Human-Machine Systems Summary

1. Basics of human-machine interaction

To start off our journey in the wondrous world of human-machine interaction, we first need to learn some basic principles. What is human-machine interaction, and how can we characterize it? That is discussed here.

1.1 Introduction to human-machine interaction

1.1.1 The history of human-machine interaction

Ever since humans started to build tools, there was interaction between the humans and the machines. This interaction has evolved over time.

Initially, before the second world war, people were adjusted to fit to machines. In other words, humans were **trained** to use the machines. However, in the second world war, new equipment was developed so quickly that it was hard to sufficiently train humans. So the need for a systematic analysis and synthesis of the interaction between humans and machines arose.

The history of human-machine interaction can be split up into four time zones. First, in the years 1940 to 1955, developers tried to find the **limits** of human possibilities. New equipment was designed such that human controllers would just be able to deal with it. From 1955 to 1970, things advanced. In this time, researches tried to model the humans like **machines** and design products accordingly.

Around 1970, electronics were advancing. So from 1970 to 1985, this technology was used to automate a lot of tasks which normally required humans. The human stopped being the controller and started becoming the **supervisor**. This advanced much more since 1985. Now we take into account several complicated things, like **workload**, **cognitive process models**, **situational awareness** and more.

1.1.2 Eliminating human mistakes

Making mistakes is human. But the amount of mistakes needs to be minimized. The amount of mistakes highly depends on the workload. If the workload is too high, humans become tired (**fatigue**) and start to make errors. However, if the workload is too low, humans become bored and make mistakes as well. So, the amount of workload needs to be optimized.

To get rid of human mistakes, engineers initially tried to do this by eliminating the human as much as possible. However, a newer approach is to involve humans in the right way. By making a good joint-cognitive system, the workload is appropriate and the situational awareness is high.

1.1.3 Wickens' model of human information processing

A central topic in human-machine interaction is the way in which humans process information. One way to model this is by using **Wickens' model of human information processing**. This model consists of several parts.

- **Sensory processing** – The input (called **stimuli**) comes from all the human senses (also known as **sensory systems**).

- **Short-term sensory store** – Each sensory system has a mechanism which prolongs any stimulus for a short time after the stimulus has occurred.
- **Perceptual encoding** – The stimuli are assigned to a single perceptual category. This is also known as **detection**, **recognition**, **identification**, **categorization**, **pattern recognition**, etcetera.
- **Decision making** and **response selection** – After categorizing the stimulus, you must decide what to do with it.
- **Response execution** – The response (**action**) is executed.
- **Feedback** and **information flow** – You monitor the consequences of your action.
- **Attention** – You select information sources to be processed. (For example, when you hear a suspicious sound, you decide to pay more attention to your hearing than to your eyesight.)

1.2 Human behavior

1.2.1 Skill, rules and knowledge

Human behavior is often split up into three categories. There are skill-based, rule-based and knowledge-based behavior. (This is the **SRK model**.) Each of these behaviors works differently. Also, they are caused differently. Let's examine these three categories.

Skill-based behavior (SBB) concerns the basic motor performance of your body. You generally don't think about this kind of behavior: it's just something which you learn. An example is standing up straight. Without knowing it, your feet make sure that you continue to stand up straight. Skill-based behavior is caused by (continuous) **signals**. You then subconsciously use these signals to control something. Because this doesn't involve any conscious thought, SBB is a very fast type of behavior.

When applying **rule-based behavior** (RBB), you are using stored rules. These rules are often learned by experience or have been communicated by others. An example of rule-based behavior occurs when a child suddenly crosses the road in front of your car. When this happens, you have learned to break. Rule-based behavior is thus mostly caused by **signs/events**. The rules then imply which event cause which response. RBB is often confused with SBB. But the most important difference is that in RBB you can explain which rules you use, while with SBB you cannot.

Sometimes simple rules won't get you to your goal. In this case, **knowledge-based behavior** (KBB) is used. In this behavior, plans are developed and traded off, such that you remain with the plan which best reaches your goal. An example of knowledge-based behavior is planning a route to a place you haven't been before. KBB is caused by **symbols**. A symbol is a piece of information which can be used for reasoning.

1.2.2 Ergonomics

When designing human-machine interfaces, it is often wise to use ergonomics. **Ergonomics** is the application of scientific information concerning humans to the design of objects and systems for human use. Examples are ergonomic keyboards and displays. When these products were developed, scientific data about humans was used.

When applying ergonomics, you have to look at what kind of behavior is present. When influencing SBB, you need to make sure that the user has a clear and continuous feedback signal. On the other hand, when influencing RBB or KBB, the user needs a clear overview of all relevant data, and must know what it all means.

2. 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.

2.1 A human as a linear controller

2.1.1 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.

2.1.2 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.

2.1.3 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.

2.1.4 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

$$\frac{y}{y_{ref}} = \frac{Y_p Y_c}{1 + Y_p Y_c} = \frac{Y_{OL}}{1 + Y_{OL}}, \tag{2.1.1}$$

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.)

2.2 Ways of modeling the pilot

2.2.1 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

$$Y_{OL}(j\omega) = Y_p(j\omega) Y_c(j\omega) = \frac{\omega_c}{j\omega} e^{-j\omega\tau_e}. \tag{2.2.1}$$

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

$$Y_p(j\omega) = \frac{\omega_c}{Y_c(j\omega) j\omega} e^{-j\omega\tau_e}. \tag{2.2.2}$$

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.

2.2.2 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

$$Y_p(j\omega) = K_p \frac{1 + \tau_L j\omega}{1 + \tau_I j\omega} e^{-j\omega\tau_e}, \tag{2.2.3}$$

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

$$\omega_{c_0} = \frac{\pi}{2\tau_0}. \tag{2.2.4}$$

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**

$$\frac{e^2}{y_{ref}^2} = \frac{1}{3}\left(\frac{\omega_i}{\omega_c}\right)^2. \tag{2.2.5}$$

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

$$\tau_e(Y_c, \omega_i) \approx \tau_0(Y_c) - \Delta\tau_e(\omega_i). \tag{2.2.6}$$

Furthermore, we have

$$\omega_c(Y_c, \omega_i) \approx \omega_{c_0}(Y_c) + 0.18\omega_i. \tag{2.2.7}$$

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.

3. The neuromuscular and vestibular system

A pilot needs his body to influence his environment. And this body is actuated by muscles. This means that, to understand the pilot's behavior, we need to understand how his muscles work. But next to this, a pilot also gets his input from the environment. A very important contributor to this input is the vestibular system. We will also look at how this system works.

3.1 The neuromuscular system

3.1.1 The build-up of the system

The **neuromuscular system** (NMS) consists of three parts. There are the **brain** and the **spinal cord**, which together form the **central nervous system** (CNS). This system provides inputs to the third part: the **muscles**. When the NMS is used to control an aircraft, the NMS actually becomes part of the closed-loop system dynamics.

If we know how the NMS works, we can influence the system dynamics. One way to do this is to change the NMS behavior. This can for example be done by training the pilot. Alternatively, we can adjust the **feedback system** to be better suited for the NMS. For example, the stick feedback can be programmed such that the pilot flies in a more comfortable and/or more effective way.

The way in which the NMS influences the dynamics depends on the type of aircraft. In small aircraft, the control stick is more or less directly connected to the control surfaces. For such aircraft, it is often difficult to influence the feedback system. On the other hand, in fly-by-wire aircraft, a computer is in-between the NMS and the control surfaces. With such aircraft, designers have a lot of freedom varying the feedback system.

There is a possible design problem. When applying an input, accelerations of the controlled vehicle can lead to involuntary movements of the arm/hand. For example, you can push a stick forward to accelerate forward. But because your vehicle accelerates forward, you are automatically pushed back, thus reducing the desired input. When designing a neuromuscular input system, designers should take this into account.

3.1.2 Feedback systems

A **force-feedback system** is a system that provides feedback to the human controller through forces. These forces inform the user of several things, like the actual machine position, the machine limitations, the machine perception and more. Next to this, force feedback can be used to encourage/discourage certain control actions, or to increase the stability of the human-machine interaction.

There are several examples of force-feedback systems.

- In **haptic feedback systems**, the user is provided feedback through the sense of touch. We can, for example, have a haptic control panel. When the human controller pushes this panel, it pushes back. The amount in which it does this is the feedback to the human controller. By using haptic control, humans get better feedback and can thus more accurately control the system.
- In **exoskeleton force feedback**, an external mechanical 'skeleton' is used. By moving, for example, his arm, the human controller provides input. The skeleton can resist these movements to a certain degree, thus providing feedback. This method of feedback allows for a multi-dimensional input, thus increasing the number of possibilities to control the system.
- In **master-slave feedback** the human controls the master machine. This machine then sends the control signals to a slave machine. This slave machine measures its environment and sends its

feedback back to the master. This feedback mechanism is very convenient when machines need to work in situations too dangerous or too big/small for humans.

3.1.3 The build-up of muscles

Let's take a closer look at muscles. Muscles are connected to the bones through **tendons**. Muscles are activated by **motor neurons** (α-neurons). This causes the muscles to contract, thus moving the skeleton of the human. The actual response of the muscles is measured by **sensory neurons** (γ-neurons). This data is then sent back to the brain as feedback.

Muscles are mostly made out of 'muscle fibres' called **myofibrils**. Due to the chemical structure of the myofibrils, muscles can only contract. To enable bones to move in multiple directions, **antagonistic pairs** of muscles are often present in the human body. When one of these muscles (the **flexor**) contracts, tension on the bone increases. This allows rotation in one direction. When the other muscle (the **extensor**) contracts, tension on the bone is released, enabling motion in the other direction.

The γ-neurons measure the muscle stretch and the muscle stretch velocity. These two parameters are very important, since they mainly determine the amount of force that can be generated by the muscle. If the muscle is too short, too long or moving too fast, it is not capable of generating big forces.

Next to γ-neurons, there are also **Golgi tendon organs** (GTOs). A GTO measures how much the tendon is stretched. This amount of stretch is a good indication of the muscle force. This does not directly mean that the muscle is contracted though. Muscle force can also be caused by a passive muscle stretch.

Not all the measured data is sent directly to the brain. There are also 'short-cuts' in the spinal cord. This allows for very fast reactions called **reflexes**. Reflexes enable fast and complex behavior without any conscious effort from the brain. Reflexes can be both innate or acquired through learning. An example is the **stretch reflex**: when a muscle stretches too much, this stretch is counteracted. Another example is the **GTO reflex**: when the muscle force becomes too big, the muscle is relaxed. This prevents damage to the muscle and/or the tendon.

Next to muscles, the body also has **skeletal bones**. These bones provide rigidity and inertia to the body and provide attachment sites for muscle tendons. Finally, there is the **skin**. This flexible layer provides protection of the muscles and the skeleton from the outside world.

3.1.4 Modeling the neuromuscular system

How can we model the muscular system? The basic idea is that we model the muscle as a simple mass-spring-damper system. The mass of the system consists of all relevant body material, like muscles, bones and skin. But the muscle isn't the only part dealing with muscle forces. These force also needs to be transmitted through the skin. We model the skin as a simple massless spring-damper system. The force caused by the whole muscle system is then given by

$$F_s(t) = M_a \ddot{x}_a(t) + B_a \dot{x}_a(t) + K_a x_a(t) + B_s \dot{x}_s(t) + K_s x_s(t), \tag{3.1.1}$$

where x_a denotes the musculo-skeletal deflection, while x_s denotes the skin deflection. The total deflection is thus the sum of these two deflections.

The **admittance** is defined as the ratio x_s/F_s between the hand position x_s and the force F_s. Humans can adjust the admittance. When a human needs to keep the exerted force constant, he uses a high admittance, resulting in a flexible system. On the other hand, if the human needs to keep the position constant, a low admittance is used, causing a stiff system.

There is a way to measure the admittance. First, we need to give the human controller instructions. For example, we can tell him to maintain the position or to maintain the force. We then impose a perturbation force/torque D. As a result of this, the manual controller applies a force/torque F. This results in a pedal displacement X. By measuring D, F and X, we can calculate the admittance. This admittance strongly influences the NMS dynamics. So when a manual control system is designed, the influence of the admittance on the NMS dynamics has to be taken into account.

3.2 The vestibular system

The **vestibular system** is a system in our inner ear. It arranges our feeling for balance and spatial orientation. The vestibular system has several subsystems, each having a specific task. We will look at these individual subsystems.

3.2.1 The otholith organ

The **otolith organ** (OTO) is sensitive to **specific forces**. (The specific force consists of accelerations and gravity.) It is located in both the **utricle** and the **saccule**. The utricle mainly has horizontal sensitivity, while the saccule has vertical sensitivity. The otholith organ can be seen as a membrane which deforms when specific forces are applied. The deformation of the membrane is generally modeled as

$$H(s) = \frac{1}{(\tau_{OTO}s + 1)^2}, \tag{3.2.1}$$

where the time constant τ_{OTO} is small (only a few milliseconds). Another way to model the otholith organ is

$$K\frac{1 + \tau_n s}{(1 + \tau_1 s)(1 + \tau_2 s)}, \tag{3.2.2}$$

where $K = 3.4$, $\tau_n = 1$s, $\tau_1 = 0.5$s and $\tau_2 = 0.016$s. The otholith organ measures both tonic and phasic units. **Tonic** units concern the magnitude of the acceleration, while **phasic** units concern changes in the acceleration.

3.2.2 Semi-circular canals

The vestibular system also has three **semi-circular canals** (SCC). (One for each direction.) These canals are filled with a fluid. When there is a rotational velocity, the fluid moves in the canals. Small hairs detect this shift. So the semi-circular canals measure angular velocities. Modeling the SCC can be done using

$$H_{SCC}(s) = \frac{1 + \tau_L s}{(1 + \tau_1 s)(1 + \tau_2 s)}, \tag{3.2.3}$$

where $\tau_L = 0.1097$ s, $\tau_1 = 5.924$ s and $\tau_2 = 0.005$ s. The frequency range of the SCC is from 0.1 rad/s to 10 rad/s. The output is proportional to the rotational velocity.

However, not only your vestibular system tells you that you are moving. Often your eyes tell you this too. If you move, then your environment appears to move around you. Your eyes see this and inform your brain of the motion. This perception of self-motion due to visual stimuli is called **vection**. An exception occurs when your environment moves, but you do not. In this case, your brain will get a wrong signal. This is called **retinal slip**.

The signals from your visual system and from the vestibular system are both feedback for your eyes. Using this information, the two images from both eyes are merged and missing pieces of information are filled in.

3.2.3 The subjective vertical

You always have a notation of your orientation. This is captured in the **subjective vertical** (SV). This is the vector of which you believe is straight down. If the SV strongly deviates from the actual vertical, then you are disoriented.

But how does the vestibular system know what the vertical is? For this, it uses both the otholith organ and the semi-circular canals. The otholith organ detects specific forces. So it is simply able to find out which direction the specific force points to. From this it can derive the direction of gravity. However, simple accelerations of the body should not be taken into account. And this is where the semi-circular canals come in. If the direction of the specific force changes, but you don't rotate, then you must be accelerating. So, based on the specific forces of the OTO and the rotational velocities indicated by the SCC, the subjective vertical is updated.

You might be wondering: don't you use visual information as well to determine the SV? In fact, you do. Let's call $\hat{\omega}_{vest}$ the rotations as indicated by the vestibular system and $\hat{\omega}_{vis}$ the rotations as indicated by the visual system. These two signals need to be merged. This is modeled according to the **visual attractor** model, which states that the processed rotations $\hat{\omega}_{total}$ are given by

$$\hat{\omega}_{total} = (\hat{\omega}_{vis} - \hat{\omega}_{vest})H_{VA}(j\omega) + \hat{\omega}_{vest}. \tag{3.2.4}$$

Here, the transfer function $H_{VA}(j\omega)$ is a low pass filter. So when the vestibular system and the visual system give different signals, your brain initially believes the vestibular system, but then slowly starts to believe the visual system.

Next to the visual system and the vestibular system, there is also the **proprioceptive system**. This system consists of skin pressure receptors. It measures forces exerted on the body. Based on these forces, your brain makes another estimate of your motion. And again, these signals are merged to get an output that is as meaningful as possible.

3.2.4 Modeling manual control

When modeling manual control, we also need to take into account how the body acquires its data. We have seen that it mainly does this through the vestibular system and the visual system. An important difference is that the vestibular system generally measures the actual orientation y, while the visual system measures the orientation difference $e = y_{ref} - y$. So, the two systems have different input signals.

It also turns out that the vestibular system is faster than the visual system. That is, it has a lead with respect to the visual system. In other words, the perception delay is lower for the vestibular system. Especially when following a target, this means that the vestibular system contributes to the inner loop, while the visual system makes up the outer loop of the human controller.

4. The visual system

In the previous chapter, we have already mentioned the **visual system**. Now we pay more attention to it. The visual system consists of the eyes and everything directly related to them. How does it work? And what principles should we use when designing displays for it? That's what this chapter is about.

4.1 Setting up visual displays

4.1.1 The build-up of the visual system

Vision can be split up into foveal and peripheral vision. In your **fovea**, you have a lot of **cones**. (Cones detect color.) Your acuity is large, and you can thus see a lot of detail. In your periphery, you don't have cones. Instead, there are only **rods**. (Rods can't detect color, but can detect brightness and motion.) Although the acuity of the periphery is small, it is very good at detecting motion.

Let's examine the foveal vision a bit closer. This type of vision concerns the focussing of the eye. The eye can try to follow a certain moving object. We then have **pursuit eye movements**. Alternatively, the eye can jump from one place to another place. We are then dealing with **saccadic eye movements**. Important parameters are now the **location** of the focus, as well as the **dwell time**, which is the amount of time which the eye focusses on a certain point.

4.1.2 The eye focus of the pilot

Examine a pilot in a cockpit. When an emergency occurs, the pilot needs to be notified. Visual cues can be used for this. But these visual cues will quite probably not be in the foveal vision of the pilot, but in the peripheral vision. So to make sure that the pilot sees it, motion needs to be involved. A good way to warn the pilot is thus to use blinking lights.

Of course, the pilot also needs to check on all his indicators regularly. This is done by **visual sampling**: checking one indicator, then checking another one, and so on. But how often does the pilot check each indicator? Research has shown that this depends on the **risk** involved with not checking the indicator. This risk is the **expected cost** of not checking the indicator. Subconsciously, the pilot tries to minimize it.

Basically, this means that people create a **mental model** of the statistical properties of events in the environment. They use this model to guide their visual sampling. We can use this idea when designing the cockpit. Displays that are frequently sampled must be placed centrally and close together. Alternatively, we can also condense data in one display or minimize the data which needs to be displayed.

Examples of displays that minimize data are the **primary flight display** and the **navigation display**. They are both used by the flight director. The **flight director** uses these two displays to tell the pilot what he needs to do to keep for example the right heading. The pilot first defines the control task. (e.g. maintain heading.) The automatic controller then uses the available information to generate a control signal FD. This control signal is presented to the pilot, who then tries to follow it using his control stick. (This is called '**follow the needle**'.)

4.1.3 Having multiple displays

There are a lot of displays in the cockpit. How does the brain process this? The first phase is the **preattentive phase**. The brain then organizes the visual world into objects and groups of objects. Objects are often grouped together when they are close together, when they lie in a line/curve and/or when they look alike. In the second phase, called **focal attention**, we then limit our attention to a certain (group of) objects.

A pilot can also process multiple data next to each other. We call this **parallel processing**. Parallel processing can be achieved in multiple ways. In **multi-modality**, information is provided by multiple systems, like the visual system, the vestibular system, the auditory system and such. In **visual modality**, information comes from both the foveal and the peripheral vision. Finally, when only using the **foveal vision**, we can process several items within the useful field together.

When using multiple stimuli (like multiple displays), **emergent features** might occur. This is the case when the displays have a global property that is not evident when one looks at the displays individually. (For example, if all displays suddenly get a background color, this might seem like an interesting feature. But if these colors then happen to write a message on the complete dashboard, we have emergent features.)

4.1.4 Principles of display design

When designing the dashboard in an aircraft, it is wise to take into account several principles. We'll list a few.

- The **proximity compatibility principle** (PCP) states that, if information sources need to be integrated, it will be benificial to present the information in an integrated format. So, displaying related information close to eachother is positive.
- The **principle of pictorial realism** states that, when information is mentally represented in an analog fashion, it should also be displayed in an analog format. Also, the orientation and ordering of these displays should be compatible. (In an altimeter, high altitudes should be displayed up, not down.)
- The **principle of the moving part** states that the direction of movement of an indicator on a display should be compatible with the direction of movement of an operator's mental model of the variable. For example, it doesn't make sense if a pitch-up motion is displayed by an indicator going downward.

When setting up a display, another very important thing to decide is which reference frame to use. In an **ego-centered (ego-centric) frame of reference**, the pilot's perspective is used. (An example is the tunnel-in-the-sky display.) However, in the **world-centered (exo-centric) frame of reference**, a moving map is displayed. The camera now provides a top-down view of the aircraft. When displaying information about the motion of the aircraft, an ego-centric frame of reference is generally preferred.

4.2 Perceiving a three-dimensional situation

4.2.1 Perceiving depth

In the old conventional cockpit instruments, the information about the location and motion of the aircraft was divided over at least four separate instruments. Mentally integrating this information is difficult. A solution was offered by the **electronic flight instrument system** (EFIS). This integrated all the data in two displays: the PFD and the ND.

However, the EFIS displays also have a downside: they are planar (i.e. two-dimensional). However, the pilot needs to get a three-dimensional image of the situation. So this could be a cause of a lack of situational awareness. It would be better if a new version of the PFD presents data in a spatio-temporal way. One such example is using an **ecological display**. Such a display tries to provide visual cues which humans naturally would also use to perceive their motion in the environment.

There are several ways to add depth in an image. Examples are using tricks like linear perspective lines, light and shadow or the relative sizes of objects. If the eyes are 'fooled' in this way, we are talking about **indirect perception**. The opposite is **direct perception**. Here, the depth information is contained in the way the light enters the eyes.

4.2.2 Perceiving motion

Next to depth, we should also add motion to an image. A motion perspective causes **optical flow**. The **gradients** of this optical flow (i.e. how fast objects appear to move) determines the magnitude of the relative velocity of the observer. On the other hand, the **pattern** of the optical flow is an indication of the direction of the relative motion. (That is, if all object appear to be moving to the right, you are moving to the left. But if all objects appear to be going away from each other, you are moving towards them.)

How do we perceive motion? Let's suppose that we are flying over a ground with a lot of straight parallel lines. First, let's say we're flying parallel to these lines on a height z. The horizontal distance between our airplane and a certain line is y. The so-called **optical splay angle** Ω is the angle which this line appears to make with our longitudinal axis. (Though in reality these lines are parallel.) This angle, and the **splay angle rate** $\dot{\Omega}$, can be found using

$$\Omega = \arctan\left(\frac{y}{z}\right) \qquad \text{and} \qquad \dot{\Omega} = -\left(\frac{\dot{z}}{z}\right)\cos\Omega\sin\Omega + \left(\frac{\dot{y}}{z}\right)\cos^2\Omega. \tag{4.2.1}$$

It is interesting to note that Ω does not depend on x or $\dot{x}$ and is only scaled by z.

Now examine the case where we are flying perpendicular to all the horizontal lines on the ground. This time our distance towards a line is denoted by x. The **optical depression angle** and the **depression rate** are now given by

$$\delta = \arctan\left(\frac{z}{x}\right) \qquad \text{and} \qquad \dot{\delta} = \left(\frac{\dot{z}}{z}\right)\cos\delta\sin\delta - \left(\frac{\dot{x}}{z}\right)\sin^2\delta. \tag{4.2.2}$$

This time, δ does not depend on y or $\dot{y}$. But z still does scale δ.

When we're in a rectilinear motion, there are two important parameters which determine our perception of velocity. First, there is the **optical edge rate** (OER). This is the rate at which local discontinuities cross a fixed point of reference in the observer's view. It depends on the texture and the velocity, but not on the height above the surface. Second, there is the **global optical flow rate** (GOFR), defined as $\dot{x}/z = V/h$. It depends on the velocity and the height, but not on the texture.

Another important parameter is the **time-to-contact** (TTC) $\tau(t)$. Let's suppose that an object is a distance $Z(t)$ away, but travelling towards us with a velocity V. On our retina, this object has a size $r(t)$, but this size is growing with a velocity $v(t)$. It can now be shown that

$$\frac{Z(t)}{V} = \frac{r(t)}{v(t)} = \tau(t). \tag{4.2.3}$$

So, by using the speed with which an object grows on our retina, we can calculate the time-to-contact. This is another way of acquiring a feeling for three-dimensional depth.

5. Human performance

When we use humans to control things (either manually or as supervisor), we would like to know how well they are doing. For this, we need human performance models. In this chapter, we first examine supervisory control, and how humans are involved in it. Second, we look at the errors which humans make. Finally, we examine human thoughts; we delve into the secrets of cognition.

5.1 Supervisory control

5.1.1 What is supervisory control?

When dealing with very complicated plants, human operators cannot control everything theirselves. So instead, we let machines control everything. The humans then only act as **supervisors**, exerting supervisory control. Here, **supervisory control** is defined as giving instructions to automatic controllers, who then translate these commands to detailed actions.

When applying supervisory control, there are several steps that have to be done.

1. **Plan** – Think of what you want the machine to do.
2. **Teach** – Tell the machine what it needs to do.
3. **Monitor** – Check if the machine is doing what it needs to do.
4. **Intervene** – If the machine doesn't exactly do what you want it to do, manually change things.
5. **Learn** – When a plan doesn't fully work out, insert a better plan into the machine.

Even when you use supervisory control, humans still need to be informed of what is happening in/around the plant. This can be very complicated, as tens of thousands of variables may be involved. These data need to be displayed in a simple and intuitive way. One example in which this could be done is the **synoptic display**, which shows a schematic overview of the plant. Another type of display is the **polar star display**. Here, several parameters are displayed together in a circular fashion. The higher a parameter is, the further away it is displayed from the center.

5.1.2 Supervisory control and humans

Automation isn't always positive. There are often risks involved. When there is too much automation, humans might get unemployed, lose skills, feel unproductive and/or feel like they lost power. Next to this, for automation to work, humans must also trust automation. When they don't know what machines are doing, why they're doing it and/or how to influence them, then they will not gain trust in the automation technology. Trust can only be achieved when a human perceives competence in automation.

Let's examine a human applying supervisory control. A model describing the human performance is called a **human performance model** (HPM). One way to get such a model is by modeling the processes/physics in a human. We then wind up with an **anthropomorphic model**.

When examining a human, we also need to keep track of what this human knows and perceives. The knowledge of the human operator about a system and the task to be performed is called the **internal representation**. However, it is usually quite hard to find out exactly what the human perceives/knows. So we use a model of this internal representation. This model is called an **internal model** and is used in a human performance model.

5.2 Human errors

5.2.1 Types of errors

Making errors is human. But what exactly is an error? An **error** occurs when an action, or a series of actions, fails to achieve the desired result, and this failure cannot be attributed to some chance event.

There are two kinds of errors: slips/lapses and mistakes. A **slip/lapse** occurs when the intention of the operator is correct, but the action is wrong. (E.g. misreading an altimeter.) A **mistake** occurs when the intention is wrong. (E.g. reading an altimeter and interpreting it as the height above the ground.) While slips/lapses are generally skill-based errors, mistakes are rule-based or knowledge-based errors.

We can also make a distinction between active and latent errors. An **active error** is an error with (nearly) immediate consequences. (Think of accidentally pushing the control stick all the way forward.) On the other hand, a **latent error** is an error with delayed consequences. Only after days, weeks or years will the consequences take place. (Think of a design error which will only cause problems when the aircraft becomes a bit old and rusty.)

5.2.2 Why do errors occur?

Let's ask ourselves, why do errors occur? Slips/lapses generally occur due to **underattention**, when you accidentally apply the wrong action, or due to **overattention**, when you apply an action too fast/too eagerly. Rule-based mistakes can be caused by two things. You could apply the right rule at the wrong time/place. (E.g. when you quickly break in front of a red traffic light, not noticing that there's a huge truck closely behind you.) But rule-based mistakes can also occur due to bad rules. (E.g. when you have the habit of accelerating when you see an orange traffic light.) Finally, there are also knowledge-based mistakes. These can be caused by misinterpreted information, overconfidence, illusory correlations, etcetera.

Error shaping factors are factors that make your errors more likely or less likely. For example, the frequency of use influences the likelihood of making skill-based slips. Similarly, the mind set of a person influences the likelihood of making rule-based mistakes. Finally, the amount and the usefulness of the available information influences the likelihood of making knowledge-based mistakes.

5.2.3 Preventing and predicting errors

How can we prevent errors? Making errors is human. So you could argue to apply as much automation as possible. However, there are a few problems with this. The designer of the automation is also human. So errors may slip in there too. Next to this, there are always tasks that cannot be automated. By automating the rest of the tasks, you reduce the amount of practice which the human operator gets. This will result in a worse performance when the efforts of the human operator are required. So, when trying to reduce errors, we should always check if our actions have the desired effect.

When we cannot prevent errors, we can at least try to predict them. An example of a technique which does this is the **technique for human error rate prediction** (THERP). In a so-called **probability risk assessment** (PRA), we model the system and its failures. Using this model, we can then calculate the probability of success of a mission. Of course, the downside is that we have to make several assumptions while developing our model. Do the events that turn up in incident analyses actually cause accidents? And aren't there other unknown events which might cause accidents as well?

Once an error has occurred, you should of course try to find the cause. But more importantly, you should make sure that the error cannot be made again. A very tempting but generally bad solution is to fire the error maker. Instead, you should take the point of view of the error maker at the point where he made the error. Were his decisions necessarily wrong? Or is the error caused by some underlying anomaly? If that is the case, then you should fix the anomaly, instead of firing the error maker.

5.3 Cognition

Cognition is the scientific term for the process of thought. Classic **cognitive psychology** focusses on processes inside the head, like memory, matching, pattern recognition and such. **Ecological psychology** focusses on the environment. What happens in the environment determines for a large part the reactions of human beings. An extension to this is **naturally situated cognition**. This is the view that our cognition doesn't only depend on the environment, but also on culture, upbringing and more. So cognition is best studied in an as big as possible picture.

When studying cognition, what should we look at? There are several important fields.

- **Anthropology** – By understanding humanity across the ages, we will also be able to better understand the influences of culture and knowledge on cognition.
- **Linguistics** – By understanding the languages which humans use, and how they use them, we will know better how language influences cognition.
- **Semiotics** – This is the study of signs and symbols. It allows us to understand how meaning is constructed and understood.
- **Sociology** – How do people influence eachother? What effects does this have on how people think?
- **Philosophy** – How do people consider their own existence? And how do they consider their own knowledge and beliefs?

There is a reason why cognition is so important. We have to understand how a pilot uses his flight deck, and how it influences his thoughts and believes. Only then will we be able to really improve it.

6. Using models to improve display design

When designing human-machine interfaces, like displays, we need to have information about humans. This information can be provided by models. So in this chapter we will first look at how we can model stuff. We then examine how we can use these ideas to design displays. Finally, we examine the principle of workload.

6.1 Modeling systems

6.1.1 The abstraction hierarchy

In previous chapters, we have often made models of systems. Models are needed to reduce the complexity of the system. Reducing the complexity can be done in two ways. We can either leave out details of the system, or we can only look at a certain part of the system. For example, we can only look at the purpose of the system.

One way to model a system is by using the **abstraction hierarchy** (AH). The AH splits up the system in levels. Each level then provides the means to realise the next higher level. The AH thus provides a means-ends relationship. Generally, there are five different levels in AH.

- **Functional purpose** – This level describes the goals of the system, which usually is some desired state of the environment.
- **Abstract function** – This contains high level functions, like the physics function which describe the world. (Think of mass and energy conservation laws.)
- **Generalised function** – This level contains functions which describe the processes in the system. (Think of applied aerodynamics/thermodynamics laws.)
- **Physical function** – Here we describe the functions of the actual physical parts of the system. (I.e. what are the functions of all the system parts?)
- **Physical form** – What form do the physical parts of the system have?

To see how this works, we can consider the example of an airplane wing. This wing has a certain physical form/shape. Because it has this shape, it fulfills its physical function, which is providing lift. The lift forces are then transferred to the aircraft fuselage. (This can be seen as the generalised function.) The result is an upward momentum of the aircraft, which is the abstract function. This then achieves the actual functional purpose of the aircraft: staying up in the air. So in this way, each level provides the means for the next level.

Why would we use AH? AH can be useful to discover constraints in the functionality of a system. These constraints can be caused by applying constraints on the function (at functional purpose level), by actual physical laws (at abstract function level), etcetera. Next to this, AH also forms the basis for **ecological interface design** (EID), which will be discussed in the next part.

6.1.2 Multilevel flow modeling and functional modeling

In **multilevel flow modeling**, we consider every basic unit as a flow structure. Each structure contains a number of flow functions. We can, for example, look at mass flow structures or energy flow structures. Such structures are built up out of several basic elements.

- The places where mass/energy comes from are called **sources**. They are displayed as a circle with a dot.
- The mass/energy disappears again at **sinks**. These are displayed as a circle with a cross.
- Between a source and a sink, we can place a **transport**. This transports mass/energy. It is displayed as a diamond with an arrow.
- Instead of a transport, we can put in a **barrier**. This blocks the flow. It is displayed as a diamond with two diagonal stripes.
- A **storage** is displayed as an empty hexagon. It can contain a (limited) amount of mass/energy.
- A **balance** is displayed as an hexagon with two diagonal stripes. It merges/divides a flow: the sum of inflowing and outflowing mass/energy has to be zero.

Sources, sinks, storages and balances can only be linked to transports and barriers, and vice versa. This linking is done by drawing lines between these elements.

6.2 Ecological interface design

6.2.1 Types of machines and instruments

Through evolution, the body and senses of humans are optimized to help us survive in a natural environment. But today, we have a new environment, dominated by technology. We need to make sure that this technology uses our body and senses in an optimal way: in the way which nature used to do.

We can make a distinction between simple machines and complex machines. **Simple machines** (like phones) have a deterministic behavior. Also, if they fail, there is no catastrophy and there are clear instructions on how to deal with this failure. On the other hand, **complex machines** (like airplanes) are not fully deterministic. Using them involves risks, and when a failure occurs, the operators might not be familiar with how to deal with it. To cope with complex systems, a high degree of automation is generally applied.

We can make a distinction between rote instruments and smart instruments. **Rote instruments** consist of a large number of basic type meters. All these meters can be used to derive a large variety of properties. But this can get rather complicated. On the other hand, **smart instruments** specialize on a particular task. Instead of being troubled by constraints, smart instruments use constraints to achieve their goal.

6.2.2 User-centered design and ecological interface design

Let's try to make an interface as good as possible. A smart approach would be to provide information such that it exploits the human perception. But before we can do that, we have to ask ourselves some important questions.

- **Content** – What are the goal-relevant properties of the environment that need to be measured?
- **Structure** – How are these different properties related?
- **Form** – What visual form should these properties take?

Next to the world, we also need to examine who will use the display. A **user-centered design** (UCD) focuses on the end users, their capabilities, their specific tasks and their preferences. To find out these parameters, we can perform for example a **goal-directed task analysis** or a **competencies analysis**.

In **ecological interface design** (EID), we give priority to the work domain (the **ecology**) of the display. What is the purpose of the system in its environment? What must be done to complete this purpose? How can these (sub)tasks best be accomplished? And how can multiple parts of the system/multiple operators support each other in completing these tasks? EID is useful for complex and safety-critical work environments.

You might be wondering, when should we use UCD and when should we use EID? Basically, we should use EID when we want users to become experts, such that they can handle the unexpected. Do note that EID does not replace UCD. Instead, it complements it.

6.2.3 The TCAS system

An example where EID can be usefully applied is in the **traffic alert and collision avoidance system** (TCAS). The TCAS estimates the time τ to the **closest point of approach** (CPA) by using

$$\tau = -\frac{R}{\dot{R}}, \tag{6.2.1}$$

with R the distance towards the other aircraft. There are, however, a couple of issues with the TCAS. The τ parameter is sometimes estimated incorrectly or with a low accuracy. For this reason, pilots are often unsure whether a TCAS warning is correct. This causes the pilot responses to be knowledge-based instead of rule-based, thus increasing the workload.

A possible solution would be to extend TCAS with **automatic dependent surveillance** (ADS). ADS automatically transmits (among others) aircraft position, velocity and heading data. This reduces the inaccuracies of the TCAS system.

Next to this, the TCAS should also be accompanied by a good interface. An example of such an interface is given by the **airborne separation assurance system** (ASAS). This system displays the relative velocity vector of an aircraft with respect to another aircraft. It also displays the vector regions which will cause the aircraft to come into the protected zone of the other aircraft.

6.3 Workload

6.3.1 What is workload?

When comparing human-machine systems, there are three important parameters: **operator situation awareness**, **task performance** and **operator workload**. In this part, we'll examine the workload.

There are several definitions of workload. The **task demand load** (TDL) is the mental effort required to accomplish a certain task. The **(task) mental load** ((T)ML) is the amount of mental workload as experienced by the human operator. (While the TDL does not depend on the operator, the ML does.) Finally, there is the **willing-to-spend capacity**. This is the base level of sustainable/acceptable mental load.

There is a relationship between workload and performance. This relationship is characterized by the famous **inverted U graph**. When the workload is too low, there is **underload**: the performance is low. (Think of boredom, dissatisfaction, etcetera.) However, when the workload is too high, the performance is low as well: there is **overload**. (Think of exhaustion, time-shortage, etcetera.) This means that there is an optimal workload somewhere in between, giving the maximum performance.

6.3.2 Measuring the workload

When designing a system, it would be nice if we can determine the task demand load beforehand. One way in which this can be done is by doing a **task analysis**: how many tasks does the user have at every point in time? And how much effort do these tasks require?

The problem still remains that it is rather subjective to measure the mental load. To reduce these subjective effects as much as possible, workload assessments are subject to several requirements. A few of them are listed now.

- **Selectivity** – The assessment must be immune to other variables, like the emotional load of the test person.
- **Obtrusiveness** – The technique should not interfere with the variable to be measured.
- **Sensitivity** – The assessment should be sensitive to changes in task difficulty.
- **Reliability** – Measuring under identical circumstances should yield identical valus.
- **Consistency among subjects** – There should be little variation between different test persons.

When a user has performed a test, he/she can answer questions about the perceived workload. This kind of measure is called a **subjective measure**. An example of such a subjective measure is the **(modified) Cooper-Harper scale**.

Another subjective measure is the **NASA Task Load IndeX** (NASA TLX). The NASA TLX is a multi-dimensional rating instrument. It has six subscales, being mental demand, physical demand, temporal demand, effort, performance and frustration level. After completion of the task(s), a weight is assigned to each subscale. Subjects then have to rate each task based on the six subscales. Finally, the weights and ratings are combined, resulting in a **weighted rating** for each task. Often, final ratings are normalized, such that the vector of all ratings of a subject has a zero mean and a unity variance. This is called a **Z-score**.

Another way to assess the workload of a task is the **secondary task (dual-task) measure**. Here, human subjects have to do two tasks; one of which is the primary task. When the mental load required for the primary task increases, the performance of the secondary task will decrease. So the performance of the secondary task is a measure of the workload of the primary task.

Workload can also be assessed using **physiological measurements**. We could, for example, look at the pupil diameter of the test person, the heart-rate variability, the evoked brain potentials, the muscle tension or the skin respiration. The downside is that these parameters differ significantly per test person and are also easily influenced by other circumstances.