

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.

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.

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 1.

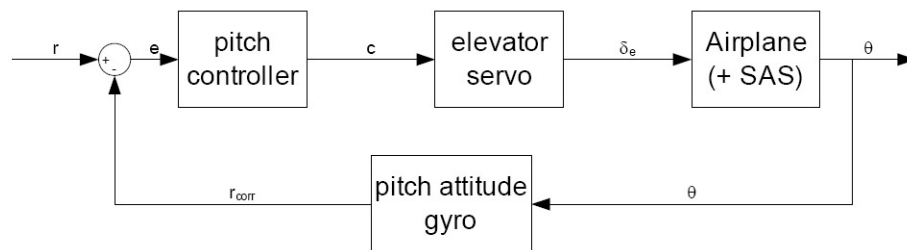


Figure 1: An overview of the pitch attitude holding system.

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

$$H_{gyro}(s) \approx 1 \quad \text{and} \quad H_{servo}(s) \approx \frac{1}{\tau_{servo}s + 1}. \quad (1.1)$$

(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 θ 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 θ 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.

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,

$$H_{altimeter}(s) \approx \frac{1}{\tau_{altimeter}s + 1}. \quad (1.2)$$

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

$$\dot{h} = V \sin \gamma \approx V \gamma \quad \Rightarrow \quad h(s) = \frac{V}{s} \gamma(s) = \frac{V}{s} (\theta(s) - \alpha(s)). \quad (1.3)$$

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.

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

$$H_{V-sensor}(s) \approx \frac{1}{\tau_{V-sensor}s + 1}. \quad (1.4)$$

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

$$H_{servo}(s) = \frac{1}{\tau_{servo}s + 1} \quad \text{and} \quad H_{engine}(s) = \frac{\Delta T(s)}{\delta_T(s)} = K_T \frac{1}{\tau_{engine}s + 1}. \quad (1.5)$$

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 .

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 γ is used. However, γ can't be measured directly. So, we use $\gamma = \theta - \alpha$. θ can be measured using a gyro, while α

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 α -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.

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?

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

$$H_{gyro}(s) \approx 1 \quad \text{and} \quad H_{servo}(s) \approx \frac{1}{\tau_{servo}s + 1}. \quad (2.1)$$

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

$$\frac{\phi(s)}{\delta_a(s)} = \frac{-L_{\delta_a}}{s(s - L_p)}. \quad (2.2)$$

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.

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 β 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 β 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 β ? 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

$$mA_Y = Y = C_Y \frac{1}{2} \rho V^2 S \approx C_{Y_\beta} \beta \frac{1}{2} \rho V^2 S \quad \Rightarrow \quad \beta = \frac{2mA_Y}{C_{Y_\beta} \rho V^2 S}. \quad (2.3)$$

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

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

$$\dot{\psi} = q \frac{\sin \phi}{\cos \theta} + r \frac{\cos \phi}{\cos \theta}. \quad (2.4)$$

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 θ is constant. Third, we assume that ϕ is small, implying that $\cos \phi \approx 1$. This then gives

$$\dot{\psi} = \frac{r}{\cos \theta} \quad \text{or} \quad \psi = \frac{r}{s \cos \theta}. \quad (2.5)$$

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