

# Automatic Flight Control Summary

---

## 1. Introduction and theory recapitulation

When flying an aircraft, you need to control it. This controlling can be done by humans. But often, it is much easier, safer, faster, more efficient and more reliable to do so by computer. But, how do we automatically control a flight? That is what this summary is all about.

### 1.1 Basic flight control concepts

#### 1.1.1 Introduction automatic flight control

The system that is used to control the flight is called the **flight control system** (FCS). In the early days of flying, the FCS was mechanical. By means of cables and pulleys, the control surfaces of the aircraft were given the necessary deflections to control the aircraft. However, new technologies brought with it the fly-by-wire FCS. In this system electrical signals are sent to the control surfaces. The signals are sent by the **flight (control) computer** (FC/FCC). In this way, the aircraft is controlled.

But what is the advantage of automatic flight control? Why would we use an FC instead of a pilot? There are several reasons for this. First of all, a computer has a much higher reaction velocity than a pilot. Also, it isn't subject to concentration losses and fatigue. Finally, a computer can more accurately know the state the aircraft is in. (Computers can handle huge amounts of data better and also don't need to read a small indicator to know, for example, the velocity or the height of the aircraft.) However, there also is a downside to FCs. They are only designed for a certain flight envelope. When the aircraft is outside of the flight envelope, the system can't really operate the aircraft anymore. For these situations, we still need pilots.

#### 1.1.2 Set-up of the flight control system

The FCS of an aircraft generally consists of three important parts.

- The **stability augmentation system** (SAS) augments to the stability of the aircraft. It mostly does this by using the control surfaces to make the aircraft more stable. A good example of a part of the SAS is the **phugoid damper** (or similarly, the **yaw damper**). A phugoid damper uses the elevator to reduce the effects of the phugoid: it damps it. The SAS is always on when the aircraft is flying. Without it, the aircraft is less stable or possibly even unstable.
- The **control augmentation system** (CAS) is a helpful tool for the pilot to control the aircraft. For example, the pilot can tell the CAS to 'keep the current heading'. The CAS then follows this command. In this way, the pilot doesn't continuously have to compensate for heading changes himself.
- Finally, the **automatic control** system takes things one step further. It automatically controls the aircraft. It does this by calculating (for example) the roll angles of the aircraft that are required to stay on a given flight path. It then makes sure that these roll angles are achieved. In this way, the airplane is controlled automatically.

There are important differences between the above three systems. First of all, the SAS is always on, while the other two systems are only on when the pilot needs them. Second, there is the matter of

**reversibility.** In the CAS and automatic control, the pilot feels the actions that are performed by the computer. In other words, when the computer decides to move a control panel, also the stick/pedals of the pilot move along. This makes these systems reversible. The SAS, on the other hand, is not reversible: the pilot doesn't receive feedback. The reason for this is simple. If the pilot would receive feedback, the only things he would feel are annoying vibrations. This is of course undesirable.

## 1.2 Flight dynamics recap

### 1.2.1 Reference frames

To be able to express the state of the aircraft, we need a reference frame. Several reference frames are around. We'll discuss the four most important ones here. (You've probably seen them before quite some times, but for completeness, we do mention them.)

- The **Earth-fixed frame of reference**  $F_E$  is a right-handed orthogonal system. The  $Z_E$  axis points to the Earth's center, the  $X_E$  axis points North and the  $Y_E$  axis points East. The origin of the system is initially positioned at the aircraft center of gravity (cog). However, the  $F_E$  reference frame is fixed to Earth. So, when the aircraft cog moves, the origin of the  $F_E$  system stays fixed with respect to the Earth.
- The **body-fixed frame of reference**  $F_B$  also is a right-handed orthogonal system. Its origin lies at the aircraft cog and is fixed to it. (So, if the aircraft moves, the frame of reference moves along.) The  $X_B$  axis is parallel to the aircraft longitudinal axis and points forward. The  $Y_B$  axis is parallel to the lateral axis and points to the right. Finally, the  $Z_B$  axis points downward.
- The **stability frame of reference**  $F_S$  also is a right-handed orthogonal system. Its origin is fixed to the aircraft cog, just like with  $F_B$ . Also, the  $Y_S$  axis coincides with the  $Y_B$  axis. However, this time the  $X_S$  axis is rotated downward by the angle of attack  $\alpha$ . To be more precise, the  $X_S$  axis is parallel to the projection of the velocity vector on the plane of symmetry of the aircraft. The  $Z_S$  axis still points downward, but it is of course also rotated by an angle  $\alpha$ .
- Finally, there is the **aircraft frame of reference**  $F_r$ . Contrary to the other systems, this is a left-handed orthogonal system. Its origin is a certain fixed point on the aircraft (though not the cog). The  $X_r$  axis points to the rear of the aircraft, the  $Y_r$  axis points to the left and the  $Z_r$  axis points upward.

### 1.2.2 The equations of motion

By using the reference frames, we can derive the equations of motion. (There are force equations, moment equations and kinematic relations.) These equations are, however, nonlinear. So to be able to work with them more easily, they are linearized about an equilibrium position of the aircraft. (The derivations for this are not given here, since this summary is not about that subject. For the derivations, see the summary of the third-year Flight Dynamics course.) After we have linearized the equations of motion, we can put them in a matrix form. For the symmetric equations of motion, we get

$$\begin{bmatrix} C_{X_u} - 2\mu_c D_c & C_{X_\alpha} & C_{Z_0} & C_{X_q} \\ C_{Z_u} & C_{Z_\alpha} + (C_{Z_{\dot{\alpha}}} - 2\mu_c) D_c & -C_{X_0} & 2\mu_c + C_{Z_q} \\ 0 & 0 & -D_c & 1 \\ C_{m_u} & C_{m_\alpha} + C_{m_{\dot{\alpha}}} D_c & 0 & C_{m_q} - 2\mu_c K_Y^2 D_c \end{bmatrix} \begin{bmatrix} \hat{u} \\ \alpha \\ \theta \\ \frac{q\hat{c}}{V} \end{bmatrix} = \begin{bmatrix} -C_{X_{\delta_e}} & -C_{X_{\delta_t}} \\ -C_{Z_{\delta_e}} & -C_{Z_{\delta_t}} \\ 0 & 0 \\ -C_{m_{\delta_e}} & -C_{m_{\delta_t}} \end{bmatrix} \begin{bmatrix} \delta_e \\ \delta_t \end{bmatrix}. \quad (1.2.1)$$

Similarly, for the asymmetric equations of motion, we get

$$\begin{bmatrix} C_{Y_\beta} + (C_{Y_\beta} - 2\mu_b)D_b & C_L & C_{Y_p} & C_{Y_r} - 4\mu_b \\ 0 & -\frac{1}{2}D_b & 1 & 0 \\ C_{l_\beta} & 0 & C_{l_p} - 4\mu_b K_X^2 D_b & C_{l_r} + 4\mu_b K_{XZ} D_b \\ C_{n_\beta} + C_{n_\beta} D_b & 0 & C_{n_p} + 4\mu_b K_{XZ} D_b & C_{n_r} - 4\mu_b K_Z^2 D_b \end{bmatrix} \begin{bmatrix} \beta \\ \varphi \\ \frac{pb}{2V} \\ \frac{rb}{2V} \end{bmatrix} = \begin{bmatrix} -C_{Y_{\delta_a}} & -C_{Y_{\delta_r}} \\ 0 & 0 \\ -C_{l_{\delta_a}} & -C_{l_{\delta_r}} \\ -C_{n_{\delta_a}} & -C_{n_{\delta_r}} \end{bmatrix} \begin{bmatrix} \delta_a \\ \delta_r \end{bmatrix}. \quad (1.2.2)$$

To use these equations for computations, we often have to transform them into state space form. To put an equation in state space form, we first have to isolate all terms with one of the **differential operators**  $D_c$  and  $D_b$  on one side of the equation. After this, we apply the definition of these operators, being

$$D_c = \frac{\bar{c}}{V} \frac{d}{dt} \quad \text{and} \quad D_b = \frac{b}{V} \frac{d}{dt}. \quad (1.2.3)$$

If we then also move the term  $\frac{\bar{c}}{V}$  (for the symmetric equations) or the term  $\frac{b}{V}$  (for the asymmetric equations) to the other side of the equation, we have put the equation in its state space form.

## 1.3 Control theory recap: frequency domain and diagrams

### 1.3.1 The frequency domain

Let's suppose we have the state space representation of a system, but we want to examine the system in the **frequency domain**. To do this, we have to put the system in the frequency domain first. To accomplish this, we need to follow several steps. First, we have to rewrite the state space form in the Laplace domain. (Take the Laplace transform of the equation.) We can assume zero initial conditions here. (This simplifies matters a bit.) Second, we eliminate the **state vector**  $X(s)$  and rewrite the system of equations as  $Y(s) = F(s)U(s)$ . (Here,  $Y(s)$  is the **output vector** and  $U(s)$  is the **input vector**.) Thirdly, we substitute the Laplace variable  $s$  by  $j\omega$ , with  $j = \sqrt{-1}$  the complex variable. If everything has gone well, then we should have found

$$F(j\omega) = C(j\omega I - A)^{-1} B + D. \quad (1.3.1)$$

We can now examine the frequency domain of the system. Let's suppose that we give our system a sinusoidal input  $U(s)$ . This input has unit magnitude and frequency  $\omega$ . The result is that the output  $Y(s)$  will start to oscillate as well. However, it doesn't do that in exactly the same way. Instead, the amplitude is multiplied by the **amplitude gain**  $K$ . Next to this, there is also a **phase angle**  $\phi$ . Both parameters follow from the transfer function  $F(j\omega)$  and can be found using

$$K = |F(j\omega)| \quad \text{and} \quad \phi = \arg(F(j\omega)). \quad (1.3.2)$$

So, if  $K > 1$ , the oscillation is amplified. Otherwise, it is reduced in strength. Similarly, if  $\phi > 0$ , the system has phase lead. Otherwise, it has phase lag.

### 1.3.2 Different kinds of diagram

We would like to know how the gain  $K$  and the phase angle  $\phi$  vary with the frequency  $\omega$ . This is displayed in a **Bode diagram**. In fact, a Bode diagram consists of two plots. Both plots have on the horizontal axis the frequency  $\omega$ , on a logarithmic scale. The first plot shows the gain  $K$  in decibel (linearly). To put the gain  $K$  in decibel, you can use the equation

$$K_{dB} = 20 \cdot \log_{10} K \quad \Leftrightarrow \quad K = 10^{\left(\frac{K_{dB}}{20}\right)}. \quad (1.3.3)$$

The second plot shows the phase angle  $\phi$  (also linearly).

Next to the Bode diagram, there is also the **Nyquist diagram** (also known as the **polar plot**). To make it, we make a complex plot of  $F(j\omega)$  with respect to  $\omega$ . In other words, we plot the real part of  $F(j\omega)$  on the  $x$ -axis and its imaginary part on the  $y$ -axis for varying  $\omega$ . The distance from the origin now indicates the gain, while the (CCW) angle with respect to the  $x$ -axis indicates the phase angle.

Finally, there is the **Nichols diagram**. In this diagram, we plot the decibal gain  $K_{dB}$  (vertically) against the phase angle  $\phi$  (horizontally).

### 1.3.3 The Nyquist stability criterion

Let's suppose we have a basic feedback system, with transfer function  $F(s) = G(s)/(1 + G(s)H(s))$ .  $F(s)$  now is the **closed loop transfer function** (CL). Also,  $G(s)$  is the **feed forward transfer function** (FF) and  $G(s)H(s)$  is the **open loop transfer function** (OL). We can make a Nyquist diagram of the open loop transfer function  $G(s)H(s)$ . The **Nyquist stability criterion** now tells us something about the stability of the entire closed-loop transfer function  $F(s)$ .

First, we need to count the number of poles  $k$  of the transfer function  $G(s)H(s)$  with real part bigger than zero. (So, the number of poles in the right half plane.) Second, we need to count the number of counterclockwise encirclements of the point  $-1$  of the Nyquist diagram of  $G(s)H(s)$ . If this number is equal to the number  $k$ , then the closed loop system is stable. Otherwise, it is unstable.

## 1.4 Control theory recap: system properties

### 1.4.1 Controllability, observability, stabilizability and detectability

Let's examine a system in state space form. In other words, the system can be described by  $\dot{\mathbf{x}} = A\mathbf{x} + B\mathbf{u}$  and  $\mathbf{y} = C\mathbf{x} + D\mathbf{u}$ . We can now make several definitions concerning this system.

The system, or equivalently the pair  $(A, B)$ , is said to be **state controllable** if, for any initial state  $\mathbf{x}(0) = \mathbf{x}_0$ , any time  $t_1 > 0$  and any final state  $\mathbf{x}_1$ , there exists an input  $\mathbf{u}(t)$  such that  $\mathbf{x}(t_1) = \mathbf{x}_1$ . Otherwise the system is said to be state **uncontrollable**. To find out whether a system is state controllable, we can examine the controllability matrix  $R$ , defined as

$$R = \begin{bmatrix} B & AB & A^2B & \dots & A^{n-1}B \end{bmatrix}. \quad (1.4.1)$$

The system is controllable if, and only if, the matrix  $R$  is of full rank. In other words, all its rows are linearly independent.

The system, or equivalently the pair  $(A, C)$ , is said to be **state observable** if, for any initial time  $t_1 > 0$ , the initial state  $\mathbf{x}(0) = \mathbf{x}_0$  can be determined from the time history of the input  $\mathbf{u}(t)$  and the output  $\mathbf{y}(t)$  in the interval  $[0, t_1]$ . Otherwise the system is said to be **state unobservable**. To find out whether a system is state observable, we can examine the observability matrix  $W$ , defined as

$$W = \begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{n-1} \end{bmatrix}. \quad (1.4.2)$$

The system is observable if, and only if, the matrix  $W$  is of full rank. In other words, all its columns are linearly independent.

The system, or equivalently the pair  $(A, B)$ , is said to be **state stabilizable** if all unstable modes are state controllable. This is the case if there exists a **feedback matrix**  $F$  which stabilizes the system (thus causing  $A + BF$  to be stable). To test for stabilizability, we can use the **Hautus test**. It says that  $(A, B)$  is stabilizable if and only if the matrix

$$\begin{bmatrix} A - \lambda I & B \end{bmatrix} \quad (1.4.3)$$

has full rank for all unstable eigenvalues  $\lambda$ . (In other words, for all  $\lambda$  with positive real parts, the above matrix has linearly independent rows.)

The system, or equivalently the pair  $(A, C)$ , is said to be **state detectable** if all unstable modes are state observable. This is the case if there exists a matrix  $L$  such that  $A + LC$  is stable (and thus has all its eigenvalues in the left half part of the complex plane). To test for detectability, we can again use the Hautus test. This time, it says that  $(A, C)$  is detectable if and only if the matrix

$$\begin{bmatrix} A - \lambda I \\ C \end{bmatrix} \quad (1.4.4)$$

has full rank for all unstable eigenvalues  $\lambda$ . (In other words, for all  $\lambda$  with positive real parts, the above matrix has linearly independent columns.)

## 1.4.2 Varying the poles and zeroes of the open loop transfer function

Let's examine a basic closed loop system with transfer function  $F(s)$ . We can add a constant gain  $K$  into this system. This turns the transfer function from

$$F(s) = \frac{G(s)}{1 + G(s)H(s)} \quad \text{into} \quad F(s) = \frac{KG(s)}{1 + KG(s)H(s)}. \quad (1.4.5)$$

By varying this gain  $K$ , we will vary the properties of the system. This is displayed by a **root locus plot**. (As you remember from Control Theory, a root locus plot shows how the poles of the closed loop transfer function vary with  $K$ .)

Sometimes, however, we can also wind up in a situation, where we are allowed to choose one pole  $p_{var}$  of the open loop transfer function  $G(s)H(s)$ . This situation is very similar to the case where we can choose the gain  $K$ . To show this, we define

$$G(s)H(s) = \left( \frac{1}{s - p_{var}} \right) Q(s). \quad (1.4.6)$$

This will turn the closed loop transfer function into

$$F(s) = \frac{G(s)}{1 + G(s)H(s)} = \frac{\frac{G(s)}{s+Q(s)}(s - p_{var})}{1 - \frac{p_{var}}{s+Q(s)}} = \frac{G_1(s)}{1 + G_1(s)H_1(s)}. \quad (1.4.7)$$

In this equation, we have defined the **modified transfer functions**  $G_1(s)$  and  $H_1(s)$  as

$$G_1(s) = \frac{G(s)}{s + Q(s)}(s - p_{var}) \quad \text{and} \quad H_1(s) = -\frac{p_{var}}{G(s)(s - p_{var})}. \quad (1.4.8)$$

We can now see something interesting. Previously, the denominator of the closed loop transfer function was  $1 + KG(s)H(s)$ . By varying  $K$ , we varied the poles. However, this time the denominator is  $1 - p_{var}(s + Q(s))^{-1}$ . It is of the same form! So, varying  $p_{var}$  is just like varying a gain  $K$ . And we can again make a root locus plot.

We can do the same with varying zeroes. If we can choose a zero  $z_{var}$  of the open loop transfer function, then we write  $G(s)H(s) = (s - z_{var})P(s)$ . The closed loop transfer function now turns into

$$F(s) = \frac{\frac{G(s)}{1+sP(s)}}{1 - \frac{z_{var}P(s)}{1+sP(s)}} = \frac{G_2(s)}{1 + G_2(s)H_2(s)}. \quad (1.4.9)$$

Again, we see that varying  $z_{var}$  is like varying the gain  $K$ . We can therefore again use it to influence the properties of the closed loop system.

## 2. Adjusting system properties

Making an aircraft stable is one thing. But giving it a satisfactory behaviour and being able to control it is another story. In this chapter, we're going to look at some parameters which a system can have. After that, we'll examine how we can influence these parameters.

### 2.1 Important system parameters

For every system, we can find several parameters that mention something about the system. Some parameters give us hints about the stability of the system. And other parameters are nice to know for other reasons. We will now examine quite some parameters.

#### 2.1.1 Phase and gain margins

Let's again examine the system with transfer function  $F(s) = G(s)/(1 + G(s)H(s))$ . We'll examine this function in the frequency domain, and thus substitute  $s$  by  $j\omega$ . If the term  $G(j\omega)H(j\omega)$  ever becomes  $-1$ , then the system becomes unstable. We are thus interested in the points where  $|G(j\omega)H(j\omega)| = 1$  and  $\arg(G(j\omega)H(j\omega)) = -180^\circ$ . The frequency at which  $\phi = \arg(G(j\omega)H(j\omega)) = -180^\circ$  is called the **phase crossover frequency**  $\omega_{\phi=-180^\circ}$ . Similarly, the frequency at which  $K = |G(j\omega)H(j\omega)| = 1$  (or  $K_{dB} = 0$ ) is called the **gain crossover frequency**  $\omega_{K=1}$ .

We would like to know how close we are to instability. So, let's suppose that we already have a phase angle of  $\phi = -180^\circ$ . (We thus have a frequency equal to the phase crossover frequency  $\omega_{\phi=-180^\circ}$ .) The **gain margin**  $GM$  is now defined as

$$GM = \frac{1}{|G(j\omega_{\phi=-180^\circ})|} = \frac{1}{K_{\phi=-180^\circ}}. \quad (2.1.1)$$

A gain margin of  $GM < 1$  (or similarly,  $GM_{dB} < 0$ ) indicates instability. As a rule of thumb, we would like to have  $GM_{dB} > 6$  dB.

Similarly, we can suppose we already have a gain of  $K = 1$ . (We thus have a frequency equal to the gain crossover frequency  $\omega_{K=1}$ .) The **phase margin**  $PM$  is now defined as

$$PM = 180^\circ + \arg(G(j\omega_{K=1})) = 180^\circ + \phi_{K=1}. \quad (2.1.2)$$

A phase margin of  $PM < 0^\circ$  indicates instability. As a rule of thumb, we would like to have  $30^\circ < PM < 60^\circ$ .

The phase and gain margins can also be found in the various plots that were discussed. To find them, you first have to find the point where  $\phi = -180^\circ$  (for the gain margin) or  $K = 1$  (for the phase margin). You then have to find the gain/phase angle, and by using the definition for the gain margin/phase margin you can find the corresponding value. In this way, you can also find the **ultimate gain**  $K_{ult}$ , which is defined as the gain  $K$  at the phase crossover frequency. It can be found from the gain margin, using

$$K_{ult} = GM = 10^{\left(\frac{GM_{dB}}{20}\right)}. \quad (2.1.3)$$

The phase and gain margins can, however, be misleading. It may happen that the phase and gain margins appear safe, but there still is a value of  $\omega$  for which  $G(j\omega)H(j\omega)$  comes close to  $-1$ . Therefore, instead of looking at phase and gain margins, it is often wise to simply look at the Nyquist plot of  $G(j\omega)H(j\omega)$  and see if it comes close to  $-1$ . If not, then the system appears to be quite alright.

## 2.1.2 Other frequency domain parameters

There are more parameters that are related to the frequency domain. Most of these parameters can easily be derived from the Bode plot. We'll discuss a couple of them now.

Let's examine a Bode diagram. In this Bode diagram is a frequency region in which the system performs satisfactory. This region is usually a region with a more or less constant gain  $K_0$ . The point(s) where the gain drops below 3 dB less than this constant gain  $K_0$  is called the **cutoff frequency**. The slope of the Bode plot at this point is called the **cutoff rate**. Also, the frequency range in which the system performs satisfactory (being the frequency range between the cutoff frequencies) is called the **bandwidth**  $\omega_b$ .

In a Bode diagram, you can often find a peak at which the gain  $K$  is at a maximum. This phenomenon is called **resonance**. The corresponding maximum value of the gain  $K$  is denoted by the **resonance peak**  $M_p$ . The frequency at which this resonance occurs is called the **resonance frequency**  $\omega_p$ .

The last important parameter for the frequency domain is the delay time. The **delay time**  $t_d(\omega)$  for a given frequency  $\omega$  is given by

$$t_d(\omega) = -\frac{d\phi}{d\omega} = -\frac{d \arg(G(j\omega))}{d\omega}. \quad (2.1.4)$$

## 2.1.3 Time domain parameters

In the time domain, there are also several parameters that are important. Let's suppose that we have a system in which the output  $y(t)$  needs to follow the input  $u(t)$ . Also, suppose that we put a step function of size  $k$  on the input. (Though usually  $k = 1$  is selected.) So, for  $t < 0$  we have  $u(t) = 0$  and for  $t > 0$  we have  $u(t) = k$ .

Of course, in the time domain, time matters. So, let's examine some characteristic times. First, the **delay time**  $t_d$  is defined such that  $y(t_d) = 0.5y_{ss}$ . In other words, at the delay time the system is halfway with adjusting itself to the new input value. We also have the **rise time**  $t_r$ . But before we can define it, we first need to define  $t_{r_{initial}}$  and  $t_{r_{final}}$ . These parameters are defined such that  $y(t_{r_{initial}}) = 0.1$  and  $y(t_{r_{final}}) = 0.9$ . The rise time is now given by  $t_r = t_{r_{final}} - t_{r_{initial}}$ . Thirdly, the **settling time**  $t_s$  is the time it takes for the system to come and stay close to the steady state output. So, for all  $t > t_s$  we must have  $|y(t) - y_{ss}| < 0.02y_{ss}$ . (Of course, the parameter 0.02 can be varied. A value of 0.05 is often used as well.)

Next to these important time parameters, there are also parameters not related to time. For example, there is the **(maximum) overshoot**  $M_p$ . This is the difference between the maximum value of  $y(t)$  and its steady state value  $y_{ss}$ . (So,  $M_p = \max(y(t)) - y_{ss}$ .) And finally, there is the **apparent time constant**  $\tau$ . To grasp its meaning, we have to suppose that the output is given by a function of the form  $y(t) = y_{ss} - Ae^{-\alpha t} \cos(\omega t + \phi)$ . The parameter  $\tau$  is now defined as  $\tau = 1/\alpha$ . In other words, it is the time it takes until the amplitude of the oscillation has reduced to 37% of its value.

## 2.1.4 Error specifications

When designing a system, there usually are requirements. These requirements can also concern the error which the system has. To examine the error, we first simply assume that  $H(s) = 1$ . Thus  $F(s) = G(s)/(1 + G(s))$ . We then rewrite the open loop transfer function  $G(s)$  of the system as

$$G(s) = K \frac{s^a \prod_{i=1}^{i=m-a} (s + z_i)}{s^b \prod_{j=1}^{j=n-b} (s + p_j)} = K_{mod} \frac{\prod_{i=1}^{i=m-a} (\tau_{z,i}s + 1)}{s^l \prod_{j=1}^{j=n-b} (\tau_{p,j}s + 1)}. \quad (2.1.5)$$

In other words, the open loop transfer function has  $m$  zeroes and  $n$  poles. A number  $a$  of these zeroes is equal to zero. Similarly, a number  $b$  of the poles is zero as well. We also have  $l = b - a$ . We will soon

see that this parameter  $l$  is very important. In fact, it denotes the **type** of the system. If  $l = 0$  then we have a **type 0** system, if  $l = 1$  then we have a **type 1** system, and so on.

The system output  $Y(s)$  should follow the system input  $U(s)$ . So, we define the error  $E(s)$  as the difference. It is thus equal to

$$E(s) = U(s) - Y(s) = U(s) - U(s)F(s) = U(s) - U(s)\frac{G(s)}{1 + G(s)} = \frac{U(s)}{1 + G(s)}. \quad (2.1.6)$$

To find the eventual error  $e(\infty)$  of the system, we can use the **final value theorem**. It implies that

$$e(\infty) = \lim_{t \rightarrow \infty} e(t) = \lim_{s \rightarrow 0} sE(s) = \lim_{s \rightarrow 0} \frac{sU(s)}{1 + G(s)} = \frac{sU(s)}{1 + \frac{K_{mod}}{s}}. \quad (2.1.7)$$

Now we can put various inputs into this system and find the error. This gives us the following results.

- First, we insert a **step input**. Thus,  $u(t) = 1$  (for  $t > 0$ ) and  $U(s) = 1/s$ . We now find that for type 0 systems, there is a **steady state error** of  $e(\infty) = 1/(1 + K_{mod})$ . However, for type 1 and beyond, the error is zero. (By the way, this error is called a **position error**.)
- Second, we insert a **ramp input**. So,  $u(t) = t$  (for  $t > 0$ ) and  $U(s) = 1/s^2$ . This time type 0 systems give an infinite error: it diverges. Type 1 systems give a steady state error of  $e(\infty) = \frac{1}{K_{mod}}$ . Type 2 systems and beyond give a zero error. (This error is called a **velocity error**.)
- Third, we insert a **parabolic input**. So,  $u(t) = \frac{1}{2}t^2$  (for  $t > 0$ ) and  $U(s) = 1/s^3$ . This time type 0 and type 1 systems give an infinite error. Type 2 systems give a steady state error of  $e(\infty) = \frac{1}{K_{mod}}$ . Type 3 systems and beyond give a zero error. (This error is called a **acceleration error**.)

I think you can understand the general trend of the above experiments now. So remember, the type of the system determines which kind of position, velocity and acceleration errors the system has.

## 2.2 Controllers - time domain

By varying the (proportional) open-loop gain  $K$  of the system, we can already vary its properties by quite a bit. But, sometimes varying this gain is not enough. In that case, we need a compensator or a controller. First, we'll examine controllers.

### 2.2.1 PID Control

Let's examine a basic feedback loop with  $H(s) = 1$ . In this feedback loop, the output signal  $Y(s)$  is fed back to the system. Usually, the signal that is fed back is proportional to the output. We thus have a **proportional controller**:  $K(s) = K_p$ . (Here,  $K_p$  is the **proportional gain**.  $K(s)$  is the **controller function**.) A proportional controller generally reduces the rise time  $t_r$ , increases the overshoot  $M_p$  and reduces the steady state error  $e_{ss}$ .

Sometimes, however, it may be convenient to get the derivative of the output as feedback signal. In this case, we use a **derivative controller**:  $K(s) = K_D s$ . ( $K_D$  is the **derivative gain**.) A derivative controller reduces the overshoot  $M_p$  and the settling time  $t_s$ .

Finally, we can also use an **integral controller**:  $K(s) = \frac{1}{s} K_I$ . ( $K_I$  is the **integral gain**.) An integral controller reduces the rise time  $t_r$  and sets the steady state error  $e_{ss}$  to zero. However, it increases the overshoot  $M_p$  and the settling time  $t_s$ .

Of course, we can also combine all these controllers. This gives us the **PID controller**:

$$K(s) = K_p + \frac{K_I}{s} + K_D s = \frac{K_D s^2 + K_p s + K_I}{s}. \quad (2.2.1)$$

By using the PID controller, we can influence the parameters  $t_r$ ,  $t_s$ ,  $M_p$  and  $e_{ss}$  in many ways. Just vary the gains  $K_p$ ,  $K_D$  and  $K_I$ . But which gains do we choose? For that, we can use tuning rules.

## 2.2.2 The Ziegler-Nichols tuning rules

We will now examine the **Ziegler-Nichols tuning rules**. There are two variants: the **quarter decay ratio** method and the **ultimate sensitivity** method. For both methods, we first write  $K(s)$  as

$$K(s) = K_p \left( 1 + \frac{1}{T_I s} + T_D s \right). \quad (2.2.2)$$

Now let's examine the quarter decay ratio method. These tuning rules should give a decay ratio of 0.25. (The **decay ratio** is the ratio of the magnitudes of two consecutive peaks of an oscillation.) First, we examine the response of the original system to a unit step input. From this we determine the **lag**  $L$ , which is the time until the system really starts moving. (We have  $L \approx t_d$ .) We also find the **slope**  $R$ , which is the average slope of the system response during its rise time. (We have  $R \approx y_{ss}/t_r$ .)

Based on the values of  $L$  and  $R$ , we can choose our gains. If we only use proportional gain, then  $K_p = \frac{1}{RL}$ . If we use a PI controller, then  $K_p = \frac{0.9}{RL}$  and  $T_I = \frac{L}{0.3}$ . Finally, if we use a PID controller, then  $K_p = \frac{1.2}{RL}$ ,  $T_I = 2L$  and  $T_D = 0.5L$ . These rules should then roughly give a decay ratio of 0.25. Although some additional tuning is often necessary/recommended.

Now let's examine the ultimate sensitivity method. First, we examine the original system with a gain equal to the ultimate gain  $K_p = K_{ult}$ . In other words, we choose  $K_p$  such that the system has continuous oscillations without any damping. The corresponding **ultimate period** of these oscillations is now denoted by  $P_u$ . (This does mean that the ultimate sensitivity method can only be used when continuous oscillations can be achieved. In other words, the root locus plot has to cross the imaginary axis at a point other than zero.)

Based on the values of  $K_{ult}$  and  $P_u$ , we can choose our gains. For proportional control, we use  $K_p = 0.5K_{ult}$ . For PI control, we use  $K_p = 0.45K_{ult}$  and  $T_I = \frac{P_u}{1.2}$ . For PID control, we use  $K_p = 0.6K_{ult}$ ,  $T_I = \frac{1}{2}P_u$  and  $T_D = \frac{1}{8}P_u$ . Again, additional tuning is often necessary/recommended.

## 2.3 Compensators - frequency domain

### 2.3.1 Three kinds of compensators

There are three important kinds of compensators. These are the lead compensator, the lag compensator and the lead-lag compensator, respectively given by the transfer functions

$$D_1(s) = K(s + z), \quad D_2(s) = \frac{K}{s + p} \quad \text{and} \quad D_3(s) = K \frac{s + z}{s + p}. \quad (2.3.1)$$

Let's look at these compensators individually.

The **lead compensator** offers PD control. This causes it to speed up the response of a system. In other words, the rise time  $t_r$  goes down. Also, the overshoot  $M_p$  becomes less. The lead compensator does have a problem though. It increases the gain of the system at high frequencies. In other words, with a lead compensator high frequencies are amplified. This is generally not very positive.

The **lag compensator** offers PI control. This means that it improves the steady state accuracy. (If you need to have  $e_{ss} \approx 0$ , then a lag compensator comes in handy.) The PI controller reduces high-frequency noise. As such, it can be used as a **low-pass filter**. (This is a filter that only lets low frequencies pass.)

The **lead-lag compensator** combines the lead and the lag compensator. In this way, the negative effects of the lead compensator can be compensated for. First, a lead compensator can be used to speed up the response of the system. Then a lag compensator is also added, such that the high frequency effects are limited. This lag compensator is made such that its effects on the biggest part of the system are negligible.

In the lead-lag compensator, the lead compensator is the most important part. However, we can also put it together such that the lag compensator is the most important part. In this case, we often call the compensator a **lag-lead compensator**.

### 2.3.2 Tuning the compensators

Using lead and lag compensators is like adding zeros and poles to the system. But when doing this, an important question arises: where do we put the zeros and poles? For this, we can use the root locus plot. We now have a nice rule of thumb: poles push the locus away, whereas zeros attract the locus. But we also have more precise rules to place the zeros and poles.

Let's suppose we're setting up a lead compensator. We thus need to choose its zero. It is often wise to put this zero in the neighbourhood of the **natural frequency**  $\omega_N$  which you want the system to have. This natural frequency can roughly be determined from the parameters  $t_r$  and/or  $t_s$  using the approximate equations

$$t_r \approx \frac{1.8}{\omega_N} \quad \text{and} \quad t_s \approx \frac{4.6}{\zeta \omega_N}. \quad (2.3.2)$$

In this equation, the value of  $\zeta$  can often be determined from the required value of the overshoot peak  $M_p$ , according to

$$\zeta \approx 0.7 \text{ when } M_p \approx 5\%, \quad \zeta \approx 0.5 \text{ when } M_p \approx 15\% \quad \text{and} \quad \zeta \approx 0.3 \text{ when } M_p \approx 35\%. \quad (2.3.3)$$

To compensate for high frequency effects, we also add a pole (as a lag compensator). This pole, however, should be relatively far away from the zero. A rule of thumb is to place the pole 5 to 20 times further from the origin as the zero. Thus,  $p \approx (5 \text{ to } 20) \cdot z$ .

## 3. System performance specifications

Previously, we have considered how to influence the parameters of a system. We can use it to give aircraft the correct parameters. Now we will take a look at the requirements which an aircraft should have. First, we examine how the requirements are built up. Later on, we'll examine which parameters are subject to these requirements.

### 3.1 Requirements on flying and handling qualities

#### 3.1.1 Flying quality requirements

Most countries have **regulating agencies** (like the **JAR** for Europe and the **FAR** for the US). These agencies specify **flying quality requirements**. These requirements are the minimum acceptable standard of the flying and handling qualities of an aircraft. They define rules subject to which the stability, control and handling of the aircraft must be designed.

The flying quality requirements differ per aircraft class. Small light airplanes are **class I**, medium weight airplanes are **class II**, big heavy airplanes are **class III** and high manoeuvrability airplanes are **class IV**. Next to this, there are also separate criteria per flight phase. **Category A** concerns non-terminal flight phases that require rapid manoeuvring, precision tracking or precise flight path control. (Think of air combat/terrain following.) **Category B** is about non-terminal flight phases that require gradual manoeuvring, less precise tracking and less accurate flight path control. (Think about climb, descent and cruise.) Finally, **category C** relates to terminal flight phases that require gradual manoeuvring and precision flight path control. (This includes take-off and landing.)

Aircraft manufacturers must demonstrate a compliance with the specifications. This is done by using flight tests. In these tests, the flying quality of the aircraft is rated. This is often done based on the **Cooper-Harper scale**. In this scale, a 1 means the aircraft has excellent handling qualities, and the pilot workload is low. On the other hand, a 10 means that there are major deficiencies in the handling quality of the aircraft. A test is never performed for just an aircraft or just a control system. It is always performed for the combination of the aircraft and the control system.

Flight requirements are generally specified for three **levels of flying quality**. **Level 1** means that the flying qualities are clearly adequate for the respective flight phase. **Level 2** means that the flying qualities are still adequate, but there is an increase in pilot workload and/or degradation in mission effectiveness. In **level 3**, the flying qualities are degraded. However, the airplane can still be controlled, albeit with an inadequate mission effectiveness and a high or limiting pilot workload. Airplanes must be designed to satisfy level 1 flying quality requirements with all systems in their normal operating state.

#### 3.1.2 Flying and handling qualities

Flying quality requirements are present to make sure aircraft have flying and handling qualities. But what do these qualities mean? **Flying qualities** concern how well a (long-term) task can be fulfilled. **Handling qualities**, however, concern how the aircraft responds (short-term) to inputs. Important parameters that influence these qualities include the **stability** of the aircraft and the **flight control system** (FCS) characteristics. Let's take a closer look at these two parameters.

With stability, we mean how easy it is to establish an equilibrium flight condition, without the aircraft having a tendency to diverge. There are two kinds of stabilities. With **static stability**, we mean that every deviation causes an opposing force/moment. However, the deviation does not have to be eliminated. It could simply be the case that a new equilibrium position is created. With **dynamic stability** we also mean that every deviation from the equilibrium position is eliminated. In other words, the system returns

to the original equilibrium position. It is usually nice if an aircraft is stable. However, if an aircraft is too stable, then it's not manoeuvrable anymore: it's too hard to get it out of its equilibrium position. So, this isn't a positive thing either.

There are three ways in which the FCS can have a bad effect on the flying qualities. First, something can occur between the cockpit and the actuators. (For example, there may be a lag in the signal that is sent to an actuator.) Second, there can also be a lag in an actuator itself. (For example, when the ailerons takes a long time to deflect.) And finally, the displays in the cockpit can lag. Because this is undesirable, requirements are made concerning these lags. Next to this, also **control system break-out forces** are important. These are the forces which the pilot must apply before his actions have any effect at all.

Giving an airplane the right flying and handling qualities usually isn't easy. It often has a bad effect on the performance and weight of the aircraft. Therefore, trade-offs often need to be made between the flying/handling quality and the performance of the aircraft. That is, if trade-offs can be made. Requirements on the flying and handling qualities are simply present and they have to be followed.

## 3.2 Parameters subject to requirements

### 3.2.1 Longitudinal flight requirements

Now let's look at some actual requirements for aircraft. First, we'll examine requirements on the longitudinal flight. After that, we'll also consider the lateral flight.

In longitudinal flight, there are requirements on the **control forces** which the pilot needs to exert. These forces are often indicated by the **stick force**  $F_s$ . In a manoeuvring flight, the gradient  $\partial F_s / \partial n$  with respect to the **load factor**  $n$  is important. It should fall within limits. (We're not going to mention any numbers here. There are just too many to mention, and you're not going to remember them anyway.) Also, there should be no significant nonlinearities in it. Next to this, when the airplane configuration changes a bit (e.g. the flaps are deployed), the control forces shouldn't change significantly either.

When the aircraft changes its speed, no strange things may happen either. The **stick-force-speed-gradient**  $\partial F_s / \partial V$  must be stable and must meet other specs. Also, the **return-to-trim-speed-behaviour** must meet certain requirements. During take-off and landing (with fixed trim controls) the control force must be within certain limits. Also, during a dive, the control force may not exceed certain values. These values also depend on whether the aircraft is equipped with a stick or a wheel. Also, the allowable values depend on whether it concerns pushing or pulling the wheel/stick.

Next to control forces, also the dynamic behaviour of the aircraft is important. Let's consider the **phugoid requirements** first. The phugoid must have a certain prescribed damping  $\zeta$ . And if the eigenmotion is allowed to be unstable (for example, during a level 3 flying quality situation), then there will be a requirement on the **time to double amplitude**  $T_{2ph}$ . This time can be found, by using

$$A_{ph} e^{-\zeta_{ph} \omega_{n_{ph}} (t_1 + T_{2ph})} = 2 A_{ph} e^{-\zeta_{ph} \omega_{n_{ph}} t_1} \quad \Rightarrow \quad T_{2ph} = \frac{\ln 2}{-\zeta_{ph} \omega_{n_{ph}}} \quad (3.2.1)$$

In the above equation,  $A_{ph}$  is the amplitude of the phugoid motion,  $\zeta_{ph}$  the damping ratio and  $\omega_{n_{ph}}$  the natural frequency.

Similar to the phugoid, there are also **short period motion requirements**. These requirements of course concern the damping ratio  $\zeta_{sp}$ . But for the short period motion also the natural frequency  $\omega_{n_{sp}}$  is quite important. However, for (highly) augmented airplanes, these requirements are not used. Instead, the **control anticipation parameter** (CAP) is used, which is defined as

$$\text{CAP} = \frac{\dot{q}(t=0)}{n_z(t=\infty)} = \frac{\omega_{n_{sp}}^2}{n_\alpha} \quad (3.2.2)$$

In this equation,  $n_\alpha = \partial n / \partial \alpha$  is the gust- or load-factor-sensitivity. By using flight dynamics equations, expressions/approximations can be calculated for  $\omega_{n_{sp}}$  and  $n_\alpha$ , after which the CAP can be found.

Finally, the aircraft must have **flight path stability**. This means that there are requirements on the derivative  $\partial \gamma / \partial V_P$ . The term  $\partial V_P$  here denotes the changes in velocity that are caused by pitch control only. (So, things like engine throttle effects are not taken into account.)

### 3.2.2 Lateral flight requirements

There are several requirements for lateral flight as well. First, we'll look at the **lateral control forces**. These control forces concern both the (sideways) forces on the stick and the forces on the rudder pedals. The force requirements depend on the situation. (For example, there are separate requirements for the situation where one engine isn't functioning anymore.) It also matters whether it concerns a short/temporary force or a prolonged force.

Now let's examine the **Dutch roll**. For this eigenmotion, of course the damping ratio  $\zeta_d$  is very important, as well as the natural frequency  $\omega_{n_d}$ . Therefore, there are certain minimum values for these parameters. There are also requirements on the product  $\zeta_d \omega_{n_d}$ . And, depending on the roll angle  $\phi$  and the sideslip angle  $\beta$ , more complicated requirements can be put on the damping and frequency.

For the **spiral** eigenmotion, divergence is usually allowed. Now the time  $T_{2_s}$  until a double amplitude is reached (with cockpit controls free) is an important parameter. For the **roll mode**, the roll mode time constant  $T_R$  is important. A maximum value is usually specified.

Stability and manoeuvrability also matter. How quickly can an airplane reach a certain roll angle? Or alternatively, in a given time, what maximum roll angle can be achieved? And is the aircraft **directionally stable**? In other words, it is required that  $C_{n_\beta} > 0$ . Next to this, it is also desired/required that  $C_{Y_\beta} < 0$  and  $C_{l_\beta} < 0$ .

### 3.2.3 The Gibson criterion

The Gibson criterion is a special criterion to prevent certain aircraft behaviour. It is mainly relevant when a pilot is trying to change the pitch rate of the aircraft. The Gibson criterion can be split up into the dropback criterion and the phase rate criterion. We'll examine the dropback criterion first.

Let's suppose that we have a certain pitch angle and we want to reach another pitch angle. We deflect the elevator, until we have reached the desired pitch angle. Then, we let go of the control surfaces. What happens next? If we go back to a situation with a smaller pitch angle, then **dropback** (DB) occurs. However, if the pitch angle continues to increase, then **overshoot** (OS) occurs. (Dropback can thus be seen as negative overshoot and vice versa.)

The **dropback criterion** concerns dropback. Important parameters are the **maximum pitch rate**  $q_m$ , the **steady state value** of the pitch rate  $q_s$  and the **pitch rate overshoot ratio**  $q_m/q_s$ . The dropback criterion now describes a region in which the values of  $DB/q_s$  and  $q_m/q_s$  should be. Basically, zero dropback is optimal. However, some dropback is preferred to overshoot. Acceptable pitch rate overshoot values are  $1 \leq q_m/q_s \leq 3$ .

The **phase rate criterion** is present to prevent/reduce **pilot induced oscillations** (PIOs). A PIO can occur when the pilot continuously tries to compensate for something, but by doing so only contributes to oscillations. Important parameters now are the **frequency** at  $180^\circ$  phase lag  $\omega_{\phi=-180^\circ}$  and the **phase rate** at  $180^\circ$  phase lag  $(\partial \phi / \partial \omega)_{\phi=-180^\circ}$ . The phase rate criterion now demands that

$$\omega_{\phi=-180^\circ} \approx 1 \text{ Hz} \quad \text{and} \quad \left( \frac{\partial \phi}{\partial \omega} \right)_{\phi=-180^\circ} \leq 100 \text{ deg/Hz.} \quad (3.2.3)$$

The optimum for the parameters is for  $\omega_{\phi=-180^\circ}$  to be somewhere between 1 and 1.4 and for  $(\partial \phi / \partial \omega)_{\phi=-180^\circ}$  to be somewhere between 60 Hz and 90 Hz. In this case, then the chance that a PIO occurs is really low.

## 4. Stability augmentation systems

Stability augmentation systems make the aircraft more stable. There are SASs for both the dynamic stability (whether the eigenmotions don't diverge) and the static stability (whether the equilibrium position itself is stable). First, we'll look at the dynamic stability: how can we effect the eigenmotion properties? Second, we'll examine the static stability: how do we make sure an aircraft stays in a steady flight?

### 4.1 Dampers – Acquiring dynamic stability

An airplane has several eigenmotions. When the properties of these eigenmotions don't comply with the requirements, we need an SAS. The SAS is mostly used to damp the eigenmotions. Therefore, we will now examine how various eigenmotions are damped.

#### 4.1.1 The yaw damper: modelling important systems

When an aircraft has a low speed at a high altitude, the Dutch roll properties of the aircraft deteriorate. To prevent this, a **yaw damper** is used. An overview of this system can be seen in figure 4.1. The yaw damper gets its input (feedback) from the **yaw rate gyro**. It then sends a signal to the **rudder servo**. The rudder is then moved in such a way that the Dutch roll is damped much more quickly than usual. As a designer, we can only influence the yaw damper. However, we do need to know how the other systems work as well. For this reason, we model those systems. We usually do assume that the model of the aircraft is known. (Or we use the one that is derived in the Flight Dynamics course.) So, we only examine the other systems.

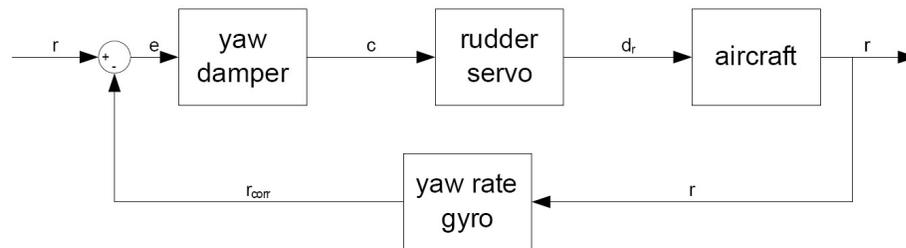


Figure 4.1: An overview of the yaw damper system.

First, let's look at the gyro. Gyros are generally very accurate in low frequency measurements, but not so good in high frequency regions. So, we can model our gyro as a low pass filter, being

$$H_{gyro}(s) = \frac{1}{s + \omega_{br}}. \quad (4.1.1)$$

The **gyro break frequency**  $\omega_{br}$  (above which the performance starts to decrease) is quite high. In fact, it usually is higher than any of the important frequencies of the aircraft. Therefore, the gyro can often also be simply modelled as  $H(s) = 1$ . In other words, it can be assumed that the gyro is sufficiently accurate.

Now let's examine the rudder servo actuator. Actuators are always a bit slow to respond: they lag behind the input. So, we model the rudder as a lag transfer function, like

$$H_{servo}(s) = \frac{K_{servo}}{1 + T_{servo}s}. \quad (4.1.2)$$

The time constant  $T_{servo}$  depends on the type of actuator. For slow electric actuators,  $T_{servo} \approx 0.25$ . However, for fast hydraulic actuators,  $T_{servo} \approx 0.05$  to  $0.1$ . This time constant (or equivalently, the servo break frequency  $\omega_{br_{servo}}$ ) can be very important. If it turns out to be different than expected, the results can also be very different. So, it is often worth while to investigate what happens if  $T_{servo}$  varies a bit.

### 4.1.2 The yaw damper: determining the transfer function

Now we'll turn our focus to the yaw damper. We know that the yaw damper has to reduce the yaw rate. But it shouldn't always try to keep the yaw rate at zero. In this case, the pilot will have a hard time to change the heading of the aircraft. Thus, a **reference yaw rate**  $r$  is also supplied to the system. This yaw rate can be calculated from the desired heading rate  $\dot{\psi}$  by using

$$r = \dot{\psi} \cos \theta \cos \phi. \quad (4.1.3)$$

In this equation,  $\theta$  is the pitch angle and  $\phi$  is the roll angle. Both of them thus need to be known. Alternatively, we can also assume that the aircraft is in a horizontal steady turn. In this case, we have

$$L \sin \phi = \frac{mg}{\cos \phi} \sin \phi = mU\dot{\psi} \quad \Rightarrow \quad \psi = \frac{g}{Us} \phi. \quad (4.1.4)$$

In this equation,  $U$  is the forward velocity of the aircraft. Also, note that we have assumed that  $\phi$  is small (by using  $\tan \phi \approx \phi$ ) and that we have transformed the equation to the frequency domain (by replacing  $\dot{\psi}$  by  $s\psi$ ).

But even if we don't know  $r$ , we can still get the system working. In this case, we can use a **washout circuit**, which is much less expensive. We then simply incorporate a washout term in the controller, being

$$H_{washout}(s) = \frac{\tau s}{\tau s + 1}. \quad (4.1.5)$$

This will cause the yaw damper to fight less when a yaw rate is continuously present. In other words, the system 'adjusts' itself to a new desired yaw rate. The time constant  $\tau$  is quite important. For too high values, the pilot will still have to fight the yaw damper. But for too low values, the yaw damper itself doesn't work, because the washout circuit simply adjusts too quickly. A good compromise is often at  $\tau = 4s$ .

Finally, we look at the yaw damper transfer function. In this transfer function, we have proportional, integral and derivative action. If the rise time should be reduced, we use proportional action. If the steady state error needs to be reduced, we add an integral action. And if the transient response needs to be reduced (e.g. to reduce overshoot) we apply a derivative action. In this way, the right values of  $K_p$ ,  $K_I$  and  $K_D$  can be chosen.

Sometimes, the optimal values of the gains  $K_p$ ,  $K_I$  and  $K_D$  differ per flight phase. In this case **gain scheduling** can be applied. The gains then depend on certain relevant parameters, like the velocity  $V$  and the altitude  $h$ . In this way, every flight phase will have the right gains.

### 4.1.3 The pitch damper

When an aircraft flies at a low speed and a high altitude, the short period eigenmotion has a low damping. To compensate for this, a **pitch damper** is used. The pitch damper is in many ways similar to the yaw damper. Also the set-up is similar. Only this time, the elevators and a pitch rate gyro are used, instead of the rudder and a yaw rate gyro. These two parts are modelled by

$$H_{gyro}(s) \approx 1 \quad \text{and} \quad H_{servo}(s) \approx \frac{K_{servo}}{1 + T_{servo}s} \approx \frac{1}{0.25s + 1}. \quad (4.1.6)$$

Just like with the yaw damper, the **reference pitch rate**  $q$  needs to be calculated. This time, this can be done by using

$$L = nW = nmg = mg + mUq \quad \Rightarrow \quad q = \frac{g}{U}(n - 1). \quad (4.1.7)$$

Alternatively, a washout circuit can again be used. This washout circuit again has the function given in equation (4.1.5). Also, a value of  $\tau \approx 4$  is again a good compromise. Just like a yaw damper, also the pitch damper has proportional, integral and derivative actions.

#### 4.1.4 The phugoid damper

To adjust the properties of the phugoid, we can use a **phugoid damper**. It is very similar to the previous two dampers we have seen. However, this damper uses the measured velocity  $U$  as input. Its output is sent to the elevator. The speed sensor and the elevator servo are modelled as

$$H_{V-sensor}(s) \approx 1 \quad \text{and} \quad H_{servo}(s) \approx \frac{K_{servo}}{1 + T_{servo}s} \approx \frac{1}{0.05s + 1}. \quad (4.1.8)$$

Note that for the servo now a break frequency of  $\omega_{br} = 20$  Hz is assumed.

A **reference velocity**  $U$  is also needed by the system. This reference velocity is simply set by the pilot/autopilot. Alternatively, a washout circuit can be used. This washout circuit is the same as those of the yaw and pitch damper. And, just like the previous two dampers, again proportional, integral and derivative actions can be used.

When using a phugoid damper, one should also keep in mind the short period motion properties. Improving the phugoid often means that the short period properties become worse.

## 4.2 Feedback – Acquiring static stability

Before an aircraft can be dynamically stable, it should first be statically stable. In other words, we should have  $C_{m_\alpha} < 0$  and  $C_{n_\beta} > 0$ . Normal aircraft already have this. But very manoeuvrable aircraft, like fighter aircraft, do not. (Remember: less stability generally means more manoeuvrability.) Then how do we make these aircraft statically stable?

### 4.2.1 Angle of attack feedback

To make an aircraft statically stable, feedback is applied. The most important part is the kind of feedback that is used. First, we'll examine **angle of attack feedback** for longitudinal control. In other words, the angle of attack  $\alpha$  is used as a feedback parameter. First, we have to model the angle of attack sensor and the (canard) servo actuator. This is often done using

$$H_{\alpha-sensor}(s) \approx 1 \quad \text{and} \quad H_{servo}(s) \approx \frac{1}{0.025s + 1}. \quad (4.2.1)$$

So, now a break frequency  $\omega_{br} = 40$  is used for the servo.

For angle of attack feedback, usually only a proportional gain  $K_\alpha$  is used. By using the models of the sensor and actuator (and of course also the aircraft), a root locus plot can be made. With this root locus plot, a nice value of the gain  $K_\alpha$  can be chosen. This gain is then used to determine the necessary canard deflection  $\delta_{canard}$ . This is done using

$$\Delta\delta_{canard} = K_\alpha \cdot \Delta\alpha. \quad (4.2.2)$$

However, a check does need to be performed on whether the canard deflections can be achieved. If gust loads can cause a change in angle of attack of  $\Delta\alpha = 1^\circ$  and the maximum canard deflection is  $25^\circ$ , then  $K_\alpha$  should certainly not be bigger than 25, or even be close to it for that matter.

## 4.2.2 Load factor feedback

There is a downside with angle of attack feedback. It is often hard to measure  $\alpha$  accurately. So instead, **load factor feedback** can be applied. Now the value of  $n$  is used as feedback. As models for the sensor and actuator, we again use

$$H_{n-sensor}(s) \approx 1 \quad \text{and} \quad H_{servo}(s) \approx \frac{1}{0.025s + 1}. \quad (4.2.3)$$

We also need a model for the aircraft. Normally, we assume that such a model is known. However, the transfer function between the load factor  $n$  and the canard deflection  $\delta_c$  is usually not part of the aircraft model. So, we simply derive it. For that, we first can use

$$\Delta n = \frac{\dot{w}}{g} = \frac{U \tan \dot{\gamma}}{g} \approx \frac{U \dot{\gamma}}{g} = \frac{U \gamma s}{g}. \quad (4.2.4)$$

We now divide the equation by  $\delta_c$ . If we also use  $\gamma = \theta - \alpha$ , then we find that

$$\frac{n(s)}{\delta_c(s)} \approx \frac{Us}{g} \left( \frac{\theta(s)}{\delta_c(s)} - \frac{\alpha(s)}{\delta_c(s)} \right). \quad (4.2.5)$$

The transfer functions from  $\delta_c$  to both  $\theta$  and  $\alpha$  usually are part of the aircraft model. So we assume that they are known. The transfer function between  $n$  and  $\delta_c$  is thus now also known. All that is left for us to do is choose an appropriate gain  $K_n$ . And of course, again it needs to be checked whether this gain  $K_n$  doesn't result into too big canard deflections.

The load factor sensor also has a downside. It is often hard to distinguish important accelerations (like the ones caused by turbulence) from unimportant accelerations (like vibrations due to, for example, a firing gun). Good filters need to be used to make sure a useful signal is obtained.

## 4.2.3 Sideslip feedback

Previously we have considered longitudinal stability. For lateral stability, **sideslip feedback** can be used. (However, sideslip feedback is not yet applied in practice.) With sideslip feedback, the sideslip angle  $\beta$  is used as feedback parameter for the rudder. The  $\beta$ -sensor and the rudder are usually modelled as

$$H_{\beta-sensor}(s) \approx 1 \quad \text{and} \quad H_{servo}(s) \approx \frac{1}{0.05s + 1}. \quad (4.2.6)$$

The transfer function between the sideslip angle  $\beta$  and the rudder deflection  $\delta_r$  usually follows from the airplane model. Now that the model is in place, a nice gain  $K_\beta$  can be chosen for the system. This should then give it the right properties.

There is a small problem with sideslip feedback. It can generate a lateral phugoid mode of vibration. To compensate for this, another feedback loop is often used, where the roll rate is used as feedback for the ailerons. This then reduces the effects of the lateral phugoid motion.

## 5. Basic autopilot systems

Previously, we have looked at the stability augmentation system. This system can be seen as the **inner loop** of the aircraft control system. In this chapter, we focus on the **outer loop**: the control augmentation system. When we want to keep a certain pitch angle, velocity, roll angle, heading, or something similar, then we use the CAS. In this way, the pilot workload can be reduced significantly. First, we'll examine holding longitudinal parameters. Second, we'll examine the lateral parameters as well.

### 5.1 Basic longitudinal autopilot systems

We will now examine how we can hold the pitch attitude, the altitude, the airspeed and the climb/descent rate constant.

#### 5.1.1 Holding the pitch attitude

The **pitch attitude hold mode** prevents pilots from constantly having to control the pitch attitude. Especially in turbulent air, this can get tiring for the pilot. This system uses the data from the vertical gyroscope as input (feedback). It then controls the aircraft through the elevators. To be more precise, it sends a signal to the SAS, which then again uses this as a reference signal to control the servo. An overview of the system can be seen in figure 5.1.

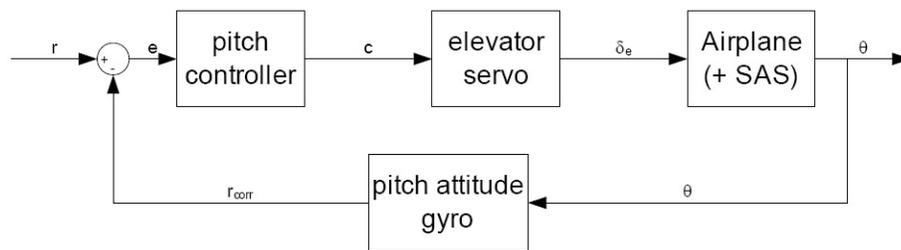


Figure 5.1: An overview of the pitch attitude holding system.

We assume that the model of the aircraft, together with the SAS, is known. But, just like in the previous chapter, we do need to model the gyro and the elevator servo. The relations used for this are

$$H_{gyro}(s) \approx 1 \quad \text{and} \quad H_{servo}(s) \approx \frac{1}{\tau_{servo}s + 1}. \quad (5.1.1)$$

(Although possibly we don't have to take into account the elevator servo. This is the case if the servo is already modelled in the SAS of the aircraft.)

There is also the reference pitch angle  $\theta$  that needs to be set. This is done when the pitch attitude hold mode is activated. In fact, the hold mode usually tries to keep the current pitch angle. The reference pitch angle  $\theta$  is thus the pitch angle that was present at the moment that the hold mode was activated.

Finally, we need to design the pitch controller block. It consists of a proportional, an integral and a derivative action. All that we need to do as designers is choose the right gains  $K_p$ ,  $K_I$  and  $K_D$ . However, once we have done this, we do need to check whether the aircraft meets all the requirements. It can happen that, with the new gains, the damping ratio of (for example) the short period motion has shifted a bit. If it falls outside of the requirements, the SAS of the aircraft needs to be adjusted.

### 5.1.2 Holding the altitude

The **altitude hold mode** prevents pilots from constantly having to maintain their altitude. The input (feedback) comes from the altimeter. The system then uses the elevator to control the altitude. The way in which the altimeter is modelled depends on the type of altimeter. For a radar or GPS altimeter, we use  $H_{altimeter} \approx 1$ . However, for a barometric altimeter, we include a lag. Thus,

$$H_{altimeter}(s) \approx \frac{1}{\tau_{altimeter}s + 1}. \quad (5.1.2)$$

The reference value of the height  $h$  is set in the mode control panel.

To control  $h$ , we must have some expression for  $h$  in our aircraft model. But  $h$  isn't one of the parameters in the basic state space model of the aircraft. So, we need to derive an expression for it. This is done, using

$$\dot{h} = V \sin \gamma \approx V \gamma \quad \Rightarrow \quad h(s) = \frac{V}{s} \gamma(s) = \frac{V}{s} (\theta(s) - \alpha(s)). \quad (5.1.3)$$

When a constant gain is used for the altitude controller, the phugoid may become unstable. (This can especially happen if a low gain is used, and if the original phugoid was already lightly damped.) To prevent this problem from occurring, several options are possible. We could for example use **vertical acceleration feedback**, or we could use **lead-lag compensation**.

The altitude hold mode also consists of a proportional, an integral and a derivative action. However, often it turns out that an integral action is not necessary. And since we generally need to keep controllers as simple as possible, we therefore simply use a PD controller.

### 5.1.3 Holding the airspeed

The **airspeed hold mode** holds a certain airspeed. It uses the airspeed sensor as input and it controls the throttle. Of course, we need to model the airspeed sensor. For GPS airspeed calculations, we can use  $H_{V-sensor}(s) \approx 1$ . However, if we use a pitot-static tube, then we use

$$H_{V-sensor}(s) \approx \frac{1}{\tau_{V-sensor}s + 1}. \quad (5.1.4)$$

Next to the sensor, there is also the engine servo and the engine itself. Both have a bit of lag. We thus model them as

$$H_{servo}(s) = \frac{1}{\tau_{servo}s + 1} \quad \text{and} \quad H_{engine}(s) = \frac{\Delta T(s)}{\delta_T(s)} = K_T \frac{1}{\tau_{engine}s + 1}. \quad (5.1.5)$$

Taking the engine model into account might seem complicated. Luckily, there is an alternative. We can also include the engine effects in the state space model. If we do that, then we add a term  $K_{th} \delta_T$  to the equation for  $\dot{u}$ . This term then represents the thrust, due to the throttle setting. If we do that, then we only have to use the model of the engine servo.

The reference value of the velocity  $V$  is often set at the mode control panel. Alternatively, it can be derived from the actions of the pilot. For example, if the pilot manually pushes the throttle forward, the computer increases the desired (reference) velocity  $V$ .

### 5.1.4 Holding the climb or descent rate

The **flight path angle hold mode** is similar to the pitch attitude hold mode. However, this time the flight path angle/climb rate is kept constant. As input (feedback), the flight path angle  $\gamma$  is used. However,  $\gamma$  can't be measured directly. So, we use  $\gamma = \theta - \alpha$ .  $\theta$  can be measured using a gyro, while  $\alpha$

follows from an angle of attack sensor. The flight path angle hold mode eventually uses the elevators to control the flight path angle.

Of course, the sensors need to be modelled. But we can simply model both the gyro and the  $\alpha$ -sensor with  $H_{sensor} \approx 1$ . We don't need to take into account the elevator servo, since that is already modelled in the SAS of the aircraft.

The flight path angle hold mode controller again consists of proportional, integral and derivative actions. But this time, the derivative action is often not required. Transient behaviour is mostly acceptable when adjusting the flight path angle. A steady state error, however, is more troubling. So integral actions are often used.

## 5.2 Basic lateral autopilot systems

It is time to turn our attention to lateral motion. How do we hold the roll angle, the coordinated roll angle and the heading angle constant?

### 5.2.1 The roll angle hold mode

The **roll angle hold mode** prevents the pilot from constantly having to adjust/control the roll angle during a turn. It uses the roll angle gyro as sensor and it effects the ailerons. The roll angle gyro and the aileron servo are again modelled as

$$H_{gyro}(s) \approx 1 \quad \text{and} \quad H_{servo}(s) \approx \frac{1}{\tau_{servo}s + 1}. \quad (5.2.1)$$

The roll angle that is used as reference angle is defined on the mode control panel.

When modelling the aircraft, it is often assumed that rolling is the only degree of freedom. This reduced model significantly simplifies matters. In fact, the transfer function between  $\phi(s)$  and  $\delta_a(s)$  becomes

$$\frac{\phi(s)}{\delta_a(s)} = \frac{-L_{\delta_a}}{s(s - L_p)}. \quad (5.2.2)$$

Nevertheless, it is often worth while to check whether the behaviour of the full model (without the simplifications) is much different from that of the reduced model. It can, for instance, occur that the Dutch roll becomes unstable in the full model, whereas the reduced model doesn't indicate this.

### 5.2.2 The coordinated roll angle hold mode

The **coordinated roll angle hold mode** is an extension of the roll angle hold mode. It also tries to make sure that the sideslip angle  $\beta$  is equal to zero. This should result in a coordinated turn, thus giving the aircraft less drag and the passengers more comfort. The coordinated roll angle hold mode uses the sideslip sensor as input (feedback). (That is, in addition to the roll angle gyro that was already used in the roll angle hold mode.) It then sends a signal to the rudder. (In addition to the signal to the aileron that was already present.)

The sideslip sensor is modelled as  $H_{\beta-sensor}(s) \approx 1$ . We don't have to model the rudder servo anymore, as this was already incorporated in the inner-loop SAS. (To be more precise, in the yaw damper.) The sideslip angle  $\beta$  that is used as reference input is always simply zero: we do not want any sideslip in a coordinated turn.

Let's ask ourselves, how do we measure  $\beta$ ? We can use a vane-type sideslip sensor (like for the angle of attack). However, the signal from such a sensor is easily distorted, due to for example aerodynamic

effects. Instead, we can also use the lateral acceleration  $A_Y$ . This then gives us

$$mA_Y = Y = C_Y \frac{1}{2} \rho V^2 S \approx C_{Y_\beta} \beta \frac{1}{2} \rho V^2 S \quad \Rightarrow \quad \beta = \frac{2mA_Y}{C_{Y_\beta} \rho V^2 S}. \quad (5.2.3)$$

We can then use this expression to find the sideslip angle  $\beta$ . Do note that we have approximated  $C_Y$  as  $C_{Y_\beta} \beta$ . In other words, we're neglecting the effects of  $p$ ,  $r$ ,  $\delta_a$  and  $\delta_r$  on  $C_Y$ .

### 5.2.3 The heading angle control mode

The **heading angle control mode** controls the heading. It does this by giving the aircraft a roll angle. In fact, it sends a signal to the (coordinated) roll angle hold mode, telling it which roll angle the aircraft should have. This roll angle is maintained until the desired heading is achieved. As sensor, this system uses the directional gyro, modelled as  $H_{gyro}(s) \approx 1$ . Its output effects the ailerons. (The latter is evident, since the system controls the roll angle hold mode.)

The reference angle  $\psi$  is defined by the pilot, through the mode control panel. There is, however, a problem. In our aircraft model, we don't have  $\psi$  as one of the state parameters. To find it, we can use the equation

$$\dot{\psi} = q \frac{\sin \phi}{\cos \theta} + r \frac{\cos \phi}{\cos \theta}. \quad (5.2.4)$$

Let's simplify this a bit. First, we assume that  $q = 0$ . (That is, we're not pitching during the turn.) Second, we assume that  $\theta$  is constant. Third, we assume that  $\phi$  is small, implying that  $\cos \phi \approx 1$ . This then gives

$$\dot{\psi} = \frac{r}{\cos \theta} \quad \text{or} \quad \dot{\psi} = \frac{r}{s \cos \theta}. \quad (5.2.5)$$

Alternatively, we can also use the relation  $\dot{\psi} = \frac{g}{U_s} \phi$  that was derived in the previous chapter. Since the latter relation is based on a lot less assumptions, it is mostly preferred.

## 6. Navigational autopilot systems

In this chapter, we'll consider more advanced autopilot systems. So, the airplane is not only going to hold a certain parameter. Instead, it's going to fly on its own. Examples of such manoeuvres are following a glide slope, automatically flaring during landing, following a localizer or following a VOR beacon. We'll examine all these actions in this chapter. First, we start with the longitudinal actions. Later on, we'll consider the lateral actions.

### 6.1 Longitudinal navigational autopilot systems

#### 6.1.1 The glide slope hold mode

The **glide slope hold mode** is a system that automatically follows a glide slope. This reduces the pilot workload, and moreover it is more accurate than when the pilot follows the glide slope.

Before we're going to discuss the glide slope hold mode, we first make some assumptions. We assume that the glide slope antenna is positioned at the aircraft CG. This antenna measures the **glide slope error angle**  $\Gamma$ . (We model this sensor as  $H_{glideslope\ receiver} \approx 1$ .) The CG of the aircraft is then driven along the glide slope. To accomplish this, the airplane is kept on the glide slope using pitch attitude control. The airspeed is controlled using the autothrottle. (So we also assume that pitch attitude control and airspeed control are already present. This makes sense, as we've already discussed them in the previous chapter.)

Let's denote the **deviation** from the glide slope by  $d$ . We can find an expression for it using

$$\dot{d} = V \sin(\gamma + 3^\circ) \approx V(\gamma + 3^\circ) \frac{\pi}{180} \quad \Rightarrow \quad d(s) = \frac{V}{s} \frac{\pi}{180} L(\gamma + 3^\circ). \quad (6.1.1)$$

(In the above equation, the  $L(\dots)$  denotes the Laplace transform.) Of course, we also need some kind of feedback. But we can't measure  $d$ . Instead, we measure the error angle  $\Gamma$ . This angle is related to the deviation  $d$  according to

$$\Gamma \approx \sin \Gamma = \frac{d}{R} \frac{180}{\pi}. \quad (6.1.2)$$

Here,  $R$  is the **slant range**. Based on the measured error angle  $\Gamma$  (which should of course be kept at zero), we calculate a desired pitch angle  $\theta$ . We then pass this angle on to the pitch attitude control system. The desired pitch angle is calculated using a **glide slope coupler**. Its transfer function is

$$H_{coupler}(s) = K_c \left( 1 + \frac{W_1}{s} \right). \quad (6.1.3)$$

In this equation,  $K_c$  is the **coupler gain**. It needs to be chosen such that we have acceptable closed loop behaviour. In fact, it is the only actual parameter that we as designers can control. Also,  $W_1$  is a **weighting constant**. It is present to cope with turbulence and such. Usually, a value of  $W_1 = 0.1$  is prescribed.

In our model, we also need to know the relation/transfer function between  $\theta$  and  $\gamma$ . This relation  $\gamma(s)/\theta(s)$  can't be obtained from the aircraft model directly. Instead, we use

$$\frac{\gamma(s)}{\theta(s)} = 1 - \frac{\alpha(s)}{\theta(s)} = 1 - \frac{\alpha(s)/\delta_e(s)}{\theta(s)/\delta_e(s)} = 1 - \frac{N_\alpha(s)}{N_\theta(s)}. \quad (6.1.4)$$

The glide slope hold mode does have a problem. When the slant range  $R$  changes, also the properties of the system change. In fact, if the gain  $K_c$  remains constant, then the closer the aircraft, the worse the performance becomes. Luckily, several solutions for this problem are available. We can apply some sort

of gain scheduling: we let  $K_c$  depend on the distance measured by the DME beacon. Or even simpler but less accurate, we let it depend on time. Finally, we can also add a lead-lag compensator to the system. If done well, this can reduce the effects of this problem significantly.

When designing an autopilot, it should be made robust. In other words, if certain parameters change, the autopilot should still work. Parameters that are subject to change in the real world are the airplane CG location, the airplane weight, the airplane speed and the presence/intensity of turbulence. The autopilot should be able to cope with these variations.

### 6.1.2 Automatic flare mode

Getting the right vertical velocity on landing is difficult. The velocity shouldn't be too high. Such **hard landings** ( $\dot{h} \leq -6$  ft/s) are challenging for both the landing gear and the passengers. As such, they're not really acceptable. Too **soft landings** ( $\dot{h} \approx 0$  ft/s) are however also undesirable, as there will be **floatation** of the aircraft. Ideally, we have a **firm landing** with  $\dot{h} = -2$  to  $-3$  ft/s.

The relationship between the normal velocity and the vertical velocity during landing is usually  $\dot{h} = -V \sin 3^\circ$ . So, the faster an aircraft flies, the harder the touchdown. This can be a problem for airplanes with a low minimum speed, which thus have to fly fast. Therefore, such aircraft usually **flare** right before touching down: they pull up their nose. By doing this, the airplane follows the so-called **flare path**. This path starts at the height  $h_{flare}$ . It ends (by touching down) 1100 ft further than the point where the glide slope ends. (That is, where the glide slope antenna is positioned.)

The airplane is kept on the flare path by the pitch attitude control system. But this system of course needs to have some input. For that, we approximate the flare path by

$$h = h_{flare} e^{-t/\tau}. \quad (6.1.5)$$

All that we need to find are the constants  $h_{flare}$  and  $\tau$ . They both depend on the time  $t_{td}$  between the start of the flare and touchdown. To see how, we first examine the horizontal distance which the airplane travels during the flare manoeuvre. This is

$$V t_{td} = 1100 + \frac{h_{flare}}{\tan 3^\circ}. \quad (6.1.6)$$

From this, we can derive the height  $h_{flare}$ . To also find the time constant  $\tau$ , we differentiate the equation for  $h$ . This gives

$$\dot{h} = -\frac{h_{flare}}{\tau} e^{-t/\tau} = -\frac{h}{\tau} \quad \Rightarrow \quad h_{flare} = -\dot{h}_{at\ h_{flare}} \tau. \quad (6.1.7)$$

Okay, we do need to know the vertical velocity  $\dot{h}_{at\ h_{flare}}$  at the flare height. But this can simply be found using  $\dot{h} = -V \sin 3^\circ$ . And once we know  $\tau$ , we will have the control law for our automatic flare mode system:  $\dot{h} = -h/\tau$ .

But how do we make sure that the aircraft stays at the correct altitude? Well, we know the vertical speed of the aircraft  $\dot{h}$ . (It can be measured.) We also know the desired vertical airspeed, which follows from our control law. Based on the difference, we calculate a desired pitch angle  $\theta$ . This is done using a **coupler**. Its transfer function is again given by

$$H_{coupler}(s) = K_c \left( 1 + \frac{W_1}{s} \right). \quad (6.1.8)$$

$K_c$  is again the coupler gain and  $W_1 = 0.1$  is again the weighting constant. The desired pitch angle is then passed on to the pitch attitude control system.

In our system, we do need a model of the aircraft. How do changes in the pitch angle effect the vertical velocity? We can find the transfer function between these two parameters using

$$\dot{h} \approx V\gamma \quad \Rightarrow \quad \frac{\dot{h}(s)}{\theta(s)} = \frac{\gamma(s)}{\theta(s)} V. \quad (6.1.9)$$

Earlier in this chapter we already derived an expression for  $\gamma(s)/\theta(s)$ . So we can apply that again here. For the automatic flare mode to work, an accurate altitude measurement system is required. A radar altimeter is usually sufficiently accurate. In fact, when it is used, the system is often so precise that aircraft always land on exactly the same spot. This often resulted in runway damage at that point. To prevent this, a **Monte Carlo scheme** is used. The automatic flare mode system now chooses a random point inside a certain acceptable box. It then makes sure that the airplane touches down at that point. This method effectively solves the problem.

## 6.2 Lateral navigational autopilot systems

### 6.2.1 The localizer hold mode

During an instrument landing, pilots need to follow the ILS localizer. But the **localizer hold mode** system can perform this much more accurately. Plus, it reduces the pilot workload.

Similar to the glide slope hold mode, we first need to make some assumptions. We assume that the airplane CG follows the localizer beam centerline. Also, we assume that the **localizer error angle**  $\lambda$  is sensed by the on-board localizer receiver. The airplane is then kept on the centerline using the heading angle controller (which we assume to be present).

We again denote the deviation from the intended path by  $d$ . We now have

$$\dot{d}(s) = V \sin(\psi(s) - \psi_{ref}(s)) \approx V(\psi(s) - \psi_{ref}(s)) \quad \Rightarrow \quad d(s) = \frac{V}{s}(\psi(s) - \psi_{ref}(s)). \quad (6.2.1)$$

In this equation,  $\psi_{ref}$  is the reference heading angle. It is the heading angle which we want to have. In other words, it is the heading angle of the runway.

We also need to have some feedback. For this, we can use the localizer error angle  $\lambda$ . It is related to the deviation  $d$  according to

$$\lambda \approx \sin \lambda = \frac{d}{R} \frac{180}{\pi}. \quad (6.2.2)$$

Based on the measured error angle  $\lambda$  (which should of course be kept at zero), we calculate a desired heading angle  $\psi$ . This desired heading angle is then passed on to the heading angle controller. The desired heading angle is calculated using a **coupler**. Its transfer function is again given by

$$H_{coupler}(s) = K_c \left( 1 + \frac{W_1}{s} \right). \quad (6.2.3)$$

$K_c$  is again the coupler gain and  $W_1 = 0.1$  is again the weighting constant.

Just like the glide slope hold mode, also the localizer hold mode has a problem. When the slant range  $R$  becomes too small, dynamic instability may occur. So, again the gain  $K_c$  needs to depend on the slant range  $R$ . Or alternatively, a compensating network  $H_{compensator}(s)$  needs to be added. Luckily, the localizer doesn't have to work for slant ranges smaller than  $R = 1$  nm. The reason for this is that the localizer antenna is at the end of the runway, while the aircraft already touches down near the start of the runway.

### 6.2.2 The VOR hold mode

The **VOR hold mode** tries to follow a certain VOR radial. The working principle of following the VOR radial is similar to the principle of following the ILS localizer path. This time, the VOR error angle  $\lambda$  is used as feedback, and should be kept at zero.

There are, of course, a few differences. The VOR transmitter has a bandwidth of  $360^\circ$ , whereas the ILS localizer only has a bandwidth of  $5^\circ$ . (The localizer only works when the aircraft is more or less in line with the runway.) Also, the range of possible slant ranges  $R$  is much different for the VOR. The maximum range of a VOR beacon is roughly 200 nm. Next to that, when an aircraft is flying at 6000 ft, the slant range simply can't become less than 6000 ft  $\approx$  1 nm. So, aircraft hardly ever come closer than 1 nm to a VOR beacon. The final difference between the VOR and the localizer is that, when an aircraft is above the VOR, it doesn't receive a signal. (The aircraft is in the so-called **cone of silence**.) The VOR hold mode system should be able to cope with that.