F}_e) = (\mathbf{0}, \mathbf{0}, 0)$ of the nonlinear error system (13.507)–(13.508) is ULES since

$$\text{rank}\{(\mathcal{M}_r^{-1}(\mathbf{x}) \mathcal{B}_r(\mathbf{x}))^\top (\mathcal{M}_r^{-1}(\mathbf{x}) \mathcal{B}_r(\mathbf{x}))\} = 1, \forall \mathbf{x}$$

and

$$\begin{aligned} \max \{\|\mathbf{h}(\mathbf{x}_1, t)\|, \|\mathbf{x}_1\|\} &= \max \left\{ \left\| \mathcal{M}_r^{-1}(\mathbf{x}) [-\mathcal{K}_r(\mathbf{x}, \mathbf{v}) + \mathcal{S}_r] \mathbf{x}_1 \right\|, \|\mathbf{x}_1\| \right\} \\ &\leq \rho_1(\|\mathbf{x}_1\|) \|\mathbf{x}_1\| \\ \|\mathbf{B}(\mathbf{x}, t)\| &= \left\| \mathcal{M}_r^{-1}(\mathbf{x}) \mathcal{B}_r(\mathbf{x}) \right\| \leq \rho_2(\|\mathbf{x}_1\|) \\ \max \left\{ \left\| \frac{\partial \mathbf{B}(\mathbf{x}, t)}{\partial t} \right\|, \left\| \frac{\partial \mathbf{B}(\mathbf{x}, t)}{\partial \mathbf{x}_i} \right\| \right\} &= \max \left\{ \left\| \frac{\partial \mathcal{M}_r^{-1}(\mathbf{x}) \mathcal{B}_r(\mathbf{x})}{\partial \mathbf{x}_i} \right\| \right\} \leq \rho_3(\|\mathbf{x}_1\|) \end{aligned}$$

Translational Control of the Circle Center

The adaptive backstepping controller satisfies a control objective O1, that is weather optimal heading control. Weather optimal position control (control objective O2) can be satisfied by moving the circle center $\mathbf{p}_0 = [N_0, E_0]^\top$ online such that the craft maintains a constant position $\mathbf{p} = [N, E]^\top$.

In order to meet the fixed position control objective, an update law for the circle center \mathbf{p}_0 must be derived. The Cartesian Earth-fixed position of the craft is given by

$$\mathbf{p} = \mathbf{L}^\top \boldsymbol{\eta} \quad (13.519)$$

where \mathbf{L} is defined in (13.474). Let $\tilde{\mathbf{p}} = \mathbf{p} - \mathbf{p}_d$ denote the corresponding deviation from the desired position vector $\mathbf{p}_d = [N_d, E_d]^\top$. The desired position can either be constant (regulation) or a smooth time-varying reference trajectory. The control law for translation of the circle center is derived by considering

the following CLF:

$$V_p = \frac{1}{2} \tilde{\mathbf{p}}^\top \tilde{\mathbf{p}} \quad (13.520)$$

$$\dot{V}_p = \tilde{\mathbf{p}}^\top (\dot{\mathbf{p}} - \dot{\mathbf{p}}_d) = \tilde{\mathbf{p}}^\top (\mathbf{L}^\top \dot{\mathbf{q}} - \dot{\mathbf{p}}_d) \quad (13.521)$$

By using (13.473), $\mathbf{L}^\top \mathbf{L} = \mathbf{I}_{2 \times 2}$ and $\dot{\mathbf{x}} = \mathbf{z}_2 + \dot{\mathbf{x}}_r$ it is seen that

$$\begin{aligned} \dot{V}_p &= \tilde{\mathbf{p}}^\top [\mathbf{L}^\top (\mathbf{R}(\gamma) \mathbf{H}(\rho) \dot{\mathbf{x}} + \mathbf{L} \dot{\mathbf{p}}_0) - \dot{\mathbf{p}}_d] \\ &= \tilde{\mathbf{p}}^\top (\dot{\mathbf{p}}_0 - \dot{\mathbf{p}}_d + \mathbf{L}^\top \mathbf{R}(\gamma) \mathbf{H}(\rho) \dot{\mathbf{x}}_r) + \tilde{\mathbf{p}}^\top \mathbf{L}^\top \mathbf{R}(\gamma) \mathbf{H}(\rho) \mathbf{z}_2 \end{aligned} \quad (13.522)$$

Now, by choosing the circle center update law as

$$\dot{\mathbf{p}}_0 = \dot{\mathbf{p}}_d - \mathbf{L}^\top \mathbf{R}(\gamma) \mathbf{H}(\rho) \dot{\mathbf{x}}_r - k_0 \tilde{\mathbf{p}} \quad (13.523)$$

where $k_0 > 0$, it is seen that

$$\dot{V}_p = -k_0 \tilde{\mathbf{p}}^\top \tilde{\mathbf{p}} + \tilde{\mathbf{p}}^\top \mathbf{L}^\top \mathbf{R}(\gamma) \mathbf{H}(\rho) \mathbf{z}_2 \quad (13.524)$$

In (13.524) a cross-term in $\tilde{\mathbf{p}}$ and \mathbf{z}_2 is noted. In order to guarantee that the time derivative of the total system $V_{\text{wopc}} = V_{3r} + V_p$ is negative semi-definite, the weather optimal controller (13.502) must be modified such that the cross-term in (13.524) is canceled.

Weather Optimal Position Control (WOPC)

The cross-terms involving $\tilde{\mathbf{p}}$ and \mathbf{z}_2 in \dot{V}_p can be removed by modifying the nonlinear controller (13.502) to

$$\boldsymbol{\tau} = \mathbf{T}^\top (\mathbf{M}_x \ddot{\mathbf{x}}_r + \mathbf{C}_x \dot{\mathbf{x}}_r + \mathbf{D}_x \dot{\mathbf{x}}_r - \mathbf{K}_p \mathbf{z}_1 - \mathbf{K}_d \mathbf{z}_2) - \mathbf{q}(\cdot) - \boldsymbol{\phi} \hat{F}_e - \mathbf{T}^\top \mathbf{H}^\top(\rho) \mathbf{R}^\top(\gamma) \mathbf{L} \tilde{\mathbf{p}} \quad (13.525)$$

The last term in $\boldsymbol{\tau}$ implies that

$$\dot{V}_{3r} = -\mathbf{z}_{1r}^\top (\mathbf{E} \mathbf{K}_p \mathbf{E}^\top) (\mathbf{E} \boldsymbol{\Lambda} \mathbf{E}^\top) \mathbf{z}_{1r} - \mathbf{z}_{2r}^\top (\mathbf{K}_d + \mathbf{D}_x) \mathbf{z}_2 - \tilde{\mathbf{p}}^\top \mathbf{L}^\top \mathbf{R}(\gamma) \mathbf{H}(\rho) \mathbf{z}_2 \quad (13.526)$$

Consider

$$V_{\text{wopc}} = V_{3r} + V_p \quad (13.527)$$

$$\dot{V}_{\text{wopc}} = -\mathbf{z}_{1r}^\top (\mathbf{E} \mathbf{K}_p \mathbf{E}^\top) (\mathbf{E} \boldsymbol{\Lambda} \mathbf{E}^\top) \mathbf{z}_{1r} - \mathbf{z}_{2r}^\top (\mathbf{K}_d + \mathbf{D}_x) \mathbf{z}_2 - k_0 \tilde{\mathbf{p}}^\top \tilde{\mathbf{p}} \quad (13.528)$$

and therefore the equilibrium point $(\mathbf{z}_{1r}, \mathbf{z}_2, \hat{F}_e, \tilde{\mathbf{p}}) = (\mathbf{0}, \mathbf{0}, 0, \mathbf{0})$ is ULES. The term $\tilde{\mathbf{p}}_0$ is needed in the expression for $\mathbf{q}(\cdot)$. This term is computed from (13.523) as

$$\ddot{\mathbf{p}}_0 = \ddot{\mathbf{p}}_d - k_0(\dot{\mathbf{p}} - \dot{\mathbf{p}}_d) - \mathbf{L}^\top \mathbf{R}(\gamma) \mathbf{H}(\rho) \dot{\mathbf{x}}_r \\ - \mathbf{L}^\top \dot{\mathbf{R}}(\gamma) \mathbf{H}(\rho) \dot{\mathbf{x}}_r - \mathbf{L}^\top \mathbf{R}(\gamma) \dot{\mathbf{H}}(\rho) \dot{\mathbf{x}}_r \quad (13.529)$$

Experiment 1: Weather Optimal Heading Control (WOHC)

The proposed WOHC system has been implemented and tested experimentally using a model ship of scale 1:70. A ducted fan was used to generate wind forces. The length of the model ship is $L_m = 1.19$ m and the mass is $m_m = 17.6$ kg. The experimental results are scaled to full scale by considering a supply vessel with mass $m_s = 4500$ tons using the bis system (see Section 7.2.5).

In the first experiment the ship was allowed to move on the circle arc and the circle center controller (13.523) was turned off; that is $N_0 = \text{constant}$ and $E_0 = \text{constant}$. This is referred to as WOHC. The fixed origin and circle arc are shown in Figure 13.27. Notice that the initial heading is approximately 30 degrees (see Figure 13.28), while the position $(N, E) \approx (13, -43)$. These values are those obtained when the fan was initially directed at 210 degrees in the opposite direction of the ship heading.

After 3000 seconds the fan was slowly rotated to 165 degrees, corresponding to a weather optimal heading of -15 degrees (see Figure 13.28). During this process, the ship starts to move on the circle arc with heading towards the circle center until it is stabilized to its new heading at -15 degrees. The new position on the circle arc is $(N, E) \approx (3, 20)$. This clearly demonstrates that the ship heading converges to the optimal value by copying the dynamics of a pendulum in the gravity field. This is done without using any external wind sensor.

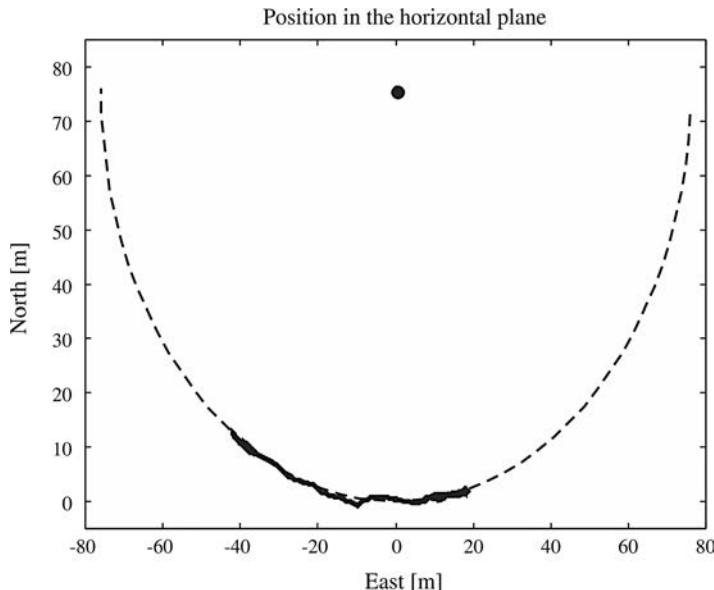


Figure 13.27 WOHC experiment showing the circular motion of the ship when the circle center controller is turned off (WOHC).

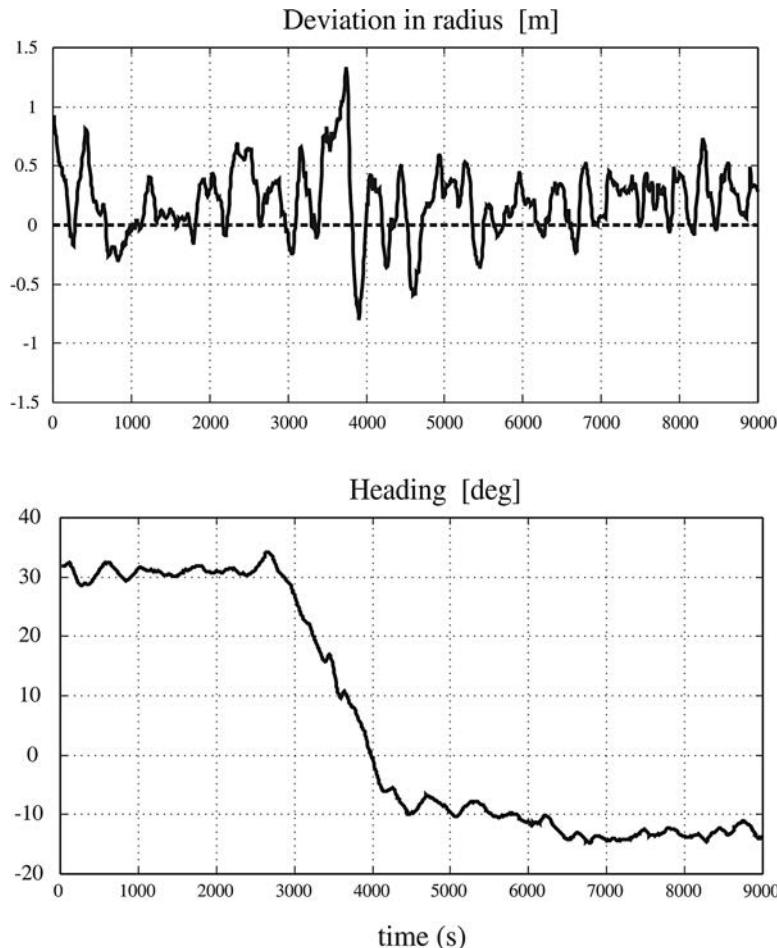


Figure 13.28 WOHC experiment showing the performance of the radius regulator (upper plot) and weather optimal heading (lower plot) versus time (s).

In the next experiment, the circle center is translated online in order to obtain a constant position (N, E).

Experiment 2: Weather Optimal Position Control (WOPC)

In the second experiment the ship should maintain its position by activating the circle center controller (13.5.23). The performance during stationkeeping and translation of the circle is shown in Figures 13.29–13.31. The position controller works within an accuracy of ± 1 m, which is the accuracy of the GNSS system.

Again the weather optimal heading is changed from approximately 23 degrees to 2 degrees but this time without changing the position (N, E) of the ship. The position deviations and the weather optimal heading are shown in Figure 13.30. These values are obtained by moving the fan from an initial angle of 203 degrees to 182 degrees.

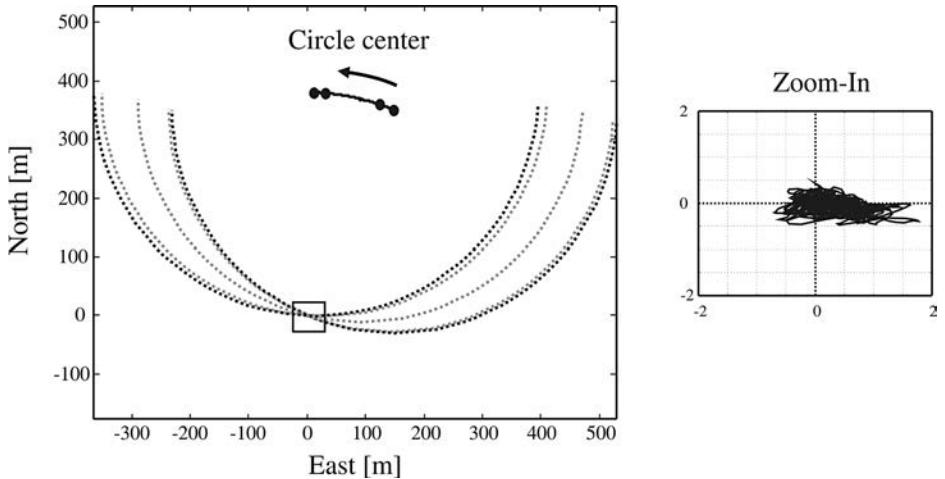


Figure 13.29 WOPC experiment showing how the circle center is moved to obtain stationkeeping to $(N_d, E_d) = (0, 0)$.

13.3.11 Case Study: Heading Autopilot for Ships and Underwater Vehicles

A nonlinear backstepping controller can be designed by writing the autopilot model (7.53) in SISO strict feedback form:

$$\dot{\psi} = r \quad (13.530)$$

$$m\dot{r} + d(r)r = \delta \quad (13.531)$$

where $m = T/K$ and $d(r) = H_N(r)/K$. The only nonlinearity in this model is due to the maneuvering characteristic $H_N(r)$.

In Section 13.3.3 it was shown that the backstepping controller for this system is

$$\delta = m\dot{\alpha}_1 + d(r)r - z_1 - k_2 z_2 - n_2(z_2)z_2 \quad (13.532)$$

$$\alpha_1 = r_d - [k_1 + n_1(z_1)]z_1 \quad (13.533)$$

where $k_1 > 0$ and $k_2 > 0$ are two feedback gains and $n_i(z_i) \geq 0$ ($i = 1, 2$) are two optional nonlinear damping terms, for instance chosen as nondecreasing functions $n_i(z_i) = \kappa_i |z_i|^{n_i}$ with $n_i \geq 1$ and $\kappa_i \geq 0$ ($i = 1, 2$) as design parameters. The following change of coordinates is needed to implement the controller:

$$z_1 = \psi - \psi_d \quad (13.534)$$

$$z_2 = r - \alpha_1 \quad (13.535)$$

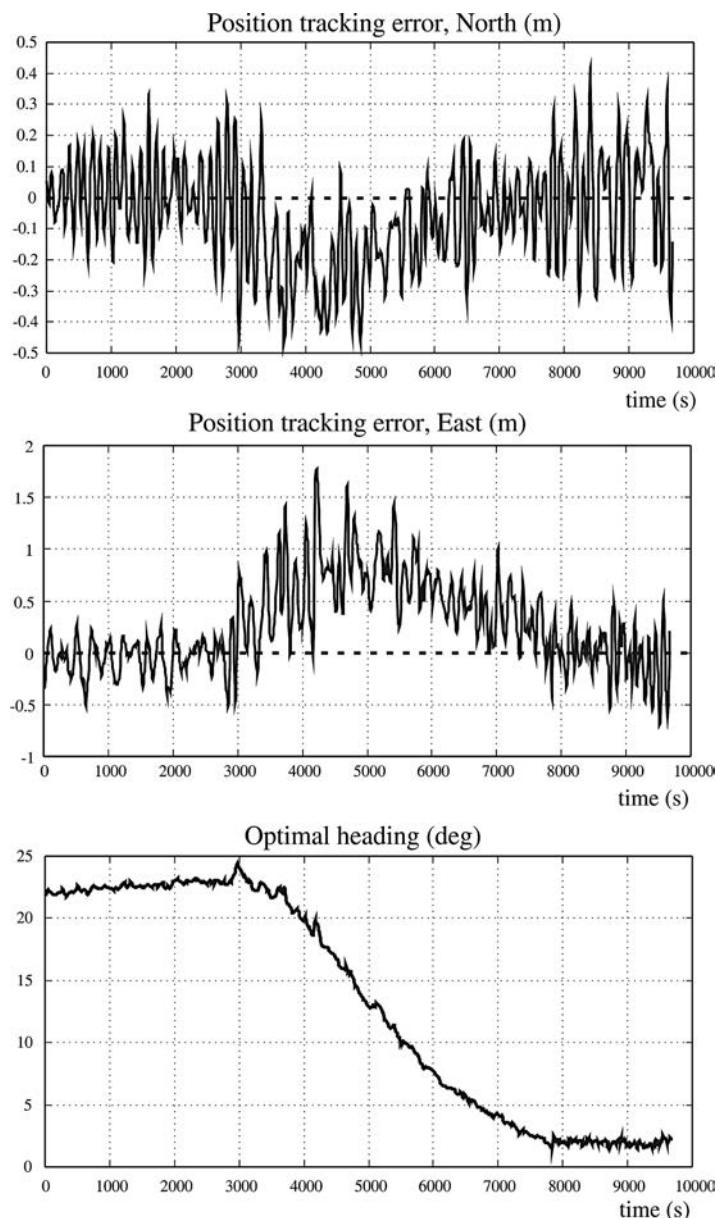


Figure 13.30 WOPC experiment showing the North and East position accuracies (upper plots) and weather optimal heading (lower plot) versus time (seconds). The position accuracy is within ± 1 m while the heading changes from 23 degrees to 2 degrees as the fan is rotated.

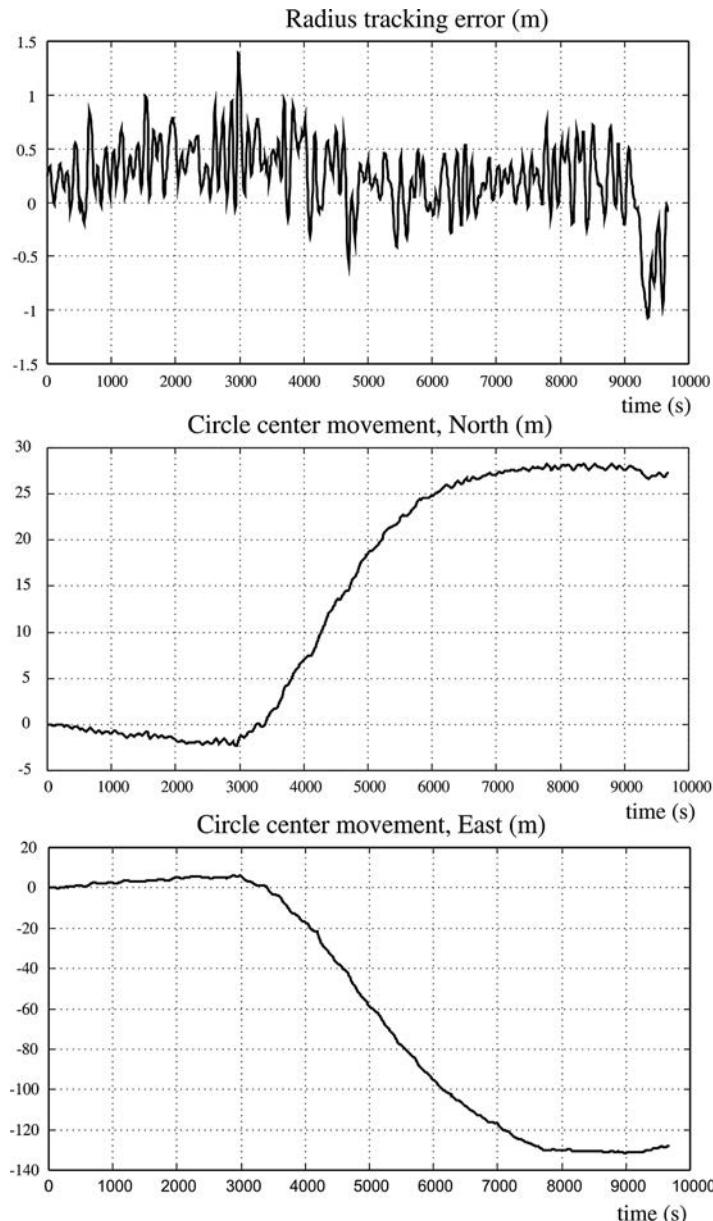


Figure 13.31 WOPC experiment showing the deviation for the radius regulator (upper plot) and the translation of the circle center (N_0 , E_0) (lower plots) versus time in seconds. The radius deviation is within ± 1 m during the rotation of the fan.

The backstepping controller includes a PD term as well as reference feedforward. In addition the nonlinear damping terms $n_i(z_i)$ ($i = 1, 2$) can be used to improve the performance and stability of the closed-loop system.

When using feedback linearization all the nonlinearities in $H_N(r)$ are compensated for. This requires that the dissipative terms are known with good accuracy, which is not true in many cases. The backstepping controller gives more design flexibility with respect to the damping terms. In fact, it is possible to exploit good damping terms such as n_3r^3 and n_1r in $H_N(r)$ instead of canceling them. This is straightforward in setpoint regulation; see Krstic *et al.* (1995), for instance. In trajectory-tracking control, however, it is not clear how good damping with respect to a time-varying reference trajectory should be defined. A discussion on backstepping versus feedback linearization is found in Section 13.3.2.

Extensions to integral action can be done by using the method of Loria *et al.* (1999) and Fossen *et al.* (2001), which is referred to as *backstepping with integral action*. Alternatively, an *integrator augmentation technique* can be applied. Both these methods are described in detail in Sections 13.3.4 and 13.3.5.

The actuator dynamics can be included in the design by using the approach of Fossen and Berge (1997) where backstepping is performed in three steps to include a first-order actuator model.

13.3.12 Case Study: Path-Following Controller for Underactuated Marine Craft

For floating rigs, semi-submersibles and supply vessels, trajectory-tracking control in *surge*, *sway* and *yaw* (3 DOF) is easily achieved since independent control forces and moments are simultaneously available in all degrees of freedom. For slow speed, this is referred to as DP and the craft is controlled by means of tunnel thrusters, azimuths and main propellers. Conventional craft, on the other hand, are usually equipped with one or two main propellers for forward speed control and rudders for turning control. The minimum configuration for waypoint tracking control is one main propeller and a single rudder. This means that only two controls are available, thus rendering the ship underactuated for the task of 3 DOF trajectory-tracking control (see Section 9.4).

Conventional waypoint guidance systems are usually designed by reducing the output space from 3 DOF position and heading to 2 DOF heading and surge (Healey and Marco, 1992). In its simplest form this involves the use of a classical autopilot system where the commanded yaw angle ψ_d is generated such that the *cross-track error* is minimized. A path-following control system is usually designed such that the ship moves forward with reference speed u_d at the same time as the cross-track error to the path is minimized. As a result, ψ_d and u_d are tracked using only two controls.

This section is based on Fossen *et al.* (2003a) and presents a maneuvering controller involving an LOS guidance system and a nonlinear feedback trajectory-tracking controller. The desired output is reduced from (x_d, y_d, ψ_d) to ψ_d and u_d using an LOS projection algorithm. The tracking task $\psi(t) \rightarrow \psi_d(t)$ is then achieved using only one control (normally the rudder), while tracking of the speed assignment u_d is performed by the remaining control (the main propeller). Since we are dealing with segments of straight lines, the LOS projection algorithm will guarantee that the task of path-following is satisfied.

First, an LOS guidance procedure is derived. This includes a projection algorithm and a waypoint switching algorithm. To avoid large bumps in ψ_d when switching, and to provide the necessary derivatives of ψ_d to the controller, the commanded LOS heading is fed through a reference model. Second, a nonlinear 2 DOF tracking controller is derived using the backstepping technique. Three stabilizing functions $\alpha := [\alpha_1, \alpha_2, \alpha_3]^\top$ are defined where α_1 and α_3 are specified to satisfy the tracking objectives in the controlled surge and yaw modes. The stabilizing function α_2 in the uncontrolled sway mode is left as a free design variable. By assigning dynamics to α_2 , the resulting controller becomes a dynamic feedback controller so that $\alpha_2(t) \rightarrow v(t)$ during path following. This is an appealing idea that adds to the extensive theory of backstepping. The presented design technique results in a robust controller for underactuated ships since integral action can be implemented for both path-following and speed control.

Problem Statement

The problem statement is stated as a maneuvering problem with the following two objectives (Skjetne *et al.*, 2004):

LOS Geometric Task: Force the marine craft position $\mathbf{p} = [x, y]^\top$ to converge to a desired path by forcing the course angle χ to converge to (see Section 10.3.2)

$$\chi_d = \text{atan2}(y_{\text{los}} - y, x_{\text{los}} - x) \quad (13.536)$$

where the LOS position $p_{\text{los}} = [x_{\text{los}}, y_{\text{los}}]^\top$ is the point along the path to which the craft should be pointed. Notice that $\psi = \chi - \beta$ and $\psi_d = \chi_d - \beta$ implies that $\bar{\psi} = \bar{\chi}$ when designing the controller.

Dynamic Task: Force the speed u to converge to a desired speed assignment u_d according to

$$\lim_{t \rightarrow \infty} [u(t) - u_d(t)] = 0 \quad (13.537)$$

where u_d is the desired speed composed along the body-fixed x axis.

A conventional trajectory-tracking control system for 3 DOF is usually implemented using a standard PID autopilot in series with an LOS algorithm. Hence, a state-of-the-art autopilot system can be modified to take the LOS reference angle as input (see Figure 12.20). This adds flexibility since the default commercial autopilot system can be used together with the LOS guidance system. The speed can be adjusted manually by the captain or automatically using the path speed profile.

Consider the 3 DOF nonlinear maneuvering model in the following form:

$$\dot{\boldsymbol{\eta}} = \mathbf{R}(\psi)\mathbf{v} \quad (13.538)$$

$$\mathbf{M}\dot{\mathbf{v}} + \mathbf{N}(\mathbf{v})\mathbf{v} = \begin{bmatrix} (1-t)T \\ Y_\delta\delta \\ N_\delta\delta \end{bmatrix} := \begin{bmatrix} \tau_1 \\ Y_\delta\delta \\ \tau_3 \end{bmatrix} \quad (13.539)$$

where $\boldsymbol{\eta} = [N, E, \psi]^\top$, $\mathbf{v} = [u, v, r]^\top$ and

$$\mathbf{R}(\psi) = \begin{bmatrix} \cos(\psi) & -\sin(\psi) & 0 \\ \sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (13.540)$$

The matrices \mathbf{M} and \mathbf{N} take the following form:

$$\mathbf{M} = \begin{bmatrix} m_{11} & 0 & 0 \\ 0 & m_{22} & m_{23} \\ 0 & m_{32} & m_{33} \end{bmatrix} = \begin{bmatrix} m - X_{\dot{u}} & 0 & 0 \\ 0 & m - Y_{\dot{v}} & mx_g - Y_{\dot{r}} \\ 0 & mx_g - N_{\dot{v}} & I_z - N_{\dot{r}} \end{bmatrix}$$

$$\mathbf{N}(\mathbf{v}) = \begin{bmatrix} n_{11} & 0 & 0 \\ 0 & n_{22} & n_{23} \\ 0 & n_{32} & n_{33} \end{bmatrix} = \begin{bmatrix} -X_u & 0 & 0 \\ 0 & -Y_v & mu - Y_r \\ 0 & -N_v & mx_g u - N_r \end{bmatrix}$$

The control force and moment in surge and yaw are denoted τ_1 and τ_3 , respectively, while sway is left uncontrolled. Notice that the rudder angle δ affects the sway equation but it will not be used to actively

control sway. The controller computes τ_1 and τ_3 which can be allocated to thrust T and rudder angle δ using

$$\tau_1 = (1 - t)T \quad (13.541)$$

$$\tau_3 = N_\delta \delta \quad (13.542)$$

where t is the thrust deduction number. This gives

$$T = \frac{1}{1-t} \tau_1 \quad (13.543)$$

$$\delta = \frac{1}{N_\delta} \tau_3 \quad (13.544)$$

Backstepping Design

The design is based on the model (13.538)–(13.539) where $\mathbf{M} = \mathbf{M}^\top > 0$. Define the error signals $z_1 \in S$ and $z_2 \in \mathbb{R}^3$ according to

$$z_1 := \chi - \chi_d = \psi - \psi_d \quad (13.545)$$

$$z_2 := [z_{2,1}, z_{2,2}, z_{2,3}]^\top = \mathbf{v} - \boldsymbol{\alpha} \quad (13.546)$$

where χ_d and its derivatives are provided by proper filtering of the LOS angle, $u_d \in \mathcal{L}_\infty$ is the desired speed and $\boldsymbol{\alpha} := [\alpha_1, \alpha_2, \alpha_3]^\top \in \mathbb{R}^3$ is a vector of stabilizing functions to be specified later. Next, let

$$\mathbf{h} = [0, 0, 1]^\top \quad (13.547)$$

such that

$$\begin{aligned} \dot{z}_1 &= r - r_d = \mathbf{h}^\top \mathbf{v} - r_d \\ &= \alpha_3 + \mathbf{h}^\top z_2 - r_d \end{aligned} \quad (13.548)$$

where $r_d = \dot{\psi}_d$ and

$$\begin{aligned} \mathbf{M}\dot{z}_2 &= \mathbf{M}\dot{\mathbf{v}} - \mathbf{M}\dot{\boldsymbol{\alpha}} \\ &= \boldsymbol{\tau} - \mathbf{N}\mathbf{v} - \mathbf{M}\dot{\boldsymbol{\alpha}} \end{aligned} \quad (13.549)$$

Consider the CLF:

$$V = \frac{1}{2}z_1^2 + \frac{1}{2}z_2^\top \mathbf{M}z_2, \quad \mathbf{M} = \mathbf{M}^\top > 0 \quad (13.550)$$

Differentiating V along the trajectories of z_1 and z_2 yields

$$\begin{aligned} \dot{V} &= z_1 \dot{z}_1 + z_2^\top \mathbf{M} \dot{z}_2 \\ &= z_1(\alpha_3 + \mathbf{h}^\top z_2 - r_d) + z_2^\top (\boldsymbol{\tau} - \mathbf{N}\mathbf{v} - \mathbf{M}\dot{\boldsymbol{\alpha}}) \end{aligned}$$

Choosing the virtual control α_3 as

$$\alpha_3 = -cz_1 + r_d \quad (13.551)$$

while α_1 and α_2 are yet to be defined gives

$$\begin{aligned}\dot{V} &= -cz_1^2 + z_1 \mathbf{h}^\top \mathbf{z}_2 + \mathbf{z}_2^\top (\boldsymbol{\tau} - N\mathbf{v} - \mathbf{M}\dot{\boldsymbol{\alpha}}) \\ &= -cz_1^2 + \mathbf{z}_2^\top (\mathbf{h}z_1 + \boldsymbol{\tau} - N\mathbf{v} - \mathbf{M}\dot{\boldsymbol{\alpha}})\end{aligned}\quad (13.552)$$

Suppose we can assign

$$\boldsymbol{\tau} = \begin{bmatrix} \tau_1 \\ Y_\delta \delta \\ \tau_3 \end{bmatrix} = \mathbf{M}\dot{\boldsymbol{\alpha}} + N\mathbf{v} - \mathbf{K}\mathbf{z}_2 - \mathbf{h}z_1 \quad (13.553)$$

where $\mathbf{K} = \text{diag}\{k_1, k_2, k_3\} > 0$. This results in

$$\dot{V} = -cz_1^2 - \mathbf{z}_2^\top \mathbf{K}\mathbf{z}_2 < 0, \quad \forall z_1 \neq 0, \mathbf{z}_2 \neq \mathbf{0} \quad (13.554)$$

and by standard Lyapunov arguments, this guarantees that (z_1, \mathbf{z}_2) is bounded and converges to zero.

However, notice from (13.553) that it is only possible to prescribe values for τ_1 and τ_3 ; that is

$$\tau_1 = m_{11}\dot{\alpha}_1 + n_{11}u - k_1(u - \alpha_1) \quad (13.555)$$

$$\tau_3 = m_{32}\dot{\alpha}_2 + m_{33}\dot{\alpha}_3 + n_{32}v + n_{33}r - k_3(r - \alpha_3) - z_1 \quad (13.556)$$

Choosing $\alpha_1 = u_d$ clearly solves the dynamic task since the closed-loop surge dynamics becomes

$$m_{11}(\dot{u} - \dot{u}_d) + k_1(u - u_d) = 0 \quad (13.557)$$

The second equation in (13.553) results in a dynamic equality constraint

$$m_{22}\dot{\alpha}_2 + m_{23}\dot{\alpha}_3 + n_{22}v + n_{23}r - k_2(v - \alpha_2) = \frac{Y_\delta}{N_\delta}\tau_3 \quad (13.558)$$

affected by the control input τ_3 . Substituting (13.556) into this expression yields

$$\begin{aligned}\left(m_{22} - \frac{Y_\delta}{N_\delta}m_{32}\right)\dot{\alpha}_2 + \left(m_{23} - \frac{Y_\delta}{N_\delta}m_{33}\right)\dot{\alpha}_3 + \left(n_{22} - \frac{Y_\delta}{N_\delta}n_{32}\right)v + \left(n_{23} - \frac{Y_\delta}{N_\delta}n_{33}\right)r \\ - k_2(v - \alpha_2) + \frac{Y_\delta}{N_\delta}(k_3(r - \alpha_3) + z_1) = 0\end{aligned}$$

Application of $\dot{\alpha}_3 = c^2z_1 - cz_{2,3} + \dot{r}_d$, $\alpha_3 = -cz_1 + r_d$, $v = \alpha_2 + z_{2,2}$ and $r = \alpha_3 + z_{2,3}$ then gives

$$\left(m_{22} - \frac{Y_\delta}{N_\delta}m_{32}\right)\dot{\alpha}_2 = -\left(n_{22} - \frac{Y_\delta}{N_\delta}n_{32}\right)\alpha_2 + \gamma(z_1, \mathbf{z}_2, r_d, \dot{r}_d) \quad (13.559)$$

where

$$\begin{aligned}\gamma(z_1, \mathbf{z}_2, r_d, \dot{r}_d) &= -\left(m_{23} - \frac{Y_\delta}{N_\delta}m_{33}\right)(c^2z_1 - cz_{2,3} + \dot{r}_d) - \left(n_{22} - \frac{Y_\delta}{N_\delta}n_{32}\right)z_{2,2} \\ &\quad - \left(n_{23} - \frac{Y_\delta}{N_\delta}n_{33}\right)(-cz_1 + r_d + z_{2,3}) + k_2z_{2,2} - \frac{Y_\delta}{N_\delta}(k_3z_{2,3} + z_1)\end{aligned}\quad (13.560)$$

The variable α_2 becomes a dynamic state of the controller according to (13.559). Furthermore, $m_{22} > (Y_\delta/N_\delta)m_{32}$ and $n_{22} > (Y_\delta/N_\delta)n_{32}$ imply that (13.559) is a stable differential equation driven

by the converging error signals (z_1, z_2) and the bounded reference signals (r_d, \dot{r}_d) within the expression of $\gamma(\cdot)$. Since $z_{2,2}(t) \rightarrow 0$, it follows that $|\alpha_2(t) - v(t)| \rightarrow 0$ as $t \rightarrow \infty$. The main result is summarized by Theorem 13.2, which is a modification of Fossen *et al.* (2003a).

Theorem 13.2 (LOS Backstepping Controller for Underactuated Craft)

The LOS maneuvering problem for the 3 DOF underactuated craft (13.538)–(13.539) is solved using the control laws

$$\tau_1 = m_{11}\dot{u}_d + n_{11}u - k_1(u - u_d)$$

$$\tau_3 = m_{32}\dot{\alpha}_2 + m_{33}\dot{\alpha}_3 + n_{32}v + n_{33}r - k_3(r - \alpha_3) - z_1$$

where $k_1 > 0$, $k_3 > 0$, $z_1 := \psi - \psi_d$, $z_2 := [u - u_d, v - \alpha_2, r - \alpha_3]^\top$ and

$$\alpha_3 = -cz_1 + r_d, \quad c > 0 \quad (13.561)$$

$$\dot{\alpha}_3 = -c(r - r_d) + \dot{r}_d \quad (13.562)$$

The reference signals u_d , \dot{u}_d , ψ_d , r_d and \dot{r}_d are provided by the LOS guidance system, while α_2 is given by

$$\left(m_{22} - \frac{Y_\delta}{N_\delta}m_{32} \right) \dot{\alpha}_2 = - \left(n_{22} - \frac{Y_\delta}{N_\delta}n_{32} \right) \alpha_2 + \gamma(z_1, z_2, r_d, \dot{r}_d)$$

This results in a UGAS equilibrium point $(z_1, z_2) = (0, \mathbf{0})$ and $\alpha_2 \in \mathcal{L}_\infty$ satisfies

$$\lim_{t \rightarrow \infty} |\alpha_2(t) - v(t)| = 0 \quad (13.563)$$

Remark 13.1

Notice that the smooth reference signal $\psi_d \in \mathcal{L}_\infty$ must be differentiated twice to produce r_d and \dot{r}_d while $u_d \in \mathcal{L}_\infty$ must be differentiated once to give \dot{u}_d . This is most easily achieved by using reference models represented by low-pass filters (see Section 10.2.1).

Proof. The closed-loop equations become

$$\begin{bmatrix} \dot{z}_1 \\ \dot{z}_2 \end{bmatrix} = \begin{bmatrix} -c & \mathbf{h}^\top \\ -\mathbf{M}^{-1}\mathbf{h} & -\mathbf{M}^{-1}\mathbf{K} \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} \quad (13.564)$$

$$\bar{m}\dot{\alpha}_2 = -\bar{n}\alpha_2 + \gamma(z_1, z_2, r_d, \dot{r}_d) \quad (13.565)$$

where

$$\bar{m} = \left(m_{22} - \frac{Y_\delta}{N_\delta}m_{32} \right), \quad \bar{n} = \left(n_{22} - \frac{Y_\delta}{N_\delta}n_{32} \right) \quad (13.566)$$

From the Lyapunov arguments (13.550) and (13.554), the equilibrium $(z_1, z_2) = (0, \mathbf{0})$ of the z subsystem is UGAS. The unforced α_2 subsystem ($\gamma = 0$) is clearly exponentially stable. Since $(z_1, z_2) \in \mathcal{L}_\infty$ and $(r_d, \dot{r}_d) \in \mathcal{L}_\infty$, then $\gamma \in \mathcal{L}_\infty$. This implies that the α_2 subsystem is input-to-state stable from γ to α_2 . This is seen by applying, for instance, $V_2 = \frac{1}{2}\bar{m}\alpha_2^2$ which differentiated along the solutions of α_2 gives $\dot{V}_2 \leq -\frac{1}{2}\bar{n}\alpha_2^2$ for all $|\alpha_2| \geq \frac{2}{\bar{n}}|\gamma(z_1, z_2, r_d, \dot{r}_d)|$. By standard comparison functions, it is then possible to show that for all $|\alpha_2| \geq \frac{2}{\bar{n}}|\gamma(z_1, z_2, r_d, \dot{r}_d)|$ then

$$|\alpha_2(t)| \leq |\alpha_2(0)| e^{-\frac{\bar{n}}{2}t} \quad (13.567)$$

Hence, α_2 converges to the bounded set $\{\alpha_2 : |\alpha_2| \leq \frac{2}{\bar{n}}|\gamma(z_1, z_2, r_d, \dot{r}_d)|\}$ since $z_{2,2}(t) \rightarrow 0$ as $t \rightarrow \infty$.

Case Study: Experiment Performed with the CS2 Model Ship

The proposed controller and guidance system were tested out at the *Marine Cybernetics Laboratory* (MCLab). MCLab is an experimental laboratory for the testing of scale models of ships, rigs, underwater vehicles and propulsion systems. The software is developed by using rapid prototyping techniques and automatic code generation under *Matlab/Simulink*. The target PC onboard the model scale vessels runs the real-time operating system, while experimental results are presented in real time on a host PC.

In the experiment, *CyberShip 2* (CS2) was used. It is a 1:70 scale model of an offshore supply vessel with a mass of 15 kg and a length of 1.255 m. The maximum surge force is approximately 2.0 N while the maximum yaw moment is about 1.5 N m. The MCLab tank is $L \times B \times D = 40 \text{ m} \times 6.5 \text{ m} \times 1.5 \text{ m}$.

Figure 13.32 shows CS2. Three spheres can be seen mounted on the ship, ensuring that its position and orientation can be identified by infrared cameras. Two infrared cameras mounted on a towing carriage currently supply the position and orientation estimates in 6 DOF, but due to a temporary poor calibration the camera measurements vanished when the ship assumed certain yaw angles and regions of the tank. This affected the results of the experiment and also limited the available space for maneuvering. Nevertheless, good results were obtained.

The desired path consists of a total of eight waypoints:

$$\begin{array}{ll} \text{wpt}_1 = (0.372, -0.181) & \text{wpt}_5 = (6.872, -0.681) \\ \text{wpt}_2 = (-0.628, 1.320) & \text{wpt}_6 = (8.372, -0.181) \\ \text{wpt}_3 = (0.372, 2.820) & \text{wpt}_7 = (9.372, 1.320) \\ \text{wpt}_4 = (1.872, 3.320) & \text{wpt}_8 = (8.372, 2.820) \end{array}$$

representing an S-shape. CS2 was performing the maneuver with a constant surge speed of 0.1 m/s. By assuming equal *Froude numbers*, this corresponds to a surge speed of 0.85 m/s for the full scale supply

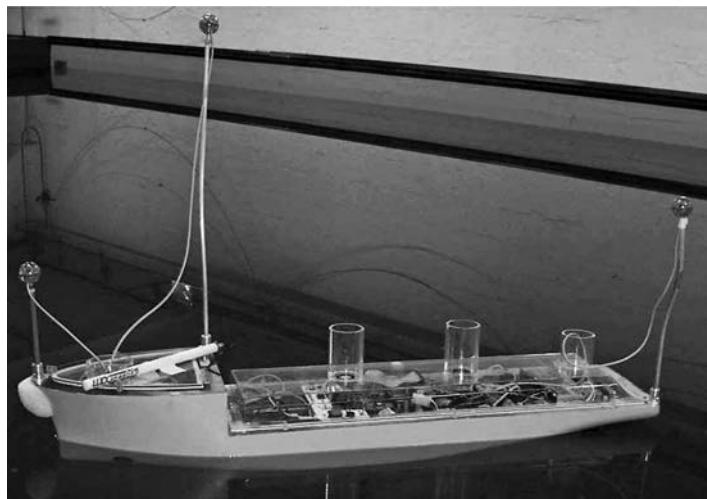


Figure 13.32 CyberShip 2 floating in the MCLab.

ship. A higher speed was not attempted because the consequence of vanishing position measurements at higher speed is quite severe. The controller used

$$\mathbf{M} = \begin{bmatrix} 25.8 & 0 & 0 \\ 0 & 33.8 & 1.0115 \\ 0 & 1.0115 & 2.76 \end{bmatrix}, \quad \mathbf{N}(\nu) = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 7 & 0.1 \\ 0 & 0.1 & 0.5 \end{bmatrix}$$

$$c = 0.75, k_1 = 25, k_2 = 10, k_3 = 2.5$$

In addition, a reference model consisting of three first-order low-pass filters in cascade delivered continuous values of ψ_d , r_d and \dot{r}_d . The ship's initial states were

$$(x_0, y_0, \psi_0) = (-0.69 \text{ m}, -1.25 \text{ m}, 1.78 \text{ rad})$$

$$(u_0, v_0, r_0) = (0.1 \text{ m/s}, 0 \text{ m/s}, 0 \text{ rad/s})$$

Both the ship enclosing circle and the radius of acceptance for all waypoints was set to one ship length. Figure 13.33 shows an *xy* plot of the CS2's position together with the desired geometrical path consisting of straight-line segments. The ship is seen to follow the path very well. To illustrate the effect of the positioning reference system dropping out from time to time, Figure 13.34 is included. It shows the actual heading angle of CS2 alongside the desired LOS angle. The discontinuities in the actual heading angle are due to the camera measurements dropping out. When the measurements return, the heading angle of the ship is seen to converge nicely to the desired angle.

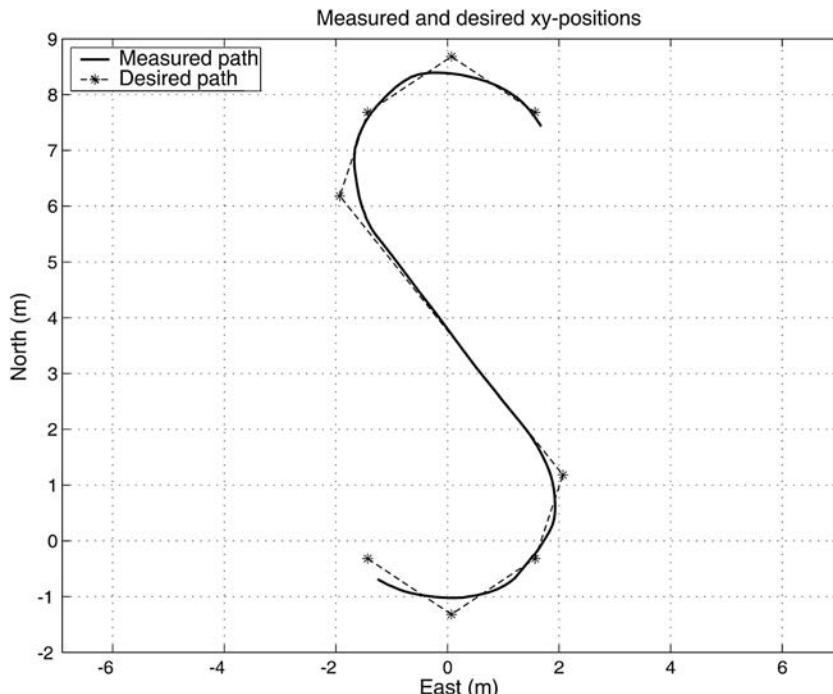


Figure 13.33 *xy* plot of the measured and desired geometrical path during the experiment.

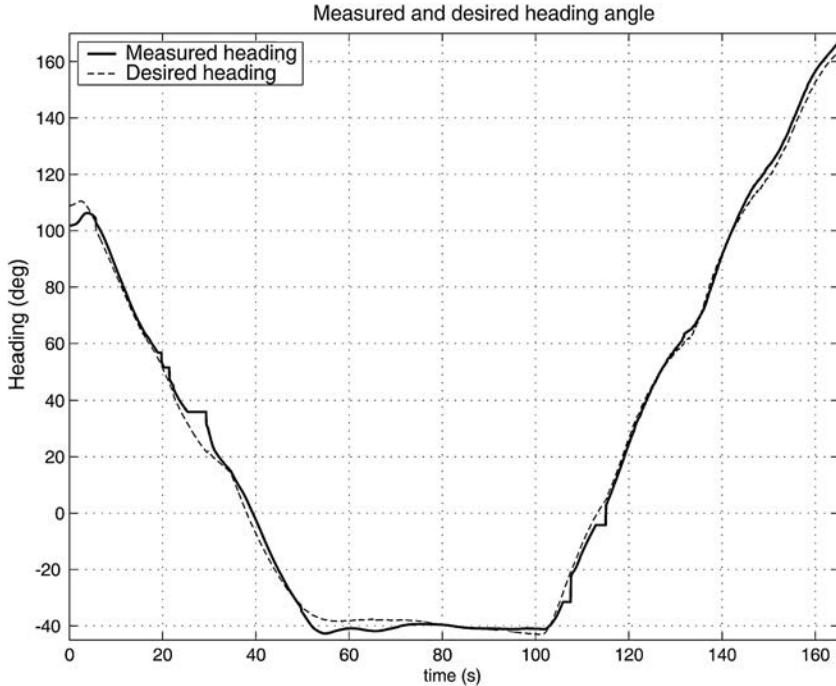


Figure 13.34 The actual yaw angle of the ship tracks the desired LOS angle well.

13.4 Sliding-Mode Control

A robust nonlinear design technique for marine craft is *sliding-mode control* (Utkin, 1977), which incorporates techniques to handle model uncertainty. Sliding-mode techniques are discussed in detail by Utkin (1992) while applications to marine craft are found in Yoerger and Slotine (1985), Slotine and Li (1991), Healey and Lienard (1993) and McGookin *et al.* (2000a, 2000b), for instance.

13.4.1 SISO Sliding-Mode Control

Define a scalar measure of tracking:

$$s := \tilde{\psi} + 2\lambda\tilde{\psi} + \lambda^2 \int_0^t \tilde{\psi}(\tau) d\tau \quad (13.568)$$

where $\tilde{\psi} = \psi - \psi_d$ is the yaw angle tracking error and $\lambda > 0$ is a design parameter reflecting the bandwidth of the controller. For $s = 0$ this expression describes a sliding surface (manifold) with exponentially stable dynamics. To see this let us define a second sliding surface:

$$s_0 := \tilde{\psi} + \lambda \int_0^t \tilde{\psi}(\tau) d\tau \quad (13.569)$$

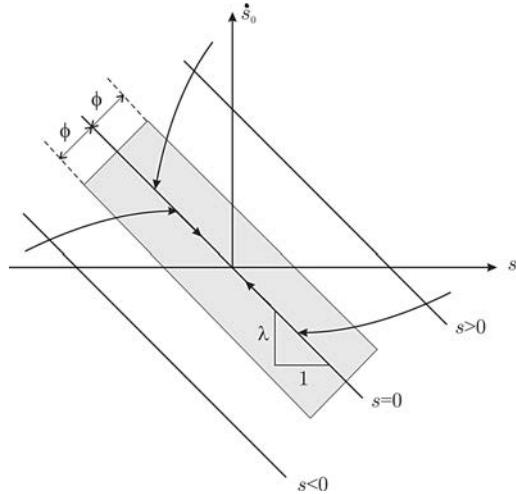


Figure 13.35 Graphical interpretation of the sliding surface $s = \dot{s}_0 + \lambda s_0$ and boundary layer $\phi > 0$.

such that the manifold $s = 0$ can be rewritten as

$$s = \dot{s}_0 + \lambda s_0 = 0 \quad (13.570)$$

Hence, both s_0 and $\tilde{\psi}$ converge exponentially to zero since the linear system

$$\begin{bmatrix} \dot{\tilde{\psi}} \\ \dot{s}_0 \end{bmatrix} = \begin{bmatrix} -\lambda & 1 \\ 0 & -\lambda \end{bmatrix} \begin{bmatrix} \tilde{\psi} \\ s_0 \end{bmatrix} \quad (13.571)$$

has two real eigenvalues at $-\lambda$. This ensures that the tracking error $\tilde{\psi} \rightarrow 0$ on the manifold $s = 0$. Hence, the *control objective* is reduced to finding a nonlinear control law which ensures that

$$\lim_{t \rightarrow \infty} s = 0 \quad (13.572)$$

A graphical interpretation of the *sliding surfaces* is given in Figure 13.35.

It is seen that a trajectory starting at $s > 0$ will move toward the sliding surface $s = 0$. When $s = 0$ is reached the trajectory will continue moving on the straight line corresponding to $s = 0$ toward the equilibrium point $s_0 = 0$. Similar behavior is observed when starting with a negative value of s .

Nonlinear Ship Autopilot

When deriving the control law, a stable ship model with nonlinear damping is considered:

$$T\dot{r} + n_3 r^3 + n_1 r = K\delta + \tau_{\text{wind}} \quad (13.573)$$

where τ_{wind} is the wind moment. Define a new signal v according to

$$v := r - s \implies s = r - v \quad (13.574)$$

such that

$$\begin{aligned} T\dot{s} &= T\dot{r} - T\dot{v} \\ &= K\delta + \tau_{\text{wind}} - (n_3 r^2 + n_1)r - T\dot{v} \\ &= K\delta + \tau_{\text{wind}} - (n_3 r^2 + n_1)(v + s) - T\dot{v} \end{aligned} \quad (13.575)$$

Consider the CLF:

$$V(s) = \frac{1}{2}Ts^2, \quad T > 0 \quad (13.576)$$

Differentiating V along the trajectories of s yields

$$\begin{aligned} \dot{V}(s) &= sT\dot{s} \\ &= s[K\delta + \tau_{\text{wind}} - (n_3 r^2 + n_1)(v + s) - T\dot{v}] \\ &= -[n_3 r^2 + n_1]s^2 + s[K\delta + \tau_{\text{wind}} - (n_3 r^2 + n_1)v - T\dot{v}] \end{aligned} \quad (13.577)$$

Let the control law be chosen as

$$\delta = \frac{\hat{T}}{\hat{K}}\dot{v} + \frac{1}{\hat{K}}[\hat{n}_3 r^2 + n_1]v - \frac{1}{\hat{K}}\tau_{\text{wind}} - K_d s - K_s \text{sgn}(s) \quad (13.578)$$

where $K_d > 0$ and $K_s > 0$, while \hat{T} , \hat{K} and \hat{n}_3 are estimates of T , K and n_3 , respectively. Notice that $n_1 = 1$ for a stable ship. The signum function is defined as

$$\text{sgn}(s) := \begin{cases} 1 & \text{if } s > 0 \\ 0 & \text{if } s = 0 \\ -1 & \text{otherwise} \end{cases} \quad (13.579)$$

This implies that

$$\begin{aligned} \dot{V}(s) &= -[n_3 r^2 + n_1 + K_d]s^2 - K_s |s| \\ &\quad + \left[\left(\frac{\hat{T}}{\hat{K}} - \frac{T}{K} \right) \dot{v} + \left(\frac{1}{\hat{K}} - \frac{1}{K} \right) [n_1 v - \tau_{\text{wind}}] \right. \\ &\quad \left. + \left(\frac{\hat{n}_3}{\hat{K}} - \frac{n_3}{K} \right) r^2 v \right] s \end{aligned} \quad (13.580)$$

In order for this expression to become negative, K_s must be chosen large enough so that the parameter errors are dominated. Consequently,

$$\begin{aligned} K_s &\geq \left| \left(\frac{\hat{T}}{\hat{K}} - \frac{T}{K} \right) \dot{v} \right| + \left| \left(\frac{1}{\hat{K}} - \frac{1}{K} \right) [n_1 v - \tau_{\text{wind}}] \right| \\ &\quad + \left| \left(\frac{\hat{n}_3}{\hat{K}} - \frac{n_3}{K} \right) r^2 v \right| \end{aligned} \quad (13.581)$$

implies that

$$\begin{aligned}\dot{V}(s) &\leq -(n_3 r^2 + n_1 + K_d)s^2 - K_s |s| \\ &< 0, \quad \forall s \neq 0\end{aligned}\quad (13.582)$$

The nonpositive term $-(n_3 r^2 + n_1 + K_d)s^2$ ensures exponential stability of $s = 0$. However, in order to converge to the manifold $s \rightarrow 0$ in finite time such that $\tilde{\psi} \rightarrow 0$ in finite time, the gain $K_s > 0$ must be positive.

One way to find an estimate of K_s is to assume, for instance, 20 % uncertainty in all elements such that

$$K_s \geq 1.2 \frac{\hat{T}}{\hat{K}} |\dot{v}| + 1.2 \frac{1}{\hat{K}} |n_1 v - \tau_{\text{wind}}| + 1.2 \frac{\hat{n}_3}{\hat{K}} |r^2 v| \quad (13.583)$$

It is well known that the switching term $K_s \text{sgn}(s)$ can lead to chattering for large values of K_s . Hence, $K_s > 0$ should be treated as a design parameter with (13.581) as a guideline. Recall that Lyapunov stability analysis results in conservative requirements for all gains.

Chattering in the controller can, however, be eliminated by replacing the signum function with a saturating function. Slotine and Li (1991) suggest smoothing out the control discontinuity inside a boundary layer according to

$$\text{sat}(s) = \begin{cases} \text{sgn}(s) & \text{if } |s/\phi| > 1 \\ s/\phi & \text{otherwise} \end{cases} \quad (13.584)$$

where $\phi > 0$ can be interpreted as the boundary layer thickness. This substitution will assign a low-pass filter structure to the dynamics inside the boundary layer (see Figure 13.35). Another possibility is to replace $K_s \text{sgn}(s)$ with $K_s \tanh(s/\phi)$, where $\phi > 0$ is a design parameter used to shape the slope of $\tanh(\cdot)$ close to the origin.

13.4.2 Sliding-Mode Control using the Eigenvalue Decomposition

Healey and Lienard (1993) have applied the theory of *sliding-mode control* to control the NPS AUV II. A related work discussing the problems of adaptive sliding-mode control in the dive plane is found in Cristi *et al.* (1990). Sliding-mode control for highly maneuverable underwater vehicles is discussed by Lyshevski (2001), who considers the 6 DOF underwater vehicle equations of motion.

The method presented in this section can be applied to the lateral model (7.132), which includes the roll mode, or to the reduced order model (7.33), which is recognized as the linearized maneuvering model (see Section 7.1.4). Consider the state-space model

$$\dot{x} = Ax + bu + f(x, t) \quad (13.585)$$

where $f(x, t)$ is a nonlinear function describing the deviation from linearity in terms of disturbances and unmodeled dynamics, $x = [v, r, \psi]^\top$ and $u = \delta_R$ is the rudder angle. Consequently,

$$A = \begin{bmatrix} a_{11} & a_{12} & 0 \\ a_{21} & a_{22} & 0 \\ 0 & 1 & 0 \end{bmatrix}, \quad b = \begin{bmatrix} b_1 \\ b_2 \\ 0 \end{bmatrix} \quad (13.586)$$

The experiments of Healey and coauthors show that this model can be used to describe a large number of AUV maneuvers. The feedback control law is composed of two parts:

$$u = -\mathbf{k}^\top \mathbf{x} + u_o \quad (13.587)$$

where $\mathbf{k} \in \mathbb{R}^3$ is the feedback gain vector. Substituting (13.587) into (13.585) yields the closed-loop dynamics

$$\dot{\mathbf{x}} = \underbrace{(\mathbf{A} - \mathbf{b}\mathbf{k}^\top)}_{\mathbf{A}_c} \mathbf{x} + \mathbf{b}u_o + \mathbf{f}(\mathbf{x}, t) \quad (13.588)$$

where \mathbf{k} is computed by means of pole placement. In order to determine the nonlinear part u_o of the feedback control law, consider the output mapping

$$s = \mathbf{h}^\top \tilde{\mathbf{x}} \quad (13.589)$$

where $\mathbf{h} \in \mathbb{R}^3$ is a design vector to be chosen such that $s \rightarrow 0$, implying convergence of the state tracking error $\tilde{\mathbf{x}} = \mathbf{x} - \mathbf{x}_d \rightarrow \mathbf{0}$. The output mapping s is also referred to as a *sliding surface*. Premultiplication of (13.588) with \mathbf{h}^\top and then subtraction of $\mathbf{h}^\top \dot{\mathbf{x}}_d$ from both sides gives

$$\dot{s} = \mathbf{h}^\top \mathbf{A}_c \mathbf{x} + \mathbf{h}^\top \mathbf{b} u_o + \mathbf{h}^\top \mathbf{f}(\mathbf{x}, t) - \mathbf{h}^\top \dot{\mathbf{x}}_d \quad (13.590)$$

Assume that $\mathbf{h}^\top \mathbf{b} \neq 0$ and let the nonlinear control law be chosen as (see Figure 13.36)

$$u_o = (\mathbf{h}^\top \mathbf{b})^{-1} [\mathbf{h}^\top \dot{\mathbf{x}}_d - \mathbf{h}^\top \hat{\mathbf{f}}(\mathbf{x}, t) - \eta \text{sgn}(s)], \quad \eta > 0 \quad (13.591)$$

where $\hat{\mathbf{f}}(\mathbf{x}, t)$ is an estimate of $\mathbf{f}(\mathbf{x}, t)$. This gives the s dynamics

$$\dot{s} = \mathbf{h}^\top \mathbf{A}_c \mathbf{x} - \eta \text{sgn}(s) + \mathbf{h}^\top \Delta \mathbf{f}(\mathbf{x}, t) \quad (13.592)$$

where $\Delta \mathbf{f}(\mathbf{x}, t) = \mathbf{f}(\mathbf{x}, t) - \hat{\mathbf{f}}(\mathbf{x}, t)$. The first term in this equation can be rewritten as

$$\mathbf{h}^\top \mathbf{A}_c \mathbf{x} = \mathbf{x}^\top \mathbf{A}_c^\top \mathbf{h} = \lambda \mathbf{x}^\top \mathbf{h} \quad (13.593)$$

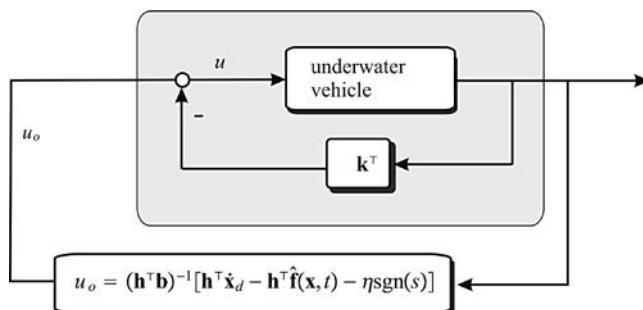


Figure 13.36 Nonlinear sliding-mode controller.

by requiring that \mathbf{h} is a *right eigenvector* of \mathbf{A}_c^\top such that

$$\mathbf{A}_c^\top \mathbf{h} = \lambda \mathbf{h} \quad (13.594)$$

where $\lambda = \lambda(\mathbf{A}_c^\top)$ is the *eigenvalue* corresponding to \mathbf{h} . Hence,

$$\dot{s} = \lambda \mathbf{x}^\top \mathbf{h} - \eta \text{sgn}(s) + \mathbf{h}^\top \Delta f(\mathbf{x}, t) \quad (13.595)$$

Computation of \mathbf{h} and \mathbf{k}

The eigenvalue λ in (13.595) can be made zero by noticing that (13.586) has one pure integrator. Let

$$\mathbf{k} = [k_1, k_2, 0]^\top \quad (13.596)$$

such that the linear part of the controller only stabilizes the sway velocity v and yaw rate r . The yaw angle ψ is left uncontrolled in the inner loop since this results in a closed-loop system matrix:

$$\mathbf{A}_c = \begin{bmatrix} a_{11} - b_1 k_1 & a_{12} - b_1 k_2 & 0 \\ a_{21} - b_2 k_1 & a_{22} - b_2 k_2 & 0 \\ 0 & 1 & 0 \end{bmatrix} \quad (13.597)$$

where one of the eigenvalues is zero. Consequently,

$$\lambda \mathbf{x}^\top \mathbf{h} = 0 \text{ if } \mathbf{h} \text{ is a right eigenvector of } \mathbf{A}_c^\top \text{ for } \lambda = 0 \quad (13.598)$$

With this choice of \mathbf{h} , the s dynamics (13.595) reduces to

$$\dot{s} = -\eta \text{sgn}(s) + \mathbf{h}^\top \Delta f(\mathbf{x}, t) \quad (13.599)$$

and it follows from

$$V = \frac{1}{2} s^2 \quad (13.600)$$

that

$$\begin{aligned} \dot{V} &= s \dot{s} \\ &= -\eta \text{sgn}(s)s + s \mathbf{h}^\top \Delta f(\mathbf{x}, t) \\ &= -\eta |s| + s \mathbf{h}^\top \Delta f(\mathbf{x}, t) \end{aligned} \quad (13.601)$$

Selecting η as

$$\eta > \| \mathbf{h} \| \cdot \| \Delta f(\mathbf{x}, t) \| \quad (13.602)$$

finally yields

$$\dot{V} \leq 0 \quad (13.603)$$

Hence, by application of Barbălat's lemma, s converges to zero in finite time if η is chosen to be large enough to overcome the destabilizing effects of the unmodeled dynamics $\Delta f(\mathbf{x}, t)$. The magnitude of η will be a trade-off between robustness and performance.

Implementation Considerations

In practical implementations, chattering should be removed by replacing $\text{sgn}(s)$ with

$$\text{sat}(s) := \begin{cases} \text{sgn}(s) & \text{if } |s/\phi| > 1 \\ s/\phi & \text{otherwise} \end{cases} \quad (13.604)$$

where the design parameter ϕ is the sliding surface boundary layer thickness. Alternatively, the discontinuous function $\text{sat}(s/\phi)$ could be replaced by the continuous function $\tanh(s/\phi)$. It should be noted that the proposed feedback control with a predescribed η usually yields a conservative estimate of the necessary control action required to stabilize the plant. This suggests that η should be treated as a tunable parameter.

13.4.3 Case Study: Heading Autopilot for Ships and Underwater Vehicles

Consider the autopilot model

$$\begin{bmatrix} \dot{v} \\ \dot{r} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & 0 \\ a_{21} & a_{22} & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} v - v_c \\ r \\ \psi \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \\ 0 \end{bmatrix} \delta \quad (13.605)$$

where $|v_c| < v_c^{\max}$ is the transverse ocean current velocity. The reference trajectory is specified according to

$$\begin{bmatrix} \dot{\psi}_d \\ \dot{r}_d \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\omega_n^2 & -2\zeta\omega_n \end{bmatrix} \begin{bmatrix} \psi_d \\ r_d \end{bmatrix} + \begin{bmatrix} 0 \\ \omega_n^2 \end{bmatrix} \psi_{\text{ref}} \quad (13.606)$$

while $v_d = 0$ during turning. Let $\mathbf{x} = [v, r, \psi]^T$ and $\mathbf{h} = [k_1, k_2, k_3]^T$ such that

$$s = \mathbf{h}^T(\mathbf{x} - \mathbf{x}_d) = h_1 v + h_2(r - r_d) + h_3(\psi - \psi_d) \quad (13.607)$$

Feedback from the sway velocity v and yaw rate r , that is $\mathbf{k} = [k_1, k_2, 0]^T$, implies that

$$\mathbf{A}_c = \mathbf{A} - \mathbf{b}\mathbf{k}^T = \begin{bmatrix} a_{11} - b_1 k_1 & a_{12} - b_1 k_2 & 0 \\ a_{21} - b_2 k_1 & a_{22} - b_2 k_2 & 0 \\ 0 & 1 & 0 \end{bmatrix} \quad (13.608)$$

where the yaw dynamics $\dot{\psi} = r$ is left unchanged. A pure integrator in yaw corresponding to the eigenvalue $\lambda = 0$ is necessary in order to satisfy

$$\lambda \mathbf{x}^T \mathbf{h} = 0 \quad (13.609)$$

The eigenvector \mathbf{h} is computed in Matlab as:

Matlab

```

p = [-1 -1 0]           % desired poles for Ac
k = place(A,b,p)        % pole placement
Ac = A-b*k'
[V,D]=eig(Ac')         % eigenvalue decomposition
for i = 1:3              % extract the eigenvector h from V
    hi = V(:,i);
    if norm(hi.*Ac') < 1e-10; h = hi; end
end

```

The resulting tracking controller is

$$\delta = -k_1 v - k_2 r + \frac{1}{h_1 b_1 + h_2 b_2} [h_2 \dot{r}_d + h_3 r_d - \eta \text{sat}(s)] \quad (13.610)$$

Since the disturbance v_c is unknown, the best guess for

$$f(\mathbf{x}, t) = - \begin{bmatrix} a_{11} \\ a_{21} \\ 0 \end{bmatrix} v_c(t) \quad (13.611)$$

is $f(\hat{\mathbf{x}}, t) = \mathbf{0}$. Hence,

$$\eta > \| \mathbf{h} \| \cdot \| -[a_{11}, a_{12}, 0]^\top v_c^{\max} \| \quad (13.612)$$

13.4.4 Case Study: Pitch and Depth Autopilot for Underwater Vehicles

Pitch and depth control of underwater vehicles is usually done by using control surfaces, thrusters and ballast systems. For a neutrally buoyant vehicle, stern rudders are effective for diving and depth changing maneuvers, since they require relatively little control energy compared to thrusters. Consider the longitudinal model in Section 7.5.6, which can be written

$$\begin{aligned}
& \begin{bmatrix} m - X_{\dot{u}} & -X_{\dot{w}} & mz_g - X_{\dot{q}} \\ -X_{\dot{w}} & m - Z_{\dot{w}} & -mx_g - Z_{\dot{q}} \\ mz_g - X_{\dot{q}} & -mx_g - Z_{\dot{q}} & I_y - M_q \end{bmatrix} \begin{bmatrix} \dot{u} \\ \dot{w} \\ \dot{q} \end{bmatrix} + \begin{bmatrix} -X_u & -X_w & -X_q \\ -Z_u & -Z_w & -Z_q \\ -M_u & -M_w & -M_q \end{bmatrix} \begin{bmatrix} u \\ w \\ q \end{bmatrix} \\
& + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -(m - X_{\dot{u}})u \\ 0 & (Z_{\dot{w}} - X_{\dot{u}})u & mx_g u \end{bmatrix} \begin{bmatrix} u \\ w \\ q \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ W \overline{B} \overline{G}_z \sin(\theta) \end{bmatrix} = \begin{bmatrix} \tau_1 \\ \tau_3 \\ \tau_5 \end{bmatrix} \quad (13.613)
\end{aligned}$$

The speed dynamics can be removed from this model by assuming that the speed controller stabilizes the forward speed such that

$$u = u_o = \text{constant} \quad (13.614)$$

Hence, (13.613) reduces to a combined pitch and diving model

$$\begin{bmatrix} m - Z_w & -mx_g - Z_{\dot{q}} \\ -mx_g - Z_{\dot{q}} & I_y - M_{\dot{q}} \end{bmatrix} \begin{bmatrix} \dot{w} \\ \dot{q} \end{bmatrix} + \begin{bmatrix} -Z_w & -Z_q \\ -M_w & -M_q \end{bmatrix} \begin{bmatrix} w \\ q \end{bmatrix} + \begin{bmatrix} 0 & -(m - X_{\dot{u}})u_o \\ (Z_w - X_{\dot{u}})u_o & mx_g u_o \end{bmatrix} \begin{bmatrix} w \\ q \end{bmatrix} + \begin{bmatrix} 0 \\ \overline{BG_z} W \sin(\theta) \end{bmatrix} = \begin{bmatrix} \tau_3 \\ \tau_5 \end{bmatrix}$$

A state-space representation of this model is

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{b}u + \mathbf{f}(\mathbf{x}, t) \quad (13.615)$$



$$\begin{bmatrix} \dot{w} \\ \dot{q} \\ \dot{\theta} \\ \dot{d} \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & 0 & 0 \\ a_{21} & a_{22} & a_{23} & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & -u_0 & 0 \end{bmatrix} \begin{bmatrix} w \\ q \\ \theta \\ d \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \\ 0 \\ 0 \end{bmatrix} \delta_s + \mathbf{f}(\mathbf{x}, t) \quad (13.616)$$

where $\mathbf{f}(\mathbf{x}, t)$ is a nonlinear function describing the deviation from linearity in terms of disturbances and unmodeled dynamics, $\mathbf{x} = [w, q, \theta, d]^\top$ and $u = \delta_s$ is the stern rudder. The kinematic equations are based on the approximations (see Section 2.2.1)

$$\dot{\theta} = p \cos(\phi) - \sin(\phi) \approx q \quad (13.617)$$

$$\dot{d} = -u_0 \sin(\theta) + v \cos(\theta) \sin(\psi) + w \cos(\theta) \cos(\psi) \approx w - u_0 \theta \quad (13.618)$$

for $v = p = 0$ and small values of θ and ϕ .

The sliding surface for pitch and diving control can be constructed as

$$s = \mathbf{h}^\top \tilde{\mathbf{x}} = h_1(w - w_d) + h_2(q - q_d) + h_3(\theta - \theta_d) + h_4(d - d_d) \quad (13.619)$$

where h_i for $i = 1, \dots, 4$ are the components of \mathbf{h} . Let $\mathbf{x}_d = [w_d, q_d, \theta_d, d_d]^\top$ be a desired state vector given by a reference model. From (13.587) and (13.591) it is seen that

$$u = -\mathbf{k}^\top \mathbf{x} + u_o \quad (13.620)$$

$$u_o = (\mathbf{h}^\top \mathbf{b})^{-1} [\mathbf{h}^\top \dot{\mathbf{x}}_d - \mathbf{h}^\top \hat{\mathbf{f}}(\mathbf{x}, t) - \eta \operatorname{sgn}(s)], \quad \eta > 0 \quad (13.621)$$

where $\mathbf{k} = [k_1, k_2, 0, k_4]^\top$. Notice that $k_3 = 0$ since there is one pure integration in the pitch channel. Hence, \mathbf{h} is found by computing the eigenvalues $\lambda(\mathbf{A}_c) = \lambda(\mathbf{A} - \mathbf{b}\mathbf{k}^\top)$ where $\mathbf{A}_c^\top \mathbf{h} = \mathbf{0}$ for $\lambda_3 = 0$. Consequently,

$$\delta_S = -k_1 w - k_2 q - k_4 d + \frac{1}{h_1 b_1 + h_2 b_2} [h_1 \dot{w}_d + h_2 \dot{q}_d + h_3 \theta_d + h_4 d_d - \eta \operatorname{sgn}(s)] \quad (13.622)$$