Chapter 3

Control engineering – the heavy machinery of feedback

Heikki Hyötyniemi
Helsinki University of Technology, Control Engineering Laboratory

Control engineering makes the idea of feedback concrete. The system behavior is measured, and if there is difference between the real and intended behaviors, appropriate control actions are applied. However, it is not always clear what kind of control happens to be appropriate – understanding these issues leads to deep theoretical questions.

3.1 Control as Common Sense

For a layman, self-regulating systems are something "intelligent". Controlling one's behavior in a reasonable way is one of the intuitive definitions of intelligence.

The goal of *control engineering* is to make systems – *any* system, really – behave somehow better than it would naturally do. The basic tool to accomplish this is *feedback*. Being an engineering discipline, control engineering exploits this feedback idea to the extreme – this chapter is a *tour de force*, trying to visualize the power of the basic ideas, and illustrating the ever-increasing challenges that are faced when more and more complicated systems are being controlled. More detailed discussion can be found in [3.2].

Control engineering makes the idea of feedback very concrete: How to learn from the system behavior and how to use this information for reaching better behavior. In a way, control engineering is "formalized common sense", a logical way of learning of experiences. To reach the intelligent-looking behavior of the controlled system, somebody needs to implement his/her knowledge in the controller. That is why, control engineering work is very knowledge-intensive: One has to understand the natural behavior of the system, and, after that, one has to be able to implement the appropriate control actions.

3.1.1 Abstractions

When speaking of feedback, it is some kind of *information* that is received and utilized. And speaking of information processing refers to something that is not so concrete; things have to be studied on a higher level of abstraction. This is the case also in control engineering – meaning that, rather than studying the physical structure, the functional or logical structure of the system is concentrated on. One of the most fundamental functional features of a system is its *causal dependency structure*.

To illustrate this structuring problem, study an example of determining the causalities – what are the *inputs* and what are *outputs* of a system (see Fig. 3.1). There is a tank whose level can be controlled by affecting the amount of outflowing liquid from the system using a valve in the outlet; incoming flow is assumed to be a disturbance. At first, it would be tempting to claim that the incoming flow $Q_{in}(t)$ would be the process input and the outgoing flow $Q_{out}(t)$ would be output. Of course, this is true on the *physical* level. But in control systems, we forget about the physical flows and concentrate on information flows: How the information can be extracted from the system and how this information can be used to change its behavior. In this sense, it is the tank level h that is being measured that serves as the source of information concerning the system state. And it is the *output flow* that can be controlled, thus changing the state of the tank system (in instrumentation diagrams, "FC" stands for *flow control* and "LI" for *level indication*).

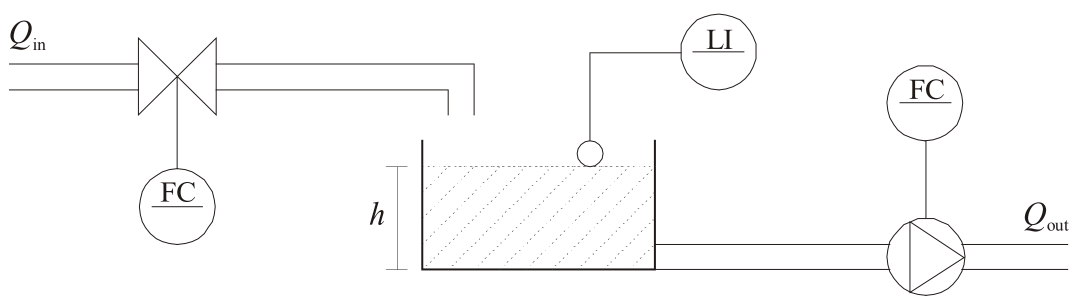

Figure 3.1. What are the inputs and outputs of the tank system?

The first step in control engineering work is to *abstract away* details that are not relevant. In its most simplified form, the model of the system is presented as a "black box", where only the input and output signals are explicitly shown (see Fig. 3.2). It needs to be noted that this kind of models are not unique: Concentrating on different things, different models can be derived.

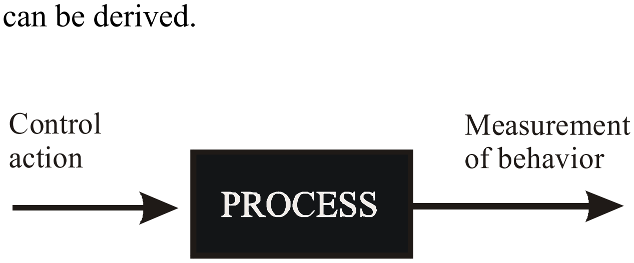

Figure 3.2. Process seen as a "black box" between input and output

3.1.2 Control strategies

Now, using the input/output process model, we can proceed towards *controlling* the plant. The idea of *negative feedback* means that if the measurement tells that the quantity to be controlled is too high, less input is fed into the system, and vice versa. To express this more explicitly, we now define the outlook of the *feedback control system* (see Fig. 3.3).

This structure is quite general, and it is applied whenever feedback control is utilized. The control signal $u(t)$ is supplied by the controller; the output of the process, the signal $y(t)$, is measured, and this real behavior is compared against the intended behavior, the reference signal $r(t)$. The difference between these two values, the error signal $e(t)$ is calculated as

$$e(t) = r(t) - y(t)\,. \tag{3.1}$$

This error signal is used for determining the appropriate new control action $u(t)$. The control signal can generally be any function of $e(t)$. How this function should be selected – this is a delicate question, and it is this selection where the control engineer's expertise is tested. In what follows, different cases are studied.

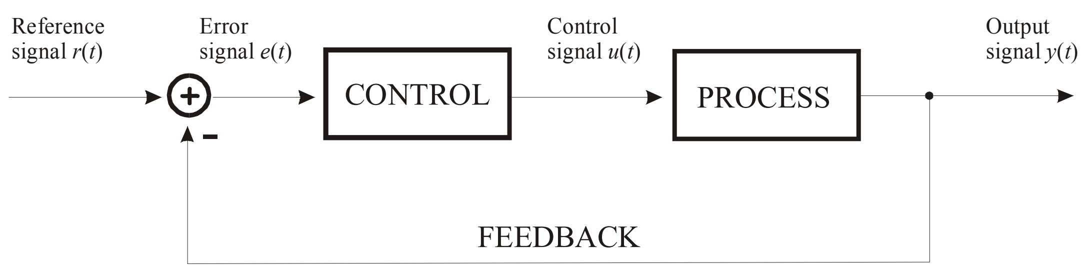

Figure 3.3. The general structure of the feedback control

First, assume that the tank system of Fig. 3.1 is to be controlled, so that we want the system output, or the tank level to follow our reference, when the control signal to be manipulated is the outgoing flow (that is, $y(t) = h(t)$ and $u(t) = Q_{\mathrm{out}}(t)$). The first control strategy that immediately comes to mind is to increase the total incoming flow if the difference between the reference value and actual tank level is positive; written in symbol form, we have the *proportional control law* (or *P control* for short):

$$u(t) = K_{\mathrm{P}} \cdot e(t)\,, \tag{3.2}$$

where K_{P} is a constant. Whatever value this calculation gives out, the outgoing flow is manipulated correspondingly. It turns out that this P control strategy works just as we imagined (see Fig. 3.4): The nearer the tank level is to the reference, the more cautious the control becomes, resulting in nice, smooth behavior. The response becomes faster and faster when the parameter K_{P} is increased.

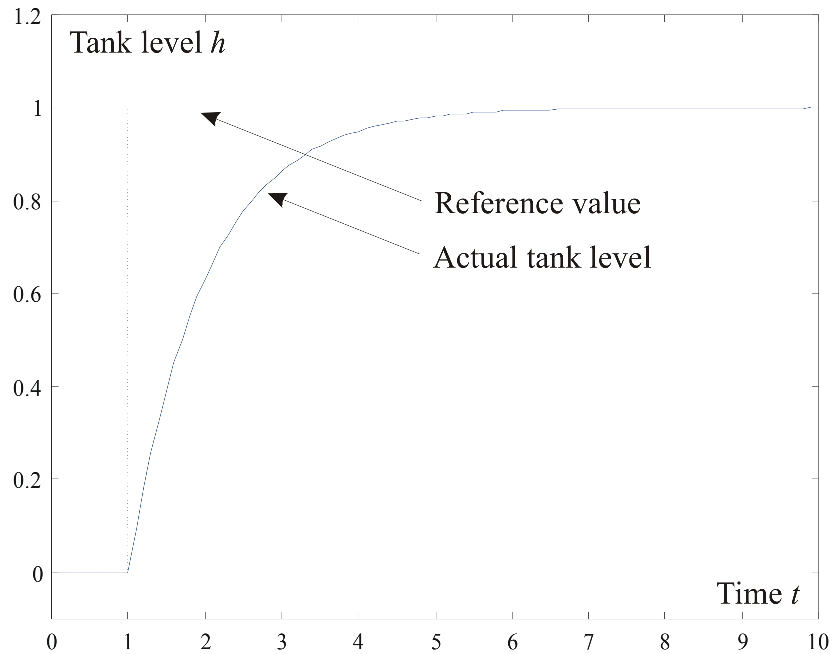

Figure 3.4. Tank level control

As compared to the above tank level control problem, a very similar-sounding process is *radiator control*: If the room temperature is too low, add heating power, and vice versa. However, it turns out that the above P control strategy no more works in this case (see Fig. 3.5): There will remain a steady-state error between the reference value and the actual temperature, no matter what is the value of the parameter K_P. To understand this phenomenon, one needs to study the basic difference between these processes – whereas the tank behaves as an *integrator*, the radiator does not.

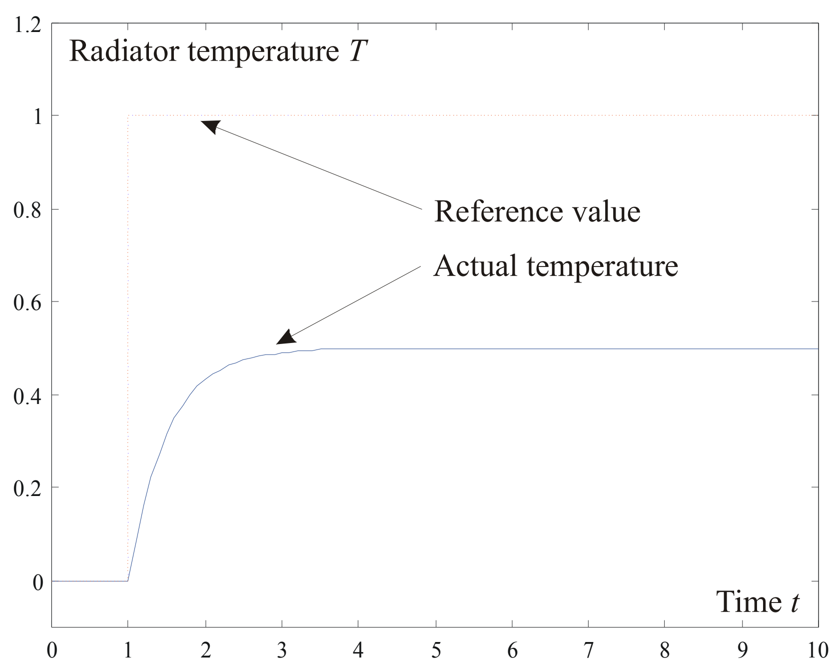

Figure 3.5. Simple radiator control *fails*!

From the control point of view, the role of the integrator is to "remember" past behaviors; if it turns out that the control action cannot reach the reference value, the control signal cumulates; finally, the cumulated control necessarily overcomes all sluggishness. On the other hand, if there is no integration in the process to begin with, one can compensate for this problem by adding integration explicitly in the control loop. And, really, it turns out that this *integrative control law* (or *I control* for short) accomplishes the control task, the integral of the error signal being used for control (now, the tuning parameter is K_I):

$$u(t) = K_\mathrm{I} \cdot \int e(\tau)\, d\tau \,. \tag{3.3}$$

Next, assume that a *frictionless mass point* is to be controlled. Surprisingly, if controlled using the P controller, harmonic oscillation results (Fig. 3.6) – the controller behaves like a string, pulling the mass point always towards the reference location, but never being able to freeze the movement. Increasing K_P only results in faster oscillations! On the other hand, I control (or any combination of P and I) is this time no solution: The system behavior becomes unstable (see Fig. 3.7). The reason for this is that the mass point is a *double integrator*, and all control actions cumulate in the system – yet another integrator can only make things worse. Intuitively, one is tempted to try an opposite strategy: Eliminate the excessive integration introducing *derivative* action in the controller. Indeed, the *derivative control law* (or *D control*) becomes

$$u(t) = K_\mathrm{D} \cdot \frac{d}{dt} e(t) \,. \tag{3.4}$$

Finally, as shown in Fig. 3.8, stabilizing control is reached. Note that D control alone cannot drive the system to the reference value, so that a combination of P and D is needed.

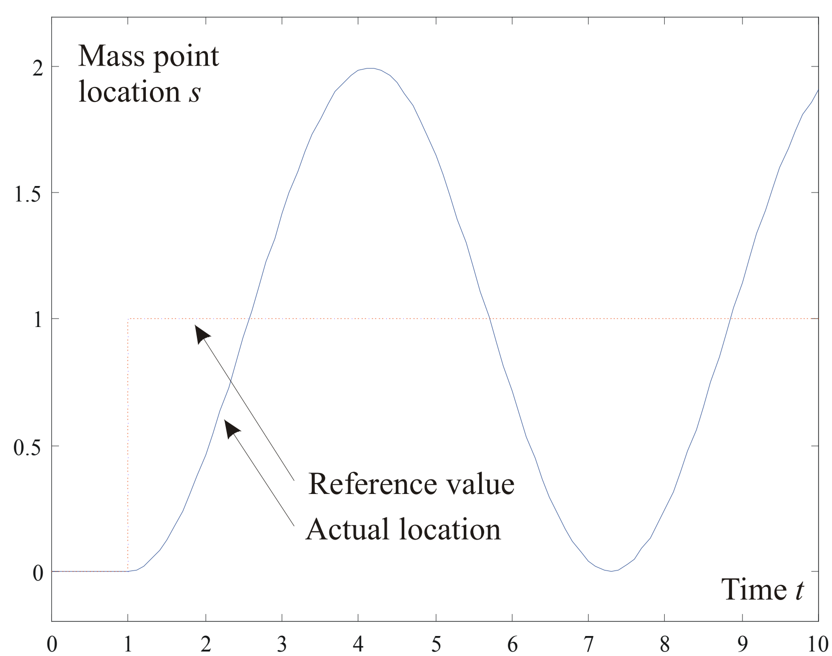

Figure 3.6. Mass point control *fails*!

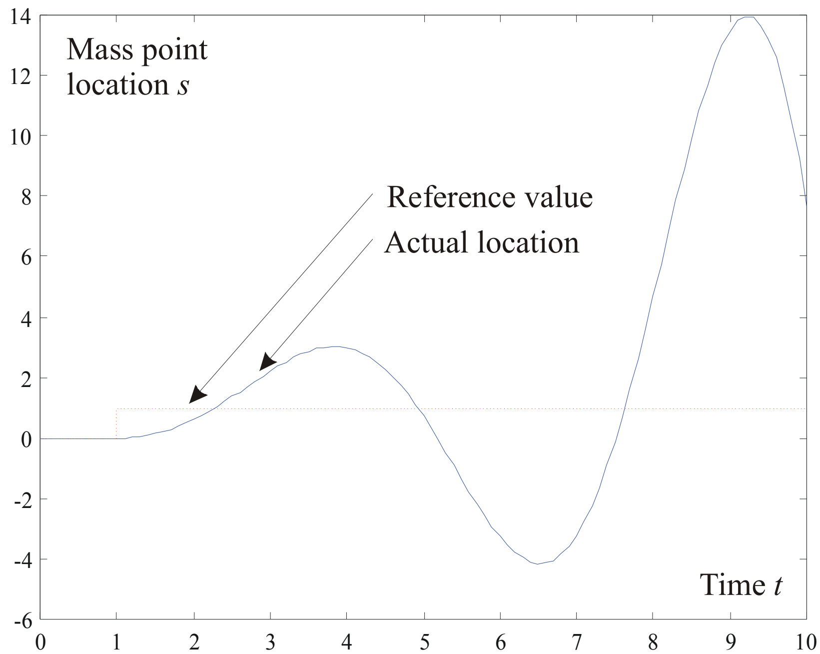

Figure 3.7.
Failure again!

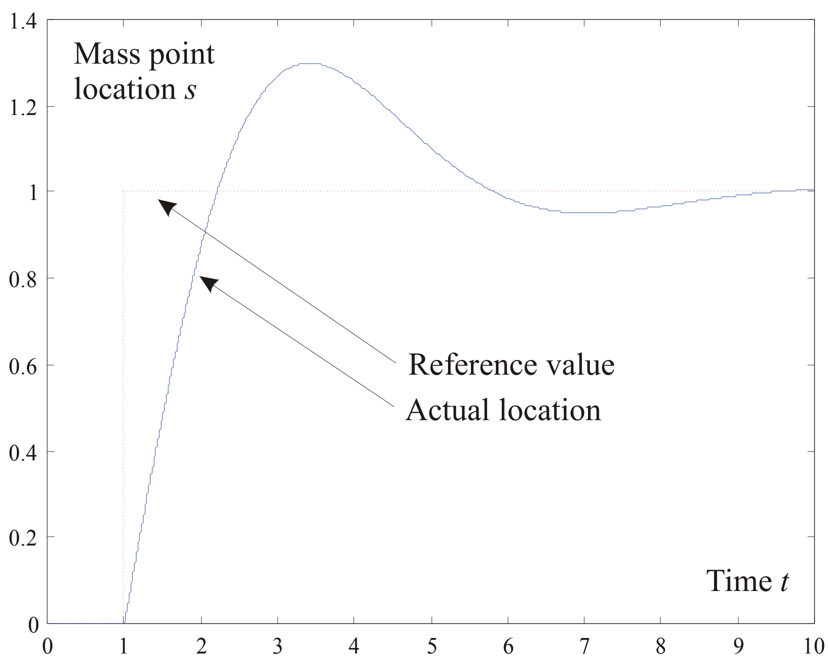

Figure 3.8.
PD control is appropriate for the mass point

Generally, all three control strategies can be combined, and the *PID controller* can be written in the form

$$u(t) = K_\mathrm{P} \cdot e(t) + K_\mathrm{I} \cdot \int e(\tau)\, d\tau + K_\mathrm{D} \cdot \frac{d}{dt} e(t), \tag{3.5}$$

where the first term (P) pushes the system towards the reference value, the second term (I) eliminates steady-state errors, and the third term (D) tries to make the controller fast, reacting to sudden changes. This PID strategy is the basic solution to industrial problems.

3.1.3 Limits of intuition

Human expertise is needed in selecting the controller structures and in tuning the parameters – this expertise is then utilized during run-time, making the controller itself intelligent looking. However, heuristic trial-and-error approaches do not always work, specially when the processes to be controlled are large-scale ones. For example, take the *inverted pendulum* (see Fig. 3.9): No PID controller alone can stabilize it, no matter how well the parameters are tuned.

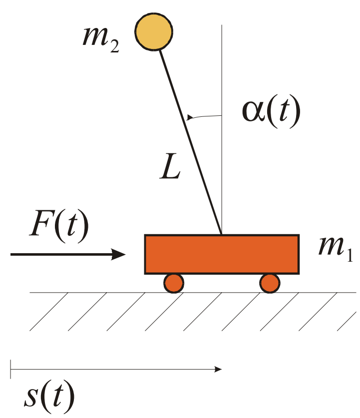

Figure 3.9. Inverted pendulum. The unstable stick should be kept in upright position applying external force *F*. The classical control approaches cannot solve this stabilization problem

Dynamic systems offer a plenty of surprises. As there are new phenomena to be compensated, the dynamic behavior of the controller itself becomes more and more complicated, and sooner or later the overall complexity becomes overwhelming. Humans are notoriously bad at mastering time-dependent, slow phenomena. It is our luck that natural processes are usually inherently stable, and lousy controls cannot make too much harm – JAS Gripen aircraft and Chernobyl power plant being famous exceptions!

Because of these difficulties, the traditional approach to controlling of large-scale systems has been to divide them into separate subsystems, and construct controllers for each of these subsystems separately. But, of course, this subsystems-oriented approach results in suboptimal behaviors. When the industrial production processes should be optimized, the intuitive approaches are no more sufficient, and analytic tools are needed to tackle with the complexity.

3.2 Control as Common Framework

Control theory has a long history – from the heuristic experiments (Watt's governor, etc.) through the classical era of compensators and single-input single-output approaches to the modern times. In the 60's, the foundations of the modern control theory were derived. The modern control systems analysis and design is based on the very solid and elegant theory of *state-space systems*. The idea is to assume a more structured view of the process – not only the inputs and outputs, but also its *states* are of interest. The system state (together with the future inputs) uniquely determines the behavior of the system in the future.

3.2.1 State-space models

The linear continuous-time state-space model can be expressed in the form

$$\frac{dx}{dt}(t) = A \cdot x(t) + B \cdot u(t)\,, \tag{3.6}$$

where $x(t)$ is the system state and $u(t)$ is its input; the state-space model also determines how the system state will change in the given circumstances. This system representation is more powerful than it first looks like – truly, the structural complexity of a system is now changed into a form of high dimensionality: Note that x and u are both vectors, containing possibly various individual variables, and A and B are matrices (matrix calculus is not repeated here; for example, see [3.1]).

Let us study some examples – first take the *mass point*. For a frictionless mass point the Newton's second law holds: The acceleration (the second derivative of the mass point location s) is proportional to the driving force F; the heavier the mass point is (mass m being bigger), the lower is the change in velocity. This holds for all time instants t:

$$\frac{d^2 s}{dt^2}(t) = \frac{1}{m} \cdot F(t)\,. \tag{3.7}$$

This is not in the standard form of equation (3.6) – but note that the second derivative can be divided in two parts, letting $v(t)$ stand for the velocity of the mass point:

$$\begin{cases} \dfrac{dv}{dt}(t) = \dfrac{1}{m} \cdot F(t) \\ \dfrac{ds}{dt}(t) = v(t) \end{cases} \tag{3.8}$$

Now, defining the system state and input vectors, respectively, as

$$x(t) = \begin{pmatrix} v(t) \\ s(t) \end{pmatrix} \qquad \text{and} \qquad u(t) = \big(F(t)\big)\,, \tag{3.9}$$

it is evident that this model can be written in the form (3.6) when one chooses

$$A = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} \qquad \text{and} \qquad B = \begin{pmatrix} 1/m \\ 0 \end{pmatrix}. \tag{3.10}$$

Next, study the *inverted pendulum*. It turns out that the (linearized) differential equations governing its behavior are

$$\begin{cases} \dfrac{d^2 s}{dt^2}(t) = \dfrac{m_2 g}{m_1} \cdot \alpha(t) + \dfrac{1}{m_1} \cdot F(t) \\ \dfrac{d^2 \alpha}{dt^2}(t) = \dfrac{(m_1 + m_2)\, g}{L m_1} \cdot \alpha(t) + \dfrac{1}{L m_1} \cdot F(t). \end{cases} \tag{3.11}$$

For explanations of the symbols, see Fig. 3.9 (the constant g = 9.81 m/s^2 is the acceleration of gravity). It is assumed that the system again is frictionless, and the stick angle is rather small to justify the linearized model. This time, we have to introduce *two* additional state variables, the trolley speed and the angular velocity of the stick, respectively:

$$\begin{cases} v(t) = \dfrac{ds}{dt}(t) \\ \omega(t) = \dfrac{d\alpha}{dt}(t), \end{cases} \tag{3.12}$$

so that the augmented differential equation model (only containing first derivatives) now reads

$$\begin{cases} \dfrac{dv}{dt}(t) = \dfrac{m_2 g}{m_1} \cdot \alpha(t) + \dfrac{1}{m_1} \cdot F(t) \\ \dfrac{ds}{dt}(t) = v(t) \\ \dfrac{d\omega}{dt}(t) = \dfrac{\left(m_1 + m_2\right) g}{L m_1} \cdot \alpha(t) + \dfrac{1}{L m_1} \cdot F(t) \\ \dfrac{d\alpha}{dt}(t) = \omega(t). \end{cases} \tag{3.13}$$

Defining the four-dimensional state vector and the one-dimensional input,

$$x(t) = \begin{pmatrix} v(t) \\ s(t) \\ \omega(t) \\ \alpha(t) \end{pmatrix} \qquad \text{and} \qquad u(t) = \left(F(t)\right), \tag{3.14}$$

the model can be written in the form (3.6) if the system matrices are defined as

$$A = \begin{pmatrix} 0 & 0 & 0 & \dfrac{m_2 g}{m_1} \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & \dfrac{(m_1 + m_2) g}{L m_1} \\ 0 & 0 & 1 & 0 \end{pmatrix} \quad \text{and} \quad B = \begin{pmatrix} \dfrac{1}{m_1} \\ 0 \\ \dfrac{1}{L m_1} \\ 0 \end{pmatrix}. \tag{3.15}$$

3.2.2 Analysis tools

The reason to see the above trouble is that for models that are written in the standard form (3.6), efficient analysis and control design methods are readily available. All analysis methods visualize the time-dependence of the system variables in different ways, making the otherwise complex phenomena comprehensible. The analysis methods constitute a beautiful construction where mathematics (mathematical analysis, function theory, linear algebra, etc.) go in parallel with physical, real-life phenomena – time-dependent transient behaviors, oscillations, and stability issues. One of the important questions to be analyzed – and a question that has a very compact solution in the state space systems framework – is whether a system can be controlled at all (see Fig. 3.10).

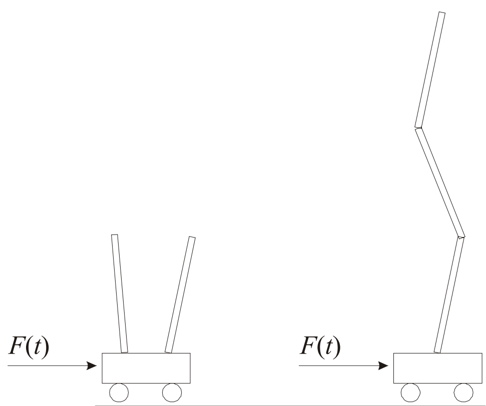

Figure 3.10. Controllability surprises. The trolley with two sticks is controllable *only if* the sticks are *not* identical. On the other hand, there can be any number of sticks on top of each other, and the system can still be controlled!

See the construction in Fig. 3.10 on the right – how *possibly* can such a system be controlled? To see a glimpse of the complete answer, let us study the stabilization of the "simple" case, stabilization of the inverted pendulum in Fig. 3.9 using the modern tools.

3.2.3 Example: Optimal control of the inverted pendulum

One of the most powerful conceptual and practical tools derived in the field of modern state-space theory is *optimal control* (for example, see [3.3]). There is a mechanical procedure for constructing a state-feedback controller that achieves *optimal* system behavior. The basic state feedback law looks like

$$u(t) = -K \cdot x(t)\,, \tag{3.16}$$

that is, the control signal is a linear weighted combination of the system states; here we assume that the state is directly measurable. To make expressions simpler, it is here assumed that the reference is in the state-space origin; all state variables should be driven to zero.

So, it is claimed that such a matrix K can be found that the system behavior is optimal. What does this optimality actually mean? Here we only study the simplified infinite end-time problem, the system starting from some state $x(0)$; in this case the cost criterion can be written as

$$J(u)=\int_0^{\infty}\left(x^T(t)\cdot Q\cdot x(t)+u^T(t)\cdot R\cdot u(t)\right)dt\,. \qquad (3.17)$$

This cost criterion consists of an integral of two quadratic forms: Essentially, within the integral, there is a scalar sum of weighted state and control variable squares. Here Q and R are some weighting matrices, so that different state components and control signals can be arbitrarily weighted (that is, deviations from the goal in some of the variable are regarded as more critical than in other variables). It is intuitively clear that if one manages to minimize the criterion (3.17), some kind of "nice" transient behavior should result.

And, indeed, under rather loose restrictions (system controllability, etc.) the criterion (3.17) really can be minimized. Without going into details, it turns out that the solution is based on the matrix S fulfilling the so called *Riccati equation*:

$$A^TS+SA-SBR^{-1}B^TS+Q=0\,. \qquad (3.18)$$

After this has been solved for S, the optimal state feedback for the system can be calculated – quite automatically – as

$$K=R^{-1}B^TS\,. \qquad (3.19)$$

Now, assume that in the model (3.15) the parameters have the following values:

- the length of the stick is $L = 1$ (m),
- the mass of the trolley is $m_1 = 1$ (kg), and
- the mass at the end of the stick is $m_2 = 1$ (kg).

Before the state feedback law can be solved, the matrices Q and R must be determined. Often one selects (for simplicity) diagonal weighting matrices, and, if all state variables and the control signal are weighted equally, one can write

$$Q=\begin{pmatrix}1&0&0&0\\0&1&0&0\\0&0&1&0\\0&0&0&1\end{pmatrix}\quad\text{and}\quad R=\left(1\right). \qquad (3.20)$$

When (3.18) is solved, one can apply (3.19), and the optimal state feedback is given as

$$K=\begin{pmatrix}-2.7 & -1.0 & 12.8 & 51.1\end{pmatrix}, \qquad (3.21)$$

meaning that each state variable (velocity, location, angular velocity, and angle, respectively) has its own contribution in the control action: If there is deviation in the trolley location (state variable number two), for example, the numeric amount of deviation is weighted by –1.0 when the control signal is calculated, so that the trolley will be pushed. Reflecting the unstable nature of the system, the variables having something to do with the stick behavior (the last two ones) seem to be rather heavily weighted, trolley position not being regarded as important. In Fig. 3.11, the system behavior is shown when the trolley starts one meter from the optimum (that is, $x_2(0) = 1$), all other state variables having zero initial values. The system stabilizes quite smoothly. In Fig. 3.12, on the other hand, it is shown how the results are changed if one selects

$$R = (10), \tag{3.22}$$

that is, the control signal is weighted relatively more. It is understandable that when less effort is now used, the responses become slower – but the result is again optimal.

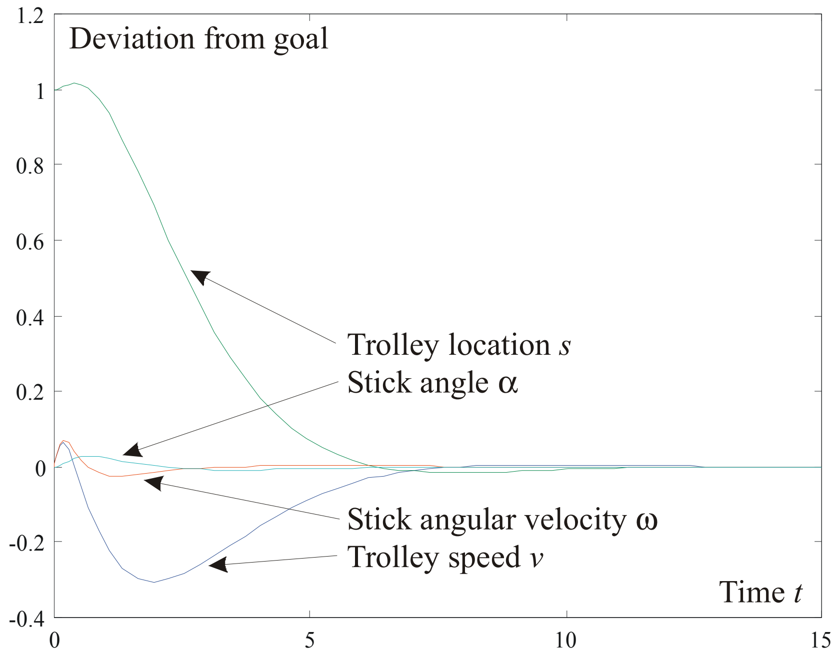

Figure 3.11. Optimal control of the inverted pendulum

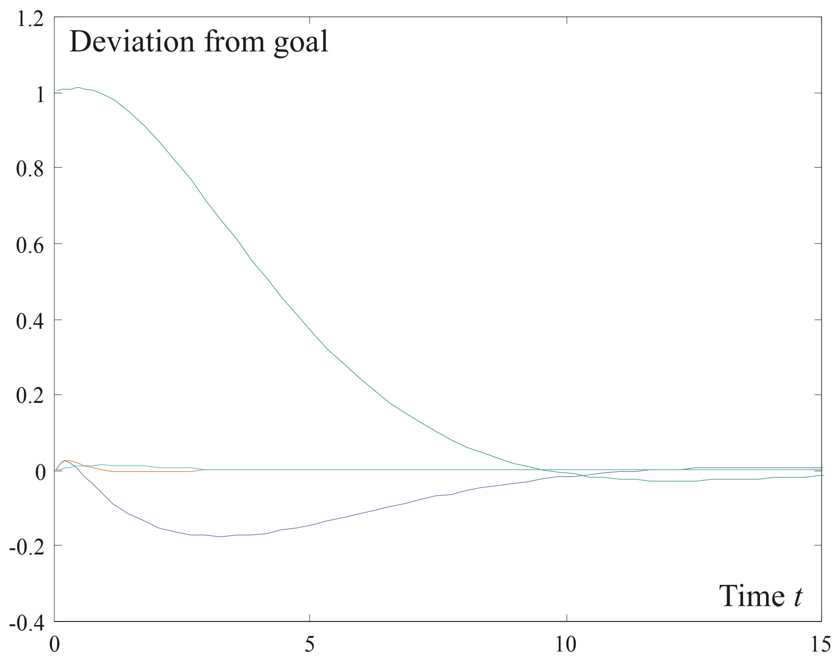

Figure 3.12. Control signal being weighted excessively

3.2.4 Further developments

The behavior of the optimal controller truly looks intelligent: For example, study the controller actions at time $t = 0$. The trolley starts seemingly in the wrong direction, away from the goal – but this is necessary to make the stick lean back. All this intelligence is achieved by the "trivial" linear feedback law. The Riccati equation seems to have some mystical powers – just as the celebrated Einstein's formula!

However, this is not yet the end of the story. Optimal results are found only if the model is linear and exactly known – and this assumption seldom holds when complicated plants are being controlled. This is where the mainstream research work in the field of control theory has been concentrating on after optimal control theory was completed. There are new paradigms for tackling the problems in different ways: For example, *robust control* searches for controllers that would not be too sensitive to model uncertainties; *adaptive control* tries to change the controller behavior according to observed nonoptimalities in the behavior, thus defining yet another level of feedback in the controller structure (see Fig. 3.13). However, only parameter values are changed in standard adaptive control – intelligent behavior assumedly is capable of changing *structures*, too. Interesting developments also in this direction are taking place today [3.5].

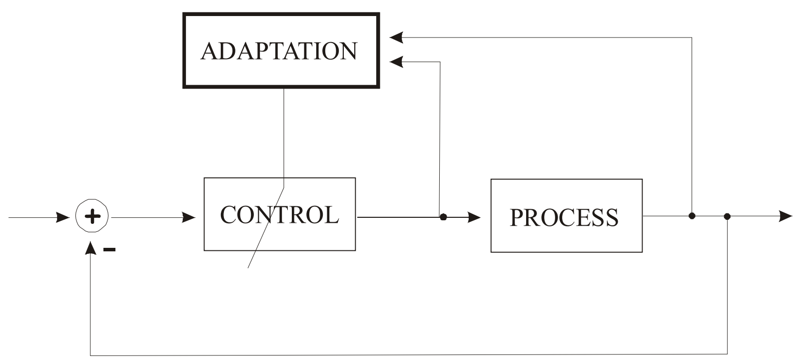

Figure 3.13.
Yet another level of feedback: Adaptive control

However, the feedback phenomena are visible in all levels of control engineering work – there is dynamics and inertia also in the control community itself. Control engineering started from intuitive, heuristic experiments and proceeded logically towards more theoretically oriented analyses – but this is only the academic view. In practice, on the factory floor level, there is the burden of old practices. New sophisticated approaches are not yet, after thirty years, generally adopted – it is PID that still rules. It seems that after the modern era there is now the period of "postmodern" methods, again emphasizing intuition and heuristic approaches instead of sophisticated theories. It is claimed that control structures that are based on *neural networks* or *fuzzy systems* are more practical than the other methods because no explicit process model is needed when using them. It is directly the operator's knowledge that is implemented as linguistic rules in the fuzzy controller – but analysis of such "intelligent controllers" is just as difficult as understanding intelligent behavior in general! Heated discussion is going on about the merits and dismerits of different approaches.

3.3 Summary: Effects of Feedback

As a conclusion, let us study what are the effects of feedback in the dynamic system. First, take the advantages:

- When using feedback, no exact model is needed; as long as the causalities are qualitatively correct, negative feedback tends to eliminate the errors. Feedback control is the *only* strategy to stabilize an originally unstable system.
- Using feedback, non-idealities can be virtually eliminated. For example, feedback is capable of compensating system nonlinearities (as seen from outside). Further, “ideal elements” like big capacitors can be constructed in electric circuits using operational amplifiers in a feedback loop (see [3.4]).

But, in addition to the above benefits, feedback causes also problems – or let us call them *challenges*:

- Feedback opens the Pandora’s box: The latent dynamics of the system are let free. For example, study a sequence of tanks. This system is always stable, no matter how many tanks are connected in series; but if a feedback loop is added, a system of three or more tanks may start oscillating and become unstable. And minor design errors (like implementing *positive* instead of negative feedback) in the feedback loop may ruin the system behavior altogether.
- Feedback loops make analysis difficult. They weaken the information content in closed loops, possibly hiding parameters. What is more, the causalities are blurred: The system output signal becomes the input through the feedback.
- Feedback changes static systems into dynamic, finite responses to infinite ones, thus making the system behavior more difficult to grasp.

Despite the risks, there are today no alternatives to sophisticated control systems – this “F-word” is here to stay. It is either *feedback* or it is *drawback*!

References

[3.1] Ayres, F.: *Theory and Problems of Matrices.* Schaum Publishing Company, New York, 1962.

[3.2] El-Hawary, M.E.: *Control System Engineering.* Reston Publishing Company, Reston, Virginia, 1984.

[3.3] Lewis, F.L. and Syrmos, V.L.: *Optimal Control* (2nd edition). John Wiley & Sons, New York, 1995.

[3.4] Millman, J.: *Microelectronics: Digital and Analog Circuits and Systems.* McGraw-Hill International, Tokyo, 1979.

[3.5] Nissinen, A.S.: *Neural and Evolutionary Computing in Modeling of Complex Systems*. Helsinki University of Technology, Control Engineering Laboratory, Report 118, 1999.