Modeling humans

For normal control systems, we already have good mathematical descriptions. (Think, for example, of linear control theory.) But when humans are involved, this becomes more complicated. One way to solve this problem is by trying to model humans as normal control systems. By doing this, we can explain and predict the behavior of the human and the system. In this chapter, we'll look at how it works.

1 A human as a linear controller

1.1 The human pilot

The human pilot is a **multimode**, **adaptive** and **learning** controller. It is capable of exhibiting an enormous variety of behavior, like...

- **System organization** – This means that the human controller can detect coherence between for example input and output signals. It then uses this coherence to control the system.
- **System adjustment** – The human controller can adjust the way in which it controls the system (i.e. its transfer function) such that the system is adequatly controlled.

1.2 Display types

Let's examine a very basic single-input single-output feedback system. It is the task of the human controller to minimize the error $e = y - y_{ref}$ between the output y and the reference signal y_{ref} of the system. The way in which the pilot can do this strongly depends on how the data is presented to him.

- In a **compensatory display**, only the error e is shown. The controller simply needs to compensate for this error.
- In the **pursuit display**, only the current output y and the reference signal y_{ref} are shown. This time the controller needs to let the output y pursuit the reference output y_{ref}.
- In the **preview display**, the controller doesn't only see y and y_{ref}. It also sees the future values of y_{ref}. In this way, the pilot can already keep in mind future changes of the reference signal.

Soon we will discuss the cross-over model of human behavior. This model only applies to the compensatory display. So it does not work for the other displays. Keep that in mind.

1.3 Human controllers as linear models

How can we model the human controller in the SISO feedback system with compensatory display? Of course, the human controller is nonlinear. However, research has shown that it can be modeled using a quasi-linear describing function. This function Y_p (known as the **causal model**) is basically a linear differential equation with constant coefficients and a time delay. It accounts for the portion of the controller's output c that is linearly related to the input e.

Of course, the describing function doesn't model the human behavior perfectly. The remaining inaccuracies are described by the **remnant model** n, which is a stationary noise process. Since this remnant model is not really understood well, we will focus on the describing function Y_p.

1.4 The ideal solution

If we don't use a human controller, what should our causal model Y_p be to ideally control the system? A good feedback control system provides a good relation between y and y_{ref}, it suppresses disturbances, is robust and provides adequate closed-loop stability margins. If we call the plant transfer function Y_c, then the system closed-loop transfer function is

$$\frac{y}{y_{ref}} = \frac{Y_p Y_c}{1 + Y_p Y_c} = \frac{Y_{OL}}{1 + Y_{OL}}, \tag{1.1}$$

with Y_{OL} the open loop transfer function. To have $y/y_{ref} \approx 1$, we can make $|Y_{OL}|$ very big for the input bandwidth and very small for the other frequencies. This normally works well, unless some lag is also added. In this case, we often wind up with an unstable system.

To solve this problem, we must make sure that the gain crossover frequency ω_c exceeds the maximum input frequency ω_i. (Remember that ω_c satisfies $|Y_{OL}(j\omega_c)| = 1$.) So within the bandwidth, $|Y_{OL}|$ is roughly constant, and much bigger than 1. Near the crossover frequency, $|Y_{OL}|$ will have a slope of -20dB/decade. Having this kind of open-loop transfer function will result in a stable system with small tracking errors. The value of Y_{OL} near ω_c now determines the dominant closed loop modes. In fact, the system stability strongly depends on the phase at ω_c. (That is, the phase margin.)

2 Ways of modeling the pilot

2.1 The crossover model

Previously, we haven't answered the question what Y_p will be if we use a human controller. This is where the crossover model theorem comes in. The **crossover model theorem** states that human controllers adjust their control behavior to the dynamics of the controlled element. They do this such that the open loop transfer function in the crossover region (i.e. with ω near ω_c) can be described by

$$Y_{OL}(j\omega) = Y_p(j\omega)Y_c(j\omega) = \frac{\omega_c}{j\omega} e^{-j\omega\tau_e}. \tag{2.1}$$

Here, ω_c is still the crossover frequency. τ_e is an **effective time delay**, which represents the time lags of the human operator. It is now interesting to note that, if we know the dynamics of the system, then we can find the transfer function of the human operator near ω_c. It is given by

$$Y_p(j\omega) = \frac{\omega_c}{Y_c(j\omega)j\omega} e^{-j\omega\tau_e}. \tag{2.2}$$

What does this imply for the performance and the stability of the system? Well, the **performance** is determined by ω_c, while the **stability** is determined by the phase margin ϕ_m. The crossover model now implies that $\phi_m = \frac{\pi}{2} - \omega_c\tau_e$. This means that, if the time lag τ_e is high, we cannot have a high crossover frequency ω_c. (Or instability can occur.) So big time lags lower the performance of the system.

2.2 A pilot describing function

Another way to model the pilot is by using the **simplified precision model**. In this model, we model the pilot as

$$Y_p(j\omega) = K_p \frac{1 + \tau_L j\omega}{1 + \tau_I j\omega} e^{-j\omega\tau_e}, \tag{2.3}$$

where K_p is the **pilot gain**, τ_L is the **lead time constant**, τ_I is the **lag time constant** and τ_e is the **effective time delay**. Basically, the pilot gain K_p puts the crossover frequency ω_c on the right position.

The time constants τ_L and τ_I are then used to give Y_{OL} a slope of -20dB/decade near the crossover frequency. The question remains how to set these parameters. There are 6 so-called **verbal adjustment rules** for that.

1. In **equalization selection and adjustment**, we want $|Y_{OL}|$ to be very big for low frequencies. For frequencies near the crossover frequency, we want the slope to be -20dB/decade.

2. Within the limitations of the pilot, we should minimize errors. The most important way to do this is to minimize τ_e as far as possible. How much this can be done depends on the input frequency ω_i. If $\omega_i \to 0$, then τ_e becomes the **basic time delay** τ_0. Also, the corresponding crossover frequency is denoted by ω_{c_0}.

3. We have to set the right crossover frequency. A good estimate is

$$\omega_{c_0} = \frac{\pi}{2\tau_0}. \tag{2.4}$$

4. If $\omega_i < 0.8\omega_{c_0}$, then changes in the input frequency ω_i don't really influence the crossover frequency ω_c. However, if ω_i becomes bigger than $0.8\omega_{c_0}$, then resonance may occur. To prevent this, the pilot should choose ω_c much lower than ω_{c_0}. This is called **regression**. It simply means that the pilot does not try to follow the high-frequency signals. If he does, resonance would occur and the error would be bigger.

5. Once the parameters have been set, we can approximate the squared error using the **one-third law**

$$\frac{e^2}{y_{ref}^2} = \frac{1}{3}\left(\frac{\omega_i}{\omega_c}\right)^2. \tag{2.5}$$

   However, this law is only valid when ω_i is much smaller than ω_c.

6. The time delay τ_e and the optimal crossover frequency ω_c depend on the bandwidth ω_i. If $\omega_i \approx 0$, then the values mainly depend on the controlled system Y_c. However, the changes in τ_e (if ω_i is different from zero) mainly depend on ω_i. So we can write

$$\tau_e(Y_c, \omega_i) \approx \tau_0(Y_c) - \Delta\tau_e(\omega_i). \tag{2.6}$$

   Furthermore, we have

$$\omega_c(Y_c, \omega_i) \approx \omega_{c_0}(Y_c) + 0.18\omega_i. \tag{2.7}$$

With these verbal adjustment rules, we can get a pilot model without doing any experiments. This is very convenient for initial pilot-in-the-loop experiments.