Feedback Systems

An Introduction for Scientists and Engineers
SECOND EDITION

Karl Johan Åström
Richard M. Murray

Version v3.0g (1 Nov 2015)

PRINCETON UNIVERSITY PRESS
PRINCETON AND OXFORD

Chapter Fourteen
Architecture and Design

> *A doctor can bury his mistakes, but an architect can only advise his clients to plant vines.*
> Frank Lloyd Wright

In this chapter we will put the simple feedback loops into the context of real control systems systems. We will illustrate what control systems look like and how they are designed. Important issues that have to be considered are: architecture, combination of “continuous time feedback” with “discrete elements” and design of safe systems. Architecture tells how sensors, actuators and computer algorithms are interconnected it is represented by schematic diagrams and block diagrams. Real control systems combine “continuous time feedback” with logic and finite state machines and they interact with humans. Design of safe systems relates to requirements, modeling, design, verification, implementation commissioning, operation and upgrading (in short systems engineering), it aims at design procedures that guarantee safe operation. **Think about different names and how to squeeze this into 25 pages.**

14.1 Introduction

Pictures, a good idea to develop a good set of pictures.

- Fig. 7.10, Fig 1.3 (We should perhaps elaborate on this one already on page 4 to mentionplant managers and reporting)
- Two pictures from Richards Board.
- Chapter 9 in Mc Ruer,Ashkenas and Graham and Fig 9-2 in that books

So far this book has dealt with relatively simple feedback systems, we will now give a glimpse of how they appear as components of real control systems. All control systems have sensors, actuators, communication, computers and operator interfaces, but they can have dramatically different sizes and shapes and very different user communities. It is somewhat surprising that such a variety of systems can be analyzed and designed using the same framwork. Two figures, one gives the broad picture, the other shows the details of the control system.

Figure 14.1 shows a schematic block diagram of a system. The figures shows that the system interacts with a physical environment and an operational environment. The physical environment generates disturbances and faults on the system. The operational environment are operators, pilots and managers who interact with

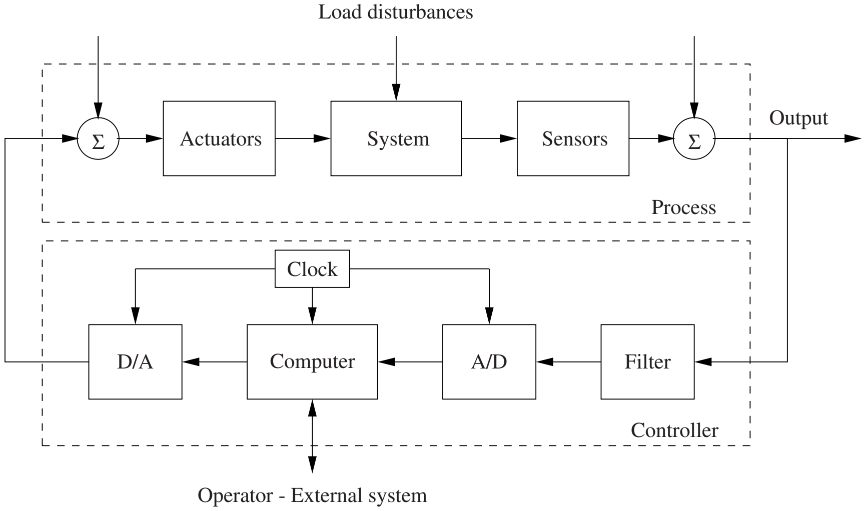


Figure 14.1: Schematic diagram of a control system with sensors, actuators, communications, comuter and interfaces.

the system in different ways, by giving commands to start or stop a system to change operationg conditions. The control system interacts with the operational environment, it observes the process by sensors and it interacts with the process through actuators. The control system has facilities for trajectory generation, filters, predictors and estimators, control algorithms but also discrete components like logic and finite state machines. Logic can be used for equipment protection and finite state machines are used for startup, shut down and other mode switches.

The control system is often built hierachically, an example is given in Figure **??**. The lowest layer consists of the process the sensors and the acturators. The second layer has filters for the sensor signals and simple feedback controllers. The third layer consists of state estimators that are making more sophisticated signal proceeing and more advanced controllers based on state feedback and model predictive control. The higher level functions are often based on mathematical models of different complexity. The system also has logic for equipment protection, it ensures that the process does not enter dangerous operating conditions. The finite state machines govern the overall operation of the system for example startup and shutdown. It also enables different alarms and different ways for operator interaction.

The complexity of real control systems can vary significantly, the cruise control system for a car is a simple system with one actuator and X sensors, the climate control system for an aircraft is a system of moderate complexity, while the control system for a large chemical process can have thousands of signals and actuators.

Figure 14.2: Control system with an hierarchical structure.

Process control and aerospace were applications where complex control systems emerged at an early stage. In process control it was customary to have one cabinet with analog controller for regulation and a relay cabinet with relay logic for startup, shut down and equipment protection. Valves are commonly used for actuation in process control. It is customary to have a feedback loop with a valve positioner to reduce effects of friction and nonlinearities at the lowest level of the hierarchy, and feedback loops for control of pressure and temperature at the next level. As technology developed the relays were replaced by programmable logic controller PLC and the analog controllers were replaced by distributed control systems DCS. The PLC's and DCS's had very different architecture, but since both were digital devices it was natural that controller functions were introduced in PLC's and logic functions in DCS systems. In process control the SCADA systems appeared ad the standard solution.

In flight control: stabilization, attitude hold, navigation and automatic landing (elaborate I have good pictures somewhere)

14.2 Design of Control Systems

Control system design includes many activities starting with requirements and modeling and ending with implementation, testing, commissioning, operation and upgrading. In between are the important steps of modeling, architecture selection, analysis, design and simulation. The procedure can be illustrated by the so-called Design V in Figure 14.3, which dates back to NASA's Apollo program. The left side of the figure shows how design moves from understanding of the process and the requirements to a simulation of the complete system. The right hand side shows how the system is implemented, tested, commisioned, operated and upgraded. Testing is only indicated at one stage of the process but it is good practice to perform it after each step in the design, this is particularly important when the design is modified. On the right leg of the V we have also shown hardware-in-the-loop (HIL) simulation, which is a simulation where some subsystems are actual hardware and other are simulated. Design always starts by developing an understanding of the system and its environment. This includes sensors, actuators, static and dynamic process characteristics, limitations on signals, bounds for safe operation and characterization of disturbances.

Even if we are focusing on control system design it it obvious that there is an interaction between process a and control design. Early analysis can reveal that there are fundamental limitations such as time delays, right half plane poles and zeros or insufficient actuation authority. Poles are inherent to the system and can only be modified by significant process modifications. The zeros can be influenced by moving or adding sensors. A nice example of dealing with insufficient actuator authority is presented in []. It was attempted to reduce risk for rotating stall in a jet engine by feedback but actuators with the required bandwidth were not available. Analysis showed that the problem could instead be alleviated by modifying the process by introducing small asymmetries.

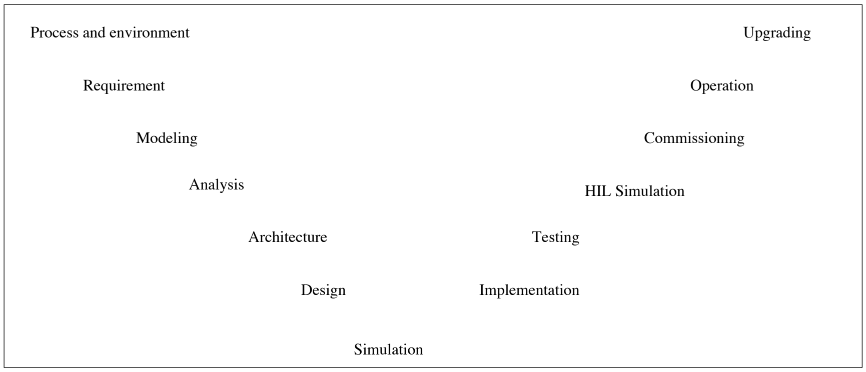


Figure 14.3: Design V Development Process. This is just a placeholder, should be redrawn to look nicer when we have agreed on labels.

The requirements are given by large and small signal behavior of the closed loop system. Large signal behavior is characterized by limitations in actuation power and its rate, small signal behavior is typically caused by measurement noise, friction, and resolution of AD and DA converters. Requirements for control systems typically include the ability to deal with disturbances, robustness to process variations and uncertatinty and requirements on command signal following. These can all be captured by linear models and they can be expressed in terms of properties of the Gangs of Four and Seven.

Refering to the block diagram in Figure 14.4 load disturbance attenuation can be characterized by the transfer function G_{yv} from load disturbance v to process output y. Measurement noise n generates undesired control actions, the effect can be captured by transfer function G_{un} from measurement noise n to control action u. Robustness to parameter variations and process uncertainty can be captured by the sensitivity functions S and T. Command signal response can be shaped independently of response to disturbances and robustness for systems with two degrees of freedom. It is characerized by the transfer functions FT and CSF. For systems with error feedback the response to command signals is characterized by the complementary transfer function T.

For specific systems load disturbances and measurement noise may enter in places that are different from Figure 14.4 and the transfer functions should then be modified accordingly. **Add references to appropriate sections**

Since many requirements are expressed in terms of properties of transfer functions of the Gang of Seven, see Section XXX, it is important to measure these transfer functions on simulated models and on real equipment. To do this the system must be provided with test points for injecting and measure signals, see Figure 14.4. The transfer function G_{yv}, which characterizes response to load disturbances, can be found by injecting a signal at w_1 and measuring the output s_{21}. Chirp signals are convenient for measuring frequency responses.

Models of the process and its environment can be obtained from physics or

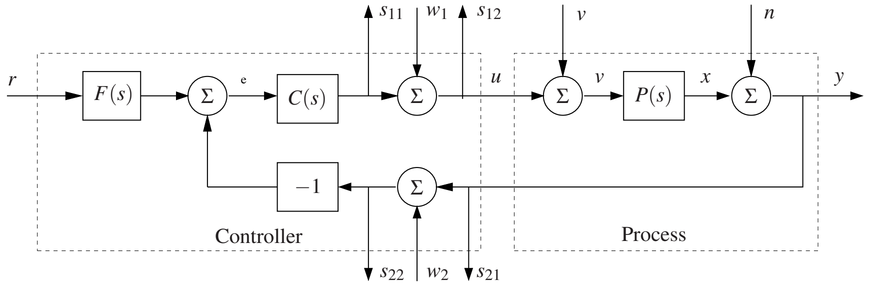


Figure 14.4: Requirements can be tested by injecting signals at test points w_k and measuring responses at s_{ij}. Compare with Figure XXX

from experiments or a combination. Experiments are typically done by changing the control signal and measuring the response. The signals can range from simple step tests to signals that are designed to give optimal information with limited process perturbations. System identification methods and software provide useful tools. The models used typically have different fidelity, cruder in the beginning and more accurate as the design progresses.

Architecture from the Greek arkhitekton (from - ”chief” and ”builder, carpenter, mason”) is both the process and the product of planning, designing, and constructing buildings and other physical structures. In the context of control systems, the physical structure consists of the process, sensors, actuators, computers, communication devices, human machine interfaces, algorithms and software. Architecture describes how these components are connected and how they interact. As for buildings choosing a good architecture is a critical design decision. For engineering systems there are unfortunately no vines that can cover a bad architecture, see Frank Lloyd Wright quote in the beginning of the chapter.

Several design methods have been discussed in Chapters X, XX, and XXX, here are many more in the literature. Many design methods are based on linear models, when environmental conditions change significantly it it necessary to use gain schedling. In many cases there is good software to execute the control design provided that models and criteria are available. There are also nonlinear design methods that are not covered in this book.

Today most control systems are implemented using computer control. Implementation then involves selection of hardware for signal conversion, communication and computing. A block diagram of a system with computer control is shown in Figure 14.5. The filter before the AD converter is necessary to ensure that high frequency disturbances do not appear as alised low frequency disturbances after sampling. The operations of the system are synchronized by a clock

Real time operating systems that coordinates sensing actuation and computing have to be selected and algorithms that implement the control laws must be generated. The sampling period and the anti-alias filter must be chosen carefully. Since a computer can only do basic arithmetic the control algorithms have to be represented as difference equations. They can be obtained by approximating dif-

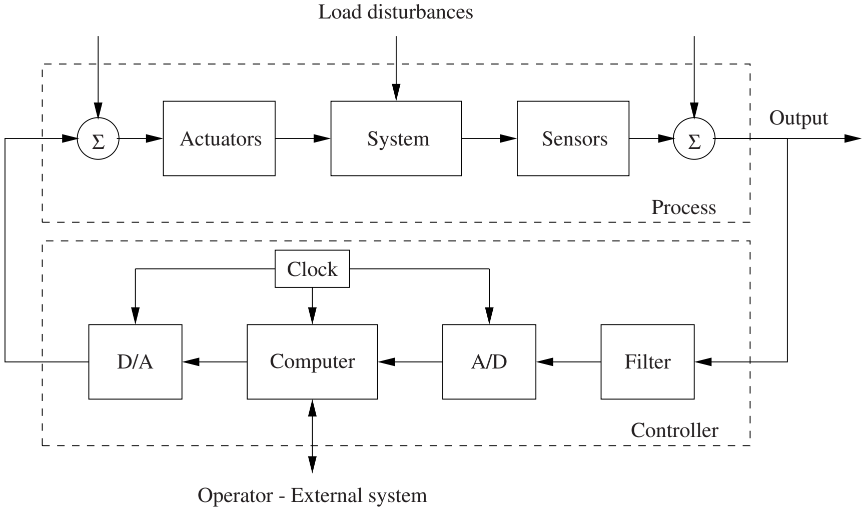


Figure 14.5: Schematic diagram of a control system with sensors, actuators, communications, comuter and interfaces.

ferential equations as was illustrated in XXX but there are also design methods that automatically gives controllers in the form of difference equations. Code can be generated automaticallly. It must also be ensured that computational delays and synchronization of algorithms do not create problems.

When the design is implemented and tested the it must be commissioned, this step may involve adjustment of some parameters, automatic tuning as discussed in Section XXX can be very useful at this stage. During operation it is important to monitor the behavior of the system to ensure that requirements are still satisfied. It may be necessary to upgrade the system when it has been operating. Requirements may also be modified due to operational experiences.

It is highly desirable to have a suite of test programs that can be used throughout the design and operation stages to ensure that requirements are satisfied.

14.3 Bottom-Up Architectures

Building complex systems from standard parts have developed in many branches of engineering. In design of mechanical devices it was found very efficient to standardize nuts an bolts. Design of electronics was similarly done by standardized transistors, circuits, boards and connectors. The Open Systems Interconnection model (OSI model) with seven layers was a key to obtain interoperability in communication systems. The central ideas developed in all fields are standardization, modularisation, layering, and abstraction. Standardization and modularization means that standard components can be developed. Layering means that the system is decomposed into layers with well defined communication protocols. A layer does not require information about the details of a lower layer which means

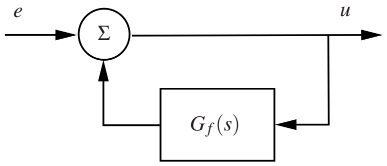


Figure 14.6: Block diagram of a controller with positive of a filtered output signal.

that they can be designed independently. Abstraction is a related concept where the design progresses from simpler to more and more complex views. **Writing can be improved significantly**

Maybe a Picture of Johns hourglass and bowtie?

In the context of control system design bottom-up desing means that the control system is built loop-by-loop. The elements for blocks are controllers (often PID), filters, nonlinear functions, filters and finite state machines. They can either be separate pieces of hardware or function blocks implemented in software, that can be combined graphically in distributed control systems using cut and paste. The system is built loop by loop by combining control principles such as feedback and feedforward, which have been discussed extensively in Chapters **??** and **??** in simple architectures. There are many other architectural structures (control principles) that emerged in an ad hoc fashion in many application areas for example *cascade control*, *mid-range control*, *selector control*and *repetitive control*, *model following*, *gain scheduling*, *adaptation* and *extremal control* which will be discussed in this section. † An advantage with the bottom-up approach is that the system can be commissioned and tuned loop by loop. There may, however, be difficulties when the loops are interacting. The disadvantage is that it is not easy to judge if additional loops will bring benefits. The system can also be unwieldy when loops are added.

RMM: Not sure what is the best terminology: architecture, control principles etxc.

Generalized Integral Control Based On Positive Feedback

In Section **??** is was shown that integral action could be implemented by positive feedback around a first order system as shown in Figure 14.6. The transfer function and the input output relation for the system in the figure is

$$C = \frac{1}{1-G_f}, \qquad u = e + G_f u$$

Intuitively the system works as follows. The filter G_f filters out the signal component that we would like to eliminate, and the filtered output u of G_f is fed back to the input with positive feedback. The net effect is to create a high gain for the frequencies in the pass band of the filter G_f. Integral action was obtained by choosing a low-pass filter but there are many other possibilities:

$$\begin{aligned} C_{constant}(s) &= \frac{1}{1+sT}, & G_{ye} &= 1+\frac{1}{sT} \\ C_{sinusoidal}(s) &= \frac{2\zeta\omega_0 s}{s^2+2\zeta\omega_0 s+\omega_0^2}, & G_{ye} &= \frac{2\zeta\omega_0 s}{s^2+\omega_0^2} \\ C_{periodic}(s) &= e^{-s\tau}, & G_{ye} &= \frac{k}{1-e^{-s\tau}} \end{aligned} \tag{14.1}$$

The controller transfer functions have infinite gains for $s = 0$, $s = i\omega_0$ and $s = 2n\pi i/L$, $n = 0, 1, \ldots$, respectively. The controllers will therefore eliminate, constant disturbances, sinusoidal disturbances with frequency $\omega +_0$ and periodic disturbances with period τ. The input/output relation for $C_{periodic}$ is

$$u(t) = ke(t) + u(t-\tau).$$

This control strategy which is called repetitive control has the property that action at time t is thus a sum of the control error $e(t)$ and the delayed control signal $u(t-\tau)$. The controller will continue to make adjustments If there is a periodic variation in the error e.

KJ Make an exercise of the commented case

The attenuation of periodic disturbances comes at the cost of sensitivity to parameter variations, see Exercise XXX. A compromise between disturbance attenuation can be made by replacing $G_f(s)$ in Figure **??** by $\alpha G_f(s)$ with $\alpha < 1$. The controllers obtained for constant, sinusoidal, and periodic signals then become

$$\begin{aligned} C_{constant}(s) &= \frac{1+sT}{1-\alpha+sT} \\ C_{sinusoidal}(s) &= \frac{s^2+2\zeta\omega_0 s+\omega_0^2}{s^2+2(1-\alpha)\zeta\omega_0 s+\omega_0^2} \\ C_{periodic}(s) &= \frac{1}{1-\alpha e^{-sT}}. \end{aligned}$$

The largest gains of the transfer functions are $1/(1-\alpha)$ in all cases. Choosing $\alpha < 1$ diminishes disturbance attenuation but improves the robustness. The controller $C_{constant}(s)$ is a lag compensator, $C_{sinusoidal}(s)$ a notch filter and $C_{periodic}(s)$ a repetitive controller.

Limiters

Limiters are often used in control systems. In Section XXX we showed how they were used as models for actuatior limitations to avoid integrator windup. Another use of limiters is to make sure that command signals do not create stress on the equipment. The behavior of ordinary limiters is straight forward. Rate limiters create time delays as illustrated in Figure 14.7. A more sophisticated limiter called a *jump and rate limiter* is shown in the lower part of Figure 14.8. The output will

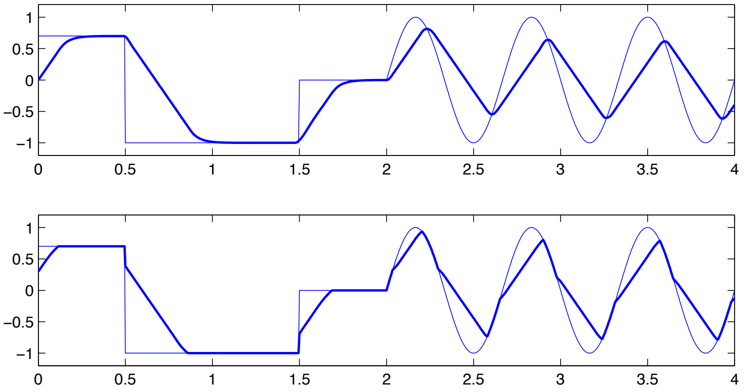


Figure 14.7: Simulation of a rate limiter (upper), and a jump and rate limiter (lower). The thin line shows the input to the limiter and the thick line shows the output of the limiter

follow the input for small changes in the input signal. At large changes, the output will follow the input with a limited rate. The jump and rate limiter can be described by

$$\frac{dx}{dt} = \operatorname{sat}(u-x,-a,a), \qquad y = x+\operatorname{sat}(u-x,-a,a). \tag{14.2}$$

If $|u-x| \leq a$ we have $y = u$, and if $u \geq x+a$ it follows $dx/dt = a$. The output thus admits a jump less than a and for larger input it approaches the input signal at the rate a.

(a) Rate limiter (b) Jump and rate limiter

Figure 14.8: Block diagram of (a) a rate limiter (upper right), and (b) a jump and rate limiter.

The properties of the different limiters are illustrated in the simulation shown in Figure 14.7. The input signal consists of a few steps and a sinusoid. The upper curve shows a rate limiter where the rate limit is 4. The figure shows that the rate

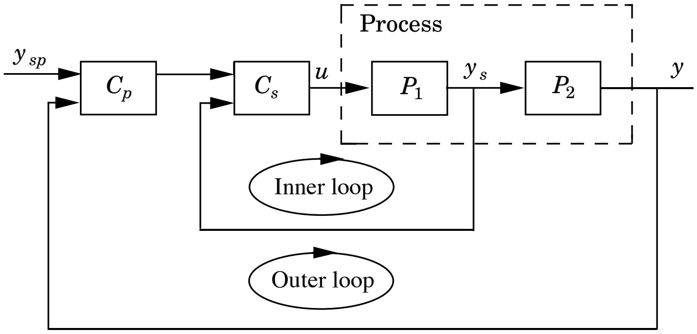


Figure 14.9: Block diagram of a system with cascade control. The system has one control variable u and two measured signals: the primary output y and the secondary or auxiliary output y_s.

of change of the output is limited. The response to a sinusoidal input shows clearly that the rate limiter gives a phase lag. The lower curve shows the response of a jump and rate limiter. Notice that the output follows rapid changes in the input as long as the difference between x and u are less than the jump limit, which is 0.5. The rate is limited to 4.

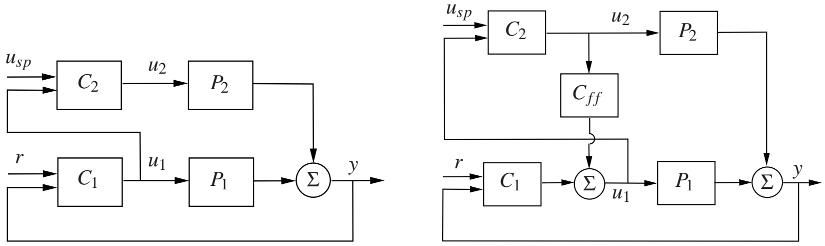
Cascade Control - Several Sensors

Cascade control is a scheme for using two or mor signals and one actuator. The block diagram in Figure 14.9 is an example. The system in the figure has two loops. The inner loop is called *the secondary loop* and the outer loop is called *the primary loop*. The reason for this terminology is that the outer loop deals with the primary measured signal. It is also possible to have a cascade control with more nested loops. The ultimate case is state feedback when all states are measured. Cascade control was used in XXX where it was called inner-outer loop design.

Cascade control is useful in when there is significant time delay or dynamics between the input u and the primary output y, but significantly less dynamics between u and the secondary output y_s, and when the major disturbances enter in the block P_1. Cascade control then admits tight feedback in the inner loop, which reduces the effect of disturbances acting on P_1. If integral action is used in both the secondary and primary control loops, it is necessary to have a scheme to avoid integral windup. Anti-windup for the secondary controller can be done in the conventional way since the controller drives the actuator directly. To provide anti-windup for the primary controller it must be told when the secondary controller goes into anti-windup mode.

Cascade control is a convenient way to use extra measurements to improve control performance.

Give reference to motor control



(a) Midranging (b) Midranging with feedforward

Figure 14.10: Block diagrams of a two architectures with mid-range control. The system in (a) is the basic architecture and (b) is a refined architecture that also uses feedforward.

Mid-Range Control - Paralell Actuators or Subsystems

Midranging is a control architecture for the dual situation is when many control signals are used to control one measured output. An example is given by the block diagram in Figure 14.10, where a single variable y is controlled using two subsystes with individual control signals u_1 and u_2 acting through dynamics described by the transfer functions P_1 and P_2. Assume that P_1 is fast and accurate but with limited actuation range (low control authority) and that P_2 has slow dynamics but wide actuation range (high control authority). The controller C_1 which drives P_1 is the primary controller that has that controls the output y to its desired reference r_1. The second controller C_2 drives the subsystem P_2 which has large range. The measured signal to C_2 is the input u_1 to the subsystem P_2, and the controller C_2 attempts to keep the variable u_1 in its mid range. Suppose that the signal u_1 is in the middle of its operating range and that only small disturbances are acting on the system. In this case, the controller C_1 manipulates u_1 to reduce the disturbance. For large disturbances, the signal u_1 may reach its actuation limit, the controller C_2 then acts to bring u_1 in range.

The block diagram in Figure **??**b is an improvement of the basic mid-range control architecture. The output of the controller C_2 is fed as a feedforward signal to the controller C_1. The feedforward transfer function is

$$G_{ff}(s) = -\frac{P_2(s)}{P_1(s)}.$$

Conventional anti-windup protection can be used in both controllers in Figure **??**a but in the advanced scheme in Figure **??**b the feedforward signal is fed into the feedforward summation point in the controller C_1 to avoid windup of the controller.

Selector Control - Equipment Protection

Selector control is commonly used to control a primary variable while keeping auxiliary variables within given constraints for safety or for equipment protection.

A selector is a static device with many inputs and one output. There are two types of selectors: *maximum* and *minimum*. For a maximum selector the output is

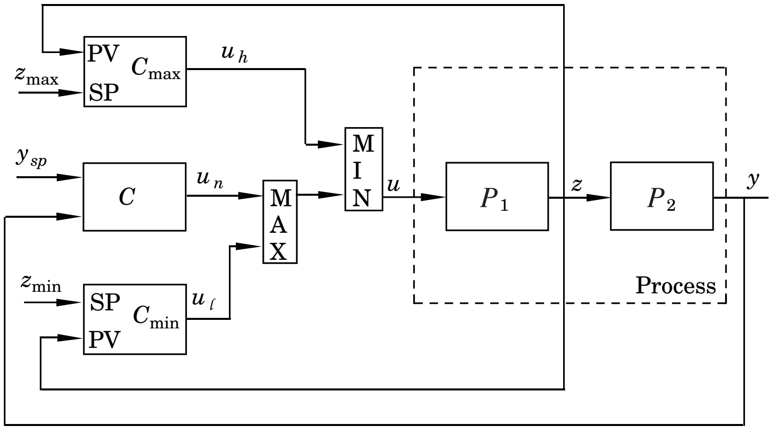


Figure 14.11: Block diagram of a system with selector control. The primary controller is C which attempts to keep y close to its reference value. The controllers C_{max} and C_{min} swithches control objectie when the variable z is outside its pemissible range.

the largest of the input signals.

Selector control is commonly used in the power industry for control of boilers and nuclear reactors. The advantage is that it is built up of simple nonlinear components and PI and PID controllers. One example of use is where the primary controlled variable is temperature and we must ensure that pressure does not exceed a certain range for safety reasons, another is compressor control where the objective is to maintain a pressure while avoiding compressor surge. An alternative to selector control is to combine of ordinary controllers with logic.

The selector control architecture is illustrated in Figure 14.11. The primary controlled variable is the process output y. There is an auxiliary measured variable z that should be kept within the limits $z_{\min}$ and $z_{\max}$. The primary controller C has process variable y, set point r, and output u_n. There are also secondary controllers with measured process variables that are the auxiliary variable z and with set points that are bounds of the variable z. The outputs of these controllers are u_h and u_l. The controller C is an ordinary PI or PID controller that gives good control under normal circumstances. The output of the minimum selector is the smallest of the input signals; the output of the maximum selector is the largest of the inputs.

Under normal circumstances the auxiliary variable is larger than the minimum value $z_{\min}$ and smaller than the maximum value $z_{\max}$. This means that the output u_h is large and the output u_l is small. The maximum selector, therefore, selects u_n, and the minimum selector also selects u_n. The system acts as if the maximum and minimum controller were not present. If the variable z reaches its upper limit, the variable u_h becomes small and is selected by the minimum selector. This means that the control system now attempts to control the variable z and drive it towards its limit. A similar situation occurs if the variable z becomes smaller than $z_{\min}$.

In a system with selectors, only one control loop at a time is in operation. The

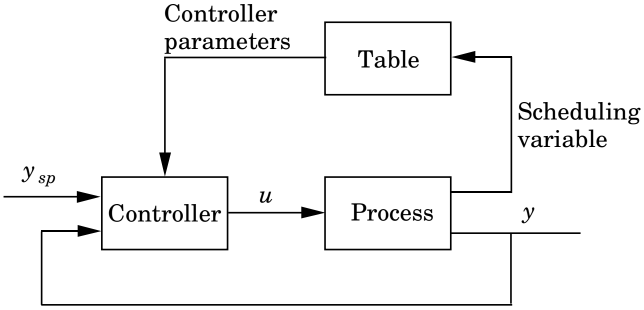


Figure 14.12: Block diagram of a system with gain scheduling.

controllers can be tuned in the same way as single-loop controllers. There may be some difficulties with conditions when the controller switches. With controllers having integral action, it is also necessary to track the integral states of those controllers that are not in operation.

So far we have only discussed maximum and minimum selectors, there are also other types of selectors such as the median selector whose outpu is the current median of the input signals. A special case is the two-out-of-three selector, commonly used for highly sensitive systems. To achieve high reliability it is possible to use redundant sensors and controllers. By inserting median selectors it is possible to have a system that will continue to function even if several components fail.

When using selector control it is important to have windup protection for controllers that are not selected, a simple way to do this is to feed use the output of the selected controller as the tracking signal for the other controllers.

Gain Scheduling

Feedback controllers can be designed to be robust to process variations. Robust control can, however, not deal with very large parameter variations. Gain scheduling is a technique that can be used when there are measured variables (scheduling variables) that correlate well with the process variations. By measuring such variables the controller parameters can then be changed accordingly. A block diagram of a controller with gain scheduling is shown in Figure 14.12. The controller design is performed for many values of the scheduling variables the controller parameters are stored in a table. During operation, the scheduling parameter is measured and the controller parameters are interpolated from the table. Care must be exercised to avoid bumps when changing controller parameters.

Systems with gain scheduling are routinely used for flight control where the scheduling variables are Mach number and height. In process control flow rates and production rates are typically used as scheduling variables.

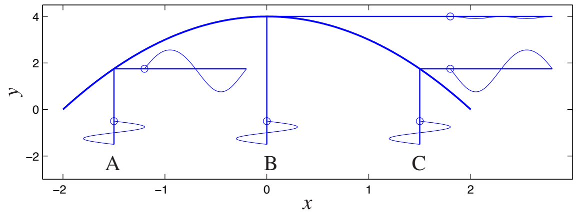


Figure 14.13: Principle for extremal seeking. The figure shows the steady state response of the the performance variable v as a funtion of the reference y_r and the effect of sinusoidal variations of the reference.

Extremum Seeking

Among other useful control structures we can mention *extremum seeking* of *self-optimization*. Instead of keeping the process output close to a specified reference value these controller these attempt to change the reference so that an objective function is minimized. To accomplish this it is necessary to change the reference value of the controller and observe the effect of the output. A simple scheme idea is illustrated in Figure **??**. The reference value of a controller is changed and the behavior of the performance criterion is observed. The performance changes very little close to the optimum. The changes are of performance are in phase with the changes of the reference if the reference is too large and out of phase if the reference is too low. The reference is changes to move towards the smallest value of the objective function. Correlation methods can be used to filter out noise, since the argument is based on a steady–state reasoning the frequency of the perturbation signal must be chosen so low that process dynamics can be neglected. There are many other more sophisticated schemes based on optimization and estimation. There are many other schems for finding optimal operating conditions. They are all characterized by the steps of probing, analysis and action. Many efficient optimization methods can be used if computation power is available.

The Smith Predictor - Phase Advance

A special controller architecture for dealing with systems having time delays have been proposed by Otto Smith. A block diagram of the controller is shown in Figure 14.14. Let the process have the transfer function $P = P_0 e^{-s\tau}$, where τ is the time delay of the process. The controller is provided with a model in parallell with the process. The provides the signal y_p which is a proxy for the output y without the the time delay and the controller C_0 is designed for the delay free dynamics P_0. Assume that $\hat{P}_0 = P_0$ then the signal ε is zero for all u. Applying block diagram algebra then gives the following transfer function for the closed loop system

$$G_{yr}(s) = \frac{P_0(s)C_0(s)}{1+P_0(s)C_0(s)} e^{-s\tau}. \tag{14.3}$$

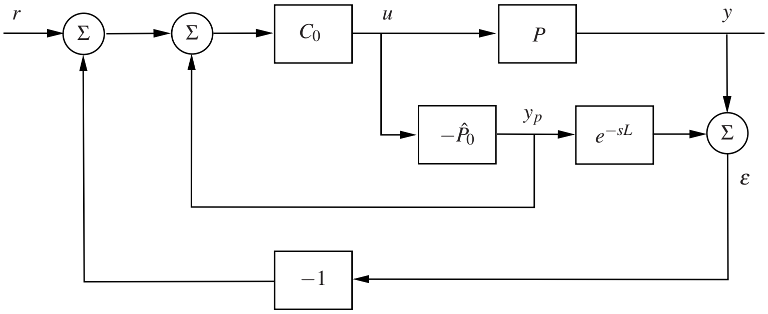


Figure 14.14: Block diagram of a closed loop system with a Smith predictor.

Make an exercise **KJ**

To obtain a desired set point response we can just design a controller for a process without the time delay.

To get some insight into the properties of the Smith predictor we observe that if $P = P_0 e^{-s\tau}$ and $\hat{P}_0 = P_0$ the block digram Figure 14.14 can be redrawn as a conventional feedback loop with the controller

$$C = \frac{C_0}{1 + C_0\bar{P}_0(1 - e^{-s\tau})} = C_0 C_{pred}, \qquad C_{pred} = \frac{1}{1 + C_0\bar{P}_0(1 - e^{-s\tau})} \tag{14.4}$$

The controller can thus be viewed as a cascade connection of the conventional controller C_0 with the predictor C_{pred}. Notice that near the gain crossover frequency for C_0P_0 we have $C_0P_0 \approx -1$ and $C_{pred} \approx e^{sL}$ indicating that the transfer function $C_{pred}(s)$ has a significant phase advance. The Bode plot of C_{pred} in Figure 14.3 shows that this is indeed the case. Notice in particular that the phase is very close to the phase of the ideal predictor e^{sl} for the frequencies $\omega = 0.8$, and 1.6.

The Smith predictor gives closed loop systems with good set point responses but load disturbance responses are not much better than those obtained with PI control. The predictor C_{pred} is however a useful transfer function to provide phase advance. Another drawback with the Smith predictor is that it does not work for processes with integration. The architecture with two parallell paths which contains integrators is a prototype example for a system that is neither reachable not observable.

Internal Model Control

Figure 14.16 shows a controller architecture that is closely related to the Smith predictor. The architecture in is called *internal model control IMC*. In the figure $\hat{P}$ denotes a model of the process, and G_f is a low-pass filter. If $\hat{P}(s) = P(s)$ it follows from Figure 14.16 that the closed loop response to set point changes is given by $G_{yr} = G_f$. The name *internal model controller* derives from the fact that the controller contains a model of the process internally. This model is connected in parallel with the process.

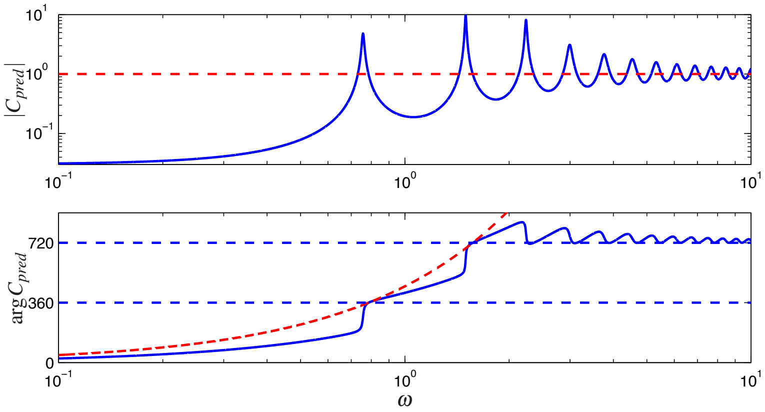


Figure 14.15: Bode plot of the predictor $C_{pred}(s)$ (blue solid) and the ideal predictor e^{sL} (red dashed lines) for $P_0C_0 = \frac{1}{s+1}\left(1+\frac{1}{0.45s}\right)$ (blue full lines) and $L = 8$.

If the model matches the process, i.e., $\hat{P} = P$, the signal e is equal to the disturbance d for all control signals u and the closed loop transfer function is G_{yr}. The signal e is therefore also called a disturbance observer.

The controller obtained by the internal model principle can be represented as an ordinary series controller with the transfer function

$$C = \frac{G_{yr}\hat{P}^{-1}}{1 - G_{yr}\hat{P}^{-1}\hat{P}}. \tag{14.5}$$

From this expression it follows that controllers of this type cancel process poles and zeros, which implies that it cannot be usef for processes with unstable poles.

Posicast Control

Posicast control is another invention of Otto Smith. Consider the problem of moving a hanging load illustrated in Figure 14.17. To move the load from position A to

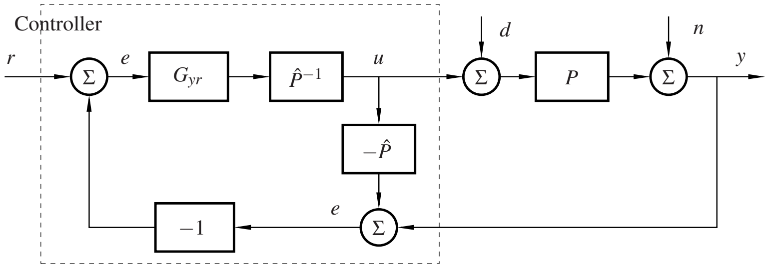
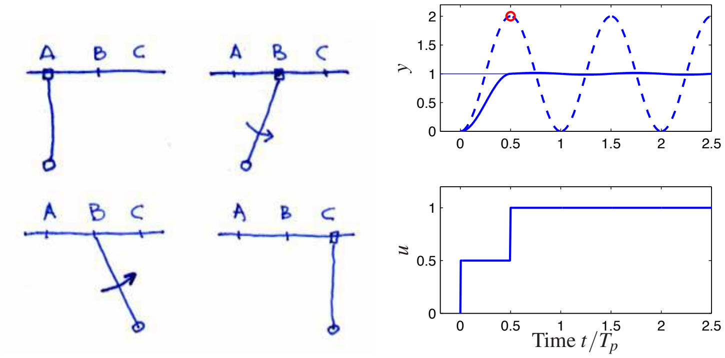


Figure 14.16: Block diagram of a closed-loop system with a controller based on the internal model principle.



(a) Idea of posicast principle (b) Input-output responses

Figure 14.17: Posicast control of a hanging load. The idea is illustrate in (a) and (b) shows time responses. The dashed line shows the response to a unit step and the full line shows the response to a unit step reference signal using the posicast controller.

position C without overshoot the cart is first moved half way to B. The load then swings towards the desired point, when this is reached the cart is quickly moved to the desired position (C) and the system is in equilibrium at the desired position. **O. J. M. Smith Posicast control of damped oscillatory systems, Proc. IRE. (45) 1957, 1249-1255**

Posicast control can be modeled as a feedforward with the transfer function

$$G_{ff}(s) = \frac{1}{2}\left(1+e^{-sT_p}\right), \tag{14.6}$$

where T_p is the period of the oscillation. This transfer function has zeros at $s = i\omega$, where $\omega = \omega_0$, $3\omega_0$, $5\omega_0\dots$, where $\omega_0 = 2\pi/{}_pT$. The transfer function thus blocks these frequencies very effectively. The unit step response of C_{ff} is shown in Figure 14.17, notice that the output settles without overshoot at $t = T_p/2$.

The transfer function in (14.6) given above is for the ideal case of a hanging load without damping, when the load is modeled as an undamped second order system. If the system is of second order with damping ratio ζ the transfer function is instead modified to

$$C(s) = \frac{k_i}{s}\left(\gamma+(1-\gamma)e^{-sT_d}\right),\ \gamma = \left(1+e^{-\zeta_0\pi/\sqrt{1-\zeta_0^2}}\right)^{-1},\ T_d = \frac{\pi}{\omega_0\sqrt{1-\zeta_0^2}} \tag{14.7}$$

Posicast control has been usef very successfully for control of cranes. It is also used in drive systems for microsystems which typically have very low damping. Using the posicast controller it is possible to make accurate positioning without exciting the oscillatory modes.

Complementary Filtering

Complemenatry filtering is a technique that can be used to fuse the information from different sensors, typically one sensor that is slow but accurate and another that is fast but may drift. Consider the situation when we want to give a good estimate of the variable x and that two sensors which gives the signals y_1 and y_2 are available

$$y_1 = x + n_1, \qquad y_2 = x + n_2$$

The disturbance n_1 has zero mean but the disturbance n_2 may drift indicating that l y_1 is slow but accurate and y_2 fast but drifting. The complementarly filter for recovering the signal x is given by

$$X_f(s) = \frac{1}{s+1}Y_1(s) + \frac{s}{s+1}Y_2(s) \tag{14.8}$$

Notice that the transfer functions for y_1 and y_2 sums to 1, which explains the name complementary filtering.

Both complementary filtering and Kalman filtering can provide improved estimates by fusing information from several sensors, they can also be obtimized if information about the noise is available. Complementary filtering requires only a model of the sensor system but the Kalman filter requires a model of the complete system dynamics. The Kalman filter can, however, also exploit the control actions. Both methods are widely used for sensor fusion both in simple and advanced systems.

Here is something that can be converted to an exercise Signal model (y_1 slow but accurate, y_2 fast but drifting) Filter for recovering the variable x

Choose G_1 as low pass filter, G_2 then becomes high pass.

Model the measured value x_1 and the drift of the second sensor as unknown constants

$$y_1 = x_1 + n_1, \qquad y_2 = x_1 + x_2 + n_2, \qquad \dot{x}_1 = 0, \qquad \dot{x}_2 = 0$$

The Kalman filter

$$\frac{d}{dt}\begin{pmatrix}\hat{x}_1\\ \hat{x}_2\end{pmatrix} = \begin{pmatrix}k_{11}(y_1 - \hat{x}_1) + k_{12}(y_2 - \hat{x}_1 - \hat{x}_2)\\ k_{21}(y_1 - \hat{x}_1) + k_{22}(y_2 - \hat{x}_1 - \hat{x}_2)\end{pmatrix}$$

After some calculations

$$\begin{aligned}\hat{X}_1(s) &= \frac{k_{11}s + k_{11}k_{22} - k_{12}k_{21}}{s^2 + (k_{11} + k_{22} + k_{12}) + k_{11}k_{22} - k_{12}k_{21}}Y_1(s)\\ &\quad + \frac{k_{12}s}{s^2 + (k_{11} + k_{22} + k_{12}) + k_{11}k_{22} - k_{12}k_{21}}Y_1(s)\end{aligned}$$

Integral Control and Windup Protection

It is convenient to build a control system bottom up by combining different components. There are, however, some important considerations. The different components may interact in an undesirable fashion. Particular care must be given to actuator staurations and integrator windup. When loops are cascaded it is important to propagate actuator saturations up the chain, a typical example is cascade control. In configurations like mid-range control and other configurations were several controller combine it is important to propagate information between the controllers.

Thes issues are easo deal with in sim[ple systems but a top-down approach may be preferrable for more complicated systems.

14.4 Interaction

A drawback with the bottom up approach when the system is built loop by loop is that there may be unintended interactions. It is therfore important to investigate when interactions. We will start by investigating a system with two inputs and two outputs, let the transfer function and its inverse be

$$P(s) = \begin{pmatrix} P_{11}(s) & P_{12}(s) \\ P_{21}(s) & P_{22}(s) \end{pmatrix}, \quad P^{-1}(s) = \frac{1}{\det P(s)} \begin{pmatrix} P_{11}(s) & -P_{21}(s) \\ -P_{12}(s) & P_{22}(s) \end{pmatrix}, \tag{14.9}$$

where $\det P(s) = P_{11}(s)P_{22}(s) - P_{12}P_{22}$. Assume that we want to control the system by two single loop controllers. The first problem is to decide if y_1 should be controlled by u_1 or u_2, this is called the *pairing problem* and the second problem is to investigate if there will be interactions between the loops. Sometimes the solution to the problem is clear from the physics of the process. The problem can resolved by trial and error: design controller for the different alternatives and explore their properties.

A clever idea that gives a lot of insight with modest calculations was proposed by Bristol. Assume for simplicity that we are exploring the possibility of controlling y_1 by u_1 and y_2 by u_2. Controllers can be designed based on the transfer functions $P_{11}(s)$ and $P_{22}(s)$. The controller for the first loop will then work well if the second loop is open but the question is how the second loop will influence the second loop. Bristol proposed to look at the ratio

$$\lambda(s) = \frac{P_{11}(s)}{\bar{P}_{11}}, \tag{14.10}$$

where $\bar{P}_{11}(s)$ is the tranfer function from u_1 to y_1 when the *prefectly controlled* meaning $y_2 = 0$. Assuming all signals are exponential functions we have for perfect control of the second loop

$$y_1 = P_{11}u_1 + P_{12}u_2, \qquad 0 = P_{21}u_1 + P_{22}u_2, \qquad y_1, u_1, u_2 \in \mathscr{E}.$$

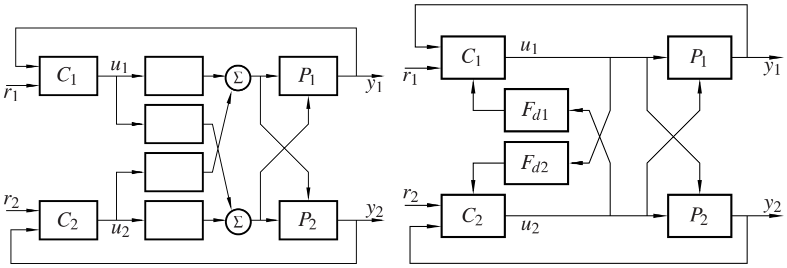
Eliminating u_2 in these equations give

$$y_1 = \frac{P_{11}P_{22} - P_{12}P_{21}}{P_{22}}u_1,$$

and we find

$$\bar{P}_{11} = \frac{P_{11}P_{22} - P_{12}P_{21}}{P_{22}}, \quad \text{and} \quad \lambda = \frac{P_{11}P_{22}}{\det P}. \tag{14.11}$$

If $|\lambda(i\omega)| > 1$ is larger than one the interaction increases the gain and it decreases the gain if $|\lambda(i\omega)| > 1$. There is no interaction if $\lambda(s) = 0$, neither if $\lambda(s) = 1$ but the loops should then be reversed so that y_1 is controlled by u_2. There are rules of thumb for interpreting the relative gain array, interactions can typically



(a) Direct decoupling (b) Indirect decoupling

Figure 14.18: Two decoupling schemes.

be neglected for $3/4 < |\lambda(i\omega)| < 3/2$, which means the controller must then be designed to cope with an additive process uncertainty such that $|\Delta/P| < 1/3$, see Figure 13.2. Decoupling or multivariable control should be considered outside this range.

It follows from (14.9) that $\lambda(s)$ is the procduct of the 11 elements of P and P^{-1} it turns out that the analysis can be generatlized to systems with n inputs and n outputs and the interactions can then be characterized by the matrix

$$\Lambda(s) = P(s) \circ P^{-T}(s) = P(s).*P^{-T}(s) \tag{14.12}$$

where $P^{-T}(s)$ denotes the transpose of $P^{-1}(s)$ and $\circ$ denotes element by element multiplication of matrices (Hadamard product). The matrix $\Lambda(s)$, which was originally derived for the steady state case ($s = 0$) is called *the relative gain array (RGA)* or *Bristol's RGA*, it was later extended to dynamics.

The relative gain array has the nice property that it is dimension free and that it gives insight into interactions and pairing of variables, by analysing the gain $|\Lambda(i\omega)|$ we also get insight into interactions at different frequencies. The RGA also gives information about the variables that should be grouped for multivariable control, see [**?**].

Decoupling

Decoupling is one way to deal to reduce the interactions between the loops. The idea is to provide the controller with a compensator which reduces the interactions. Two ways of making decoupling; direct decoupling and feedback decoupling are illustrated in Figure 14.4. In *direct decoupling* the controller is provided with a post compensator that reduces the effects of the interactions.s to design a controller that reduces the effects of the interaction. In *feedback decoupling* the compensation is instead arranged by feeding the output of one controller to the other controller and vice versa. The expression for the compensator for direct decoupling is a complicated expression in the process transfer funtion. The corresponding expressions for feedback decoupling are much simpler. If the process transfer function is given by (**??**) it becomes

$$F_{d1} = -P_{11}^{-1}P_{12}, \qquad F_{d2} = -P_{22}^{-1}P_{21}.$$

Feedback decoupling has another advantage. Since the controllers C_1 and C_2 act directly on the actuators a conventional anti-windup scheme will work provided that the cross-coupling signals are entered as feedforward signals in the controllers.

Parallel Systems

There are situations when several subsystems are used to control the same variable. Typical examples are: temperature control using when several cooling or heating devices and control of an electric car with one motor on each wheel. An extreme example is control of a power grid shich may have hundreds of energy sources. Care must be execised when making loop-by-loop design of such systems. We illustrate by an a simple example.

Example 14.1 Cruise Control for Electric Car
Consider speed control of an electrical car with motors on each wheel, for simplicity we will consider linear motion with only two motors, and we will use the simple model (4.1) in Section 4.1. Neglecting all disturbance forces F_d except the force due to gravity; F_r the model (4.3) becomes

$$m\frac{dv}{dt} = F_1 + F_2 - mg\theta, \tag{14.13}$$

where v is the speed of the car, θ the slope of the road, F_1 and F_2 the forces generated by the drive motor.

We will first consider the case when both motors have proportional controllers. Let v_r be the desired (reference) speed, the controllers are then

$$F_1 = k_{p1}(v_r - v), \qquad F_2 = k_{p1}(v_r - v,) \tag{14.14}$$

Combining equations (14.13) and (14.14) gives the following equation for the closed loop system

$$m\frac{dv}{dt} = (k_{p1} + k_{p2})(v_r - v) - mg\theta.$$

If the slope θ is constant there will be a steady state error $e_{ss} = v_r - v_{ss}$ and the steady-state controller outputs are $u_{1ss} = k_{p1}e_o$ and $u_{2ss} = k_{p1}e_o$. The proportional gains k_{p1} and k_{p2} thus determine how the *compensation for the disturbance is distributed among the motors.*

Next we will consider the case when each motor is provided by a PI controller. The closed loop system is then described by the equations

$$\begin{aligned} m\frac{dv}{dt} &= (k_{p1} + k_{p2})(v_r - v) + k_{i1}I_1 + k_{I2}I_2 + mg\theta, \\ \frac{dI_1}{dt} &= v_r - v, \qquad \frac{dI_2}{dt} = v_r - v \end{aligned} \tag{14.15}$$

This system is not stable, since $\frac{d(I_1 - I_2)}{dt} = 0$ the system has an eigenvalue at the origin. The system (14.15) has the inputs v_r and θ the output y, it is of third order.

The system has a the Kalman decomposition Figure 8.12a, with a subsystem $\Sigma_{\bar{r}\bar{o}}$, with integrator dynamics, that is neither reachable nor observable.

If both wheel motors are provided with controllers having controller integral action the integral terms of the controllers will drift in such a way that their sum is constant. As a result one control signal may increase and the other will decrease until saturation occurs. There is a very simple remedy, use one single integrator and distribute the output of that integrator to both motors. ∇

The results of the example can be generalized, if parallel systems are controlled by proportional controllers, then *the controller gains determine how disturbance attenuation is divided among the subsystems*. Moreover, *integral control cannot be used in the individual subsystems*, instead we can select one controller with integral action or we can use *a central integrator* and distribut its output to the controler of the subsystems.

14.5 Top-Down Architectures

- Introduction: controllers, logic
- Logic and FMS
- State Feedback and Observers
- State Based Control, FSM
- Model Predictive Control

Top-down paradigms start with a problem formulation in terms of an optimization problem. Paradigms that support a top-down approach are optimization, state feedback, observers, predictive control, and linearization. In the top-down approach it is natural to deal with many inputs and many outputs simultaneously. Since this is not the main topic of this book we will only give a brief discussion. The top-down approach often leads to the controller structure shown in Figure 14.19. In this system all measured process variables y together with the control variables u are sent to an observer, which uses the sensor information and a mathematical model to generate a vector $\hat{x}$ of good estimates of internal process variables and important disturbances. The estimated state $\hat{x}$ is then compared with the ideal state x_m produced by the feedforward generator, and the difference is fed back to the process. The feedforward generator also gives a feedforward signal u_{ff}, which is sent directly to the process inputs. The controller shown in Figure 14.19 is useful for process segments where there are several inputs and outputs that interact, but the system becomes very complicated when there is a large number of inputs and outputs. In such a case it may be better to decompose the system into several subsystems.

An advantage with the top-down approach is that the total behavior of the system is taken into account. A systematic approach based on mathematical modeling

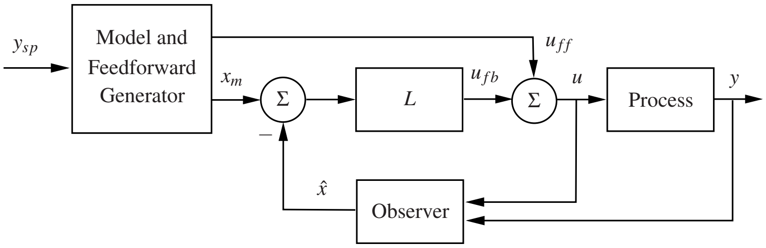


Figure 14.19: Block diagram of a controller based on model following, state feedback, and an observer.

and simulation makes it easy to understand the fundamental limitations. Commissioning of the system is, however, difficult because many feedback loops have to be closed simultaneously. When using the top-down approach it is therefore good practice to first tune loops based on simulation, possibly also hardware in the loop simulation.

14.6 Adaptation Learning and Cognition

In this section we will discuss more sophisticased control laws with abilities to adapt, learn and reason. These funtions are key elements for autonomous control. Before proceeding we will consider review the maning of the words. Adapt is to adjust to a specified use or situation, learn is to acquire knowledge or skill by study, instruction or experience, reason is the intellectual process of seeking truth or knowledge by infering from either fact of logic and autonomy is the ability of beeing self-governing. When these words are used in the engineering context it is clear that the abilities are far from what we can accomplish as humans, but the development of autonomous cars and airvehicles are good indicatotrs of progress.

Adaptive Control

Adaptive control is a technique that can be used when there are significant variations in the process and its environment and where neither robust control nor gain scheduling is applicable. Model reference control and the self-tuning controller are two common approaches to adaptive control, see Figure 14.20. Model reference adaptive control (MRAS) is primarily used for command signal following and the self-tuning regulator is used both for intended for reduction of load disturbances. Notice in Figure 14.20 that there are two feedback loops: one conventional feedback loop involving the process P and the controller C and a slower loop to adjust the controller parameters θ.

Model Reference Adaptive Control

Model reference adaptive control (MRAS) is primarily used for command signal following. A block diagram of the controller is shown in Figure 14.20a. The con-

(a) Model reference adaptive control (b) Self-tuning regulator

Figure 14.20: Block diagrams of systems with adaptive control. The left figure shows a model reference adaptive system (MRAS) and the right figure shows a self-tuning regulator (STR). The block P is the process, C is a controller with adjustable parameters θ, ϑ, G_m is a model that gives the ideal response to command signals, PA is a parameter adjustment mechanism, RPE is a recursive parameter estimator that estimates process parameters recursively and CDC is a design calculation that computes the controller parameters ϑ from the process parameters θ.

troller consists of three blocks G_m, C and PA. The desired response to command signals is given by the transfer function G_m. The controller C has adjustable parameters θ. The parameter adjustment mechanism PA, receives the process input u and output y and the derired response y_m and it generates the the process parameters. The MIT rule is given by

$$\frac{d\theta}{dt} = -\gamma e \frac{\partial e}{\partial \theta}, \tag{14.16}$$

where $e = y_m - y$ and γ is a parameter, is a very simple way to adjust the parameters. There are many other rules, some of them are derived from Lyapunov theory.

The Self Tuning Regulator

The self-tuning regulator is used both for command signal following and for regulation. The controller is based on the idea of developing a process model automatically and to apply some design method to find a suitable controller. A block diagram of a system is shown in Figure 14.20b. The controller has three blocks, a controller C with adjustable parameters θ, a recursive parameter estimator RPE and a controller design calculation CDC. The parameter estimator RPE estimtes the process parameters ϑ recursively from the process input u and output y. The controller design block CDC determines the controller parameters from the process parameters using some design method. In this calculation it is common to treat the estimates as the true parameters, a principle from decision making under uncertainty called the certainty equivalence principle, see [**?**]. Uncertainties in the estimates can be taken into account because many estimation schemes provide estimates of parameter uncertainty. The self-tuning regulator is very flexible because many different methods can be used for parameter estimation and control design.

Recursive least squares is a simple way to estimate process parameters. For the model

$$\begin{aligned} y_{t+1} &= -a_1 y_t - a_2 y_{t-1} + \cdots + b_1 u_t + \cdots + e_{t+1} = \varphi_t^T \theta + e_{t+1} \\ \varphi_t &= \begin{pmatrix} -y_t & -y_{t-1} & \cdots & u_t & u_{t-1} & \cdot \end{pmatrix} [s] \\ \vartheta &= \begin{pmatrix} a_1 & a_2 & \cdots & b_1 & b_2 & \cdots \end{pmatrix}, \end{aligned} \tag{14.17}$$

the parameter estimates are given by

$$\begin{aligned} \hat{\vartheta}_t &= \hat{\vartheta}_{t-1} + K_t (y_t - \varphi_t \hat{\vartheta}_{t-1}) \\ K_t &= P_t \varphi_t, \qquad P_t = P_{t-1} \varphi_t (\lambda + \varphi_t^T P_{t-1} \varphi_t)^{-1} \end{aligned} \tag{14.18}$$

The parameter λ controls how quickly old data is forgotten There are many other versions of the parameters estimator, of particular interest versions with directional forgetting and square root versions, where the square root of P is updated instead of P itself.

Applications of adaptive control are found in process control, for ship steering and aerospace applications we illustrate by an example.

Perhaps a few lines about aerospace from Lavretsky **KJ**

Example 14.2 An adaptive ship steering auto-pilot
A conventional autopilot for ship steering is based on PID control. The major disturbances are due to wind and waves, which can change significantly during operation of the ship. It is attractive to use an adaptive controller that attempts to model the external disturbances. This can be captures by a model of the form (14.17) with 4 *a*-parameters, constrained to contain an integrator, and 2 *b*-parameters parameters. Extensive sea trials have shown that the adaptive autopilot has better performance than the conventional autopilot in normal weather conditions and substantially better performance in bad weather conditions. Figure 14.21 shows results from evaluation of the Steermaster autopilot developed by Kockums and now marketed by Northrop Grumman. In the experiments the conventional and adaptive autopilots were run repeatedly for about an hour each during normal operation The figure shows that the adaptive autopilot has significantly smaller variations in heading than the conventional autopilot. The difference corresponds to about 3% less fuel consumption. ∇

Excitation an Estimation in Closed Loop

To obtain reliable estimates of the process parameters it is necessary that there are sufficient variations in the control signal. The excitation can be captured formally by

$$c(k) = \lim_{t\to\infty} \frac{1}{t} \sum_{i=1}^{t} u(i) u(i-k).$$

Figure 14.21: Comparison of a conventional autopilot and an adaptive autopilot for ship steering. The experiments were performed on the tanker Seascape, a 225000 ton tanker. The wind velocity was 20 m/s.

A necessary condittion obtaining reliable estimates for the model (**??**) is that the matrix

$$C_n = \begin{pmatrix} c(0) & c(1) & \dots & c(n-1) \\ c(1) & c(0) & \dots & c(n-2) \\ \vdots & & & \\ c(n-1) & c(n-2) & \dots & c(0) \end{pmatrix},$$

is positive for n larger than the number of parameters in the model (14.17). A signal with this property is called *persistently exciting* (PE) of order n An equivalent condition is that

$$U = \lim_{t\to\infty} \frac{1}{t} \sum_{k=1}^{t} (A(q)u(k))^2 > 0,$$

for all nonzero polynomials A of degree $n-1$ or less. It follows that a sinusoid is PE of order 2.

A necessary condition for obtaining reliable estimates of the parameters of the model (**??**) is that the input signal is persistently exciting of an order equal to the number of parameters in the model.

Another difficulty with adaptive control is that parameter estimates are performed when the system is in closed loop. It is then very important where the excitation signals occur. Consider for example the standard feedback loop in Figure 14.17. If the only perturbation on the system is the signal v we have

$$Y(s) = \frac{P(s)}{1+P(s)C(s)}v, \qquad U(s) = -\frac{C(s)P(s)}{1+P(s)C(s)}v,$$

and it thus follows that $Y(s) = -\frac{1}{C(s)}U(s)$ any attempt to find a model relating u and y will thus result in the negative inverse of the controller transfer function. However, if $v = 0$ we have instead

$$Y(s) = \frac{P(s)C(s)}{1+P(s)C(s)}(F(s)R(s)-N(s)), \qquad U(s) = \frac{C(s)}{1+P(s)C(s)}(F(s)R(s)-N(s)v,$$

hence $Y(s) = P(s)U(s)$ and the process model can indeed be estimated.

To have a reliable parameter estimation it is thus important to be aware of where disturbances enter and to monitor the excitation. Load disturbances of the process are particularly harmful. A scheme for detecting harmful load disturbances is presented in see Hagg+Ast. To obtain reliable estimates it is necessary to monitor the excitation of the process and only update parameters when there is sufficient excitation of the process.

Dual Control

An interesting approach to control of uncertain systems was proposed by Feldbaum, who emphasized that control should be *investigating* as well as *directing* and he coined the term dual control for this property. Feldbaum used optimal stochastic control to obtain a controller that was actively introducing perturbations in the process when the process was not properly excited by natural disturbances. The hyper state of a dual controller is the conditional probability distribution of the regular states of the process and the parameters. The computations of dual controller can only be performed in simple cases because the state of the system is a conditional probability distribution over states and parameters many adhoc schemes to monitor excitation and to introduce perturbations when needed have therefore been produced [].

Learning

A nonlinear function with a learning mechanism is a simple example of a learning system. The function is created automatically by providing it with a large number of arguments and corresponding values. Two central issues are representation of the function and construction of the learning mechanism. A simple way to represent a function of several variables is to quantize the variables which we illustrate by an example.

Example 14.3 Michie's Boxes
Michie and Chambers developed a simple program for the classical program to

stabilize an inverted pendulum. Cart position and velocity and pendulum angle and angular velocity are measured. The control signal is thus a function: $f:\mathbb{R}^4 \to \mathbb{R}$. To implement the system the states were quantized crudely: 5 levels for position and angle and 3 levels for velocities, only two levels were used for the control signal. The control law can thus be represented by a table with 225 entries. The control law is obtained by introducing the notions of life and usage. The life of a decision is the number of future decision taken before failure and the usage is the number of decisions taken on entry to this box. There is also a separation of left and right lifes and decisions. Heuristic rules based on the values of the 4 entries LL, RL, LU and RU are used to determine weather to apply a control to the left or to the right. The system is initialized for example by introducing random numbers in the table. Experiments are run and the table is updated. In a typical experiment the system was able to stabilize the pendulum in 25 minutes after a 60 hour training period, see Michie, D., & Chambers, R. A. (1968). Boxes: An Experiment in Adaptive Control. In E. Dale. & D. Michie (Eds.), Machine Intelligence 2. Edinburgh: Oliver and Boyd. ∇

Pendulum stabilization is not the best case to demonstrate learning, since it would take less than 60 hours for any reasonably good student to design a controller that swings up and stabilizes an inverted pendulum. Control performance in stabilization will also be better because a conventional design can avoid the crude quantization used in Boxes.

Boxes is similar to algoritms game-playing algorithms, where the state is determined and appropriate actions for each state are developed.

Neural Networks

- size: 10-20 layers, millions of weights and billions of interconnections
- machine learning, computer vision, handwritten speech and character recognition
- Convolution nets ConvNet
- sharper discussion supervised vs unsupervised
- hardware
- refLeCun, Bengio and Hinton Deep Learning Nature 521 (2015) 436-444

A severe drawback of schemes like Boxes is that the nonlinear function is represented by gridding the state variables. To have efficient learning schems it is necessary to find more efficient ways to represent nonlinear functions. Artificial neural betworks is suitable representation.

A real neuron has many synapses which permits it to receive inhibitory or exhibitory signals and it will emit a pulse if the received signals are above a certain

Architecture.tex, v6038 2015-09-08 03:27:09Z (murray)

Figure 14.22: Schematic diagram of an neural network with two-layers.

level. An artificial neuron mimics a real neuron but it operates on continuous variables. A simple model for an artificial neuron is

$$y = f\Big(\sum_{k=1}^{n} w_i u_i\Big), \tag{14.19}$$

where the parameters w_i are weights and the f is a sigmoid shaped function, for example $f(x) = \arctan x$ or lately the ReLU funtion $f(x) = \max(x,0)$.

A neural network is obtained by combining neurons in a network, that is typically layered, as shown in Figure 14.6. Neural networks can be used to represent functions of several variables. For example, Figure 14.6 represents the function $f : \mathbb{R}^5 \to \mathbb{R}^2$

$$f(y_1, y_2, \ldots, y_5) = g\Big(\sum_{i=1}^{5} w_{ij}^{(1)} g\big(\sum_{j=1}^{5} w_{ik}^{(2)} u_k\big)\Big), \tag{14.20}$$

where $w_{jk}^{(1)}$ and $w_{ij}^{(2)}$ are the weights in the first and second layers. An advantage with neural networks is that a function of many variables is represented by linear weights of nondecreasing functions of one variable. The usefulness of such approximations was demonstrated by a by Kolmogorov who showed that there exists fixed nondecreasing functions h_{ij} on $\mathbb{I} = [0,1]$ such that any continuous function $f : \mathbb{I}^{\ltimes} \to \mathbb{I}$ can be represented as

$$f(x_1, x_2, \ldots, x_n) = \sum_{i=1}^{2n+1} g_i\Big(\sum_{j=1}^{n} h_{ij}(x_j)\Big), \tag{14.21}$$

where g_i are continous functions of one variable.

(a) Traning a neural network (b) Training of an inverse model

Figure 14.23: Supervised training neural networks to obtain a process model (a) and the inverse model (b).

The parameters w_{iJ} of a single neuron (14.19) can be trained by the providing a series of values of input x and output y. Learning can be accomplished by feeding the systems known inputs and outputs as shown in Figure 14.6. A useful feature is that both the function and its invers can be generated from data.

The Hebb's rule

$$w_i(k+1) = w_i(k) + \gamma u_i^0(k)^0(y^0(k) - y(k)), \tag{14.22}$$

is one algorithm for updating the parameters in a single layer network. The argument k refers to the k:th experiment and the superscript ^{0} refers to the training data. This rule can be interpreted as an approximation of a gradient scheme for minimizing $\sum(y^0(k) - y(k))$, which is similar to to the MIT (14.16) for model reference adaptive control. Parameters of multi-layer neural networks can be updated by similar approximate schemes based on optimization.

Neural networks with many layers so-called deep learning have been shown to be very useful. Deep learning has proven very useful for recognition in computer vision REF??

Dynamic systems can be represented by combining neural networks as illustrated in Figure 14.24.

Positioning and Mapping

Assuring Safety of Complex Systems

- Lui Shah
- Guards
- Example Tore - KJ

Figure 14.24: Block diagram of nonlinear dynamical systems represented by two neural networks and integrators.

Autonomous Vehicles

One of the interesting areas of research in higher levels of decision-making is autonomous control of cars. Early experiments with autonomous driving were performed by Ernst Dickmanns, who in the 1980s equipped cars with cameras and other sensors [Dic07]. In 1994 his group demonstrated autonomous driving with human supervision on a highway near Paris and in 1995 one of his cars drove autonomously (with human supervision) from Munich to Copenhagen at speeds of up to 175 km/hour. The car was able to overtake other vehicles and change lanes automatically.

This application was explored anew through the DARPA Grand Challenge, a series of competitions sponsored by the U.S. government to build vehicles that can autonomously drive themselves in desert and urban environments. Over the course of approximately 4 years, hundreds of teams design and implemented autonomous vehicles, eventually demonstrating the ability to drive in both desert and urban environments. The capabilities demonstrated in the DARPA Grand Challenge are now being incorporated into advanced vehicle designs by the autonomotive industry, as well as other companies.

Caltech competed in the 2005 and 2007 Grand Challenges using a modified

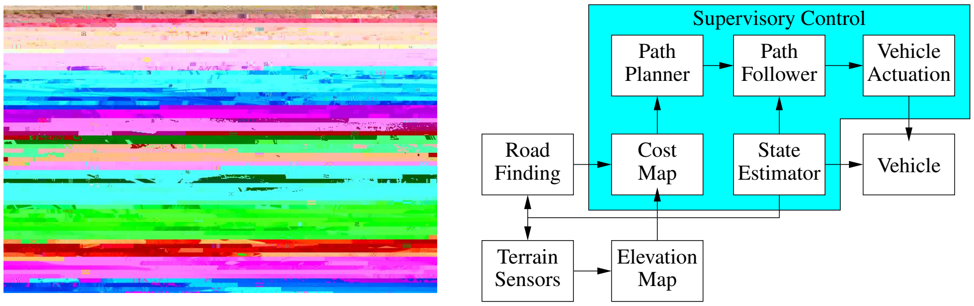


Figure 14.25: DARPA Grand Challenge. “Alice,” Team Caltech’s entry in the 2005 and 2007 competitions and its networked control architecture [CFG+06]. The feedback system fuses data from terrain sensors (cameras and laser range finders) to determine a digital elevation map. This map is used to compute the vehicle’s potential speed over the terrain, and an optimization-based path planner then commands a trajectory for the vehicle to follow. A supervisory control module performs higher-level tasks such as handling sensor and actuator failures.

Ford E-350 offroad van nicknamed “Alice.” It was fully automated, including electronically controlled steering, throttle, brakes, transmission and ignition. Its sensing systems included multiple video cameras scanning at 10–30 Hz, several laser ranging units scanning at 10 Hz and an inertial navigation package capable of providing position and orientation estimates at 5 ms temporal resolution. Computational resources included 12 high-speed servers connected together through a 1-Gb/s Ethernet switch. The vehicle is shown in Figure 14.25, along with a block diagram of its control architecture.

The software and hardware infrastructure that was developed enabled the vehicle to traverse long distances at substantial speeds. In testing, Alice drove itself more than 500 km in the Mojave Desert of California, with the ability to follow dirt roads and trails (if present) and avoid obstacles along the path. Speeds of more than 50 km/h were obtained in the fully autonomous mode. Substantial tuning of the algorithms was done during desert testing, in part because of the lack of systems-level design tools for systems of this level of complexity. Other competitors in the race (including Stanford, which won the 2005 competition) used algorithms for adaptive control and learning, increasing the capabilities of their systems in unknown environments. Together, the competitors in the Grand Challenge demonstrated some of the capabilities of the next generation of control systems and highlighted many research directions in control at higher levels of decision making.

System Architecture

A key element of our system is the use of a networked control systems (NCS) architecture that we developed in the first two grand challenge competitions. Building on the open source *Spread* group communications protocol, we have developed a modular software architecture that provides inter-computer communications be-

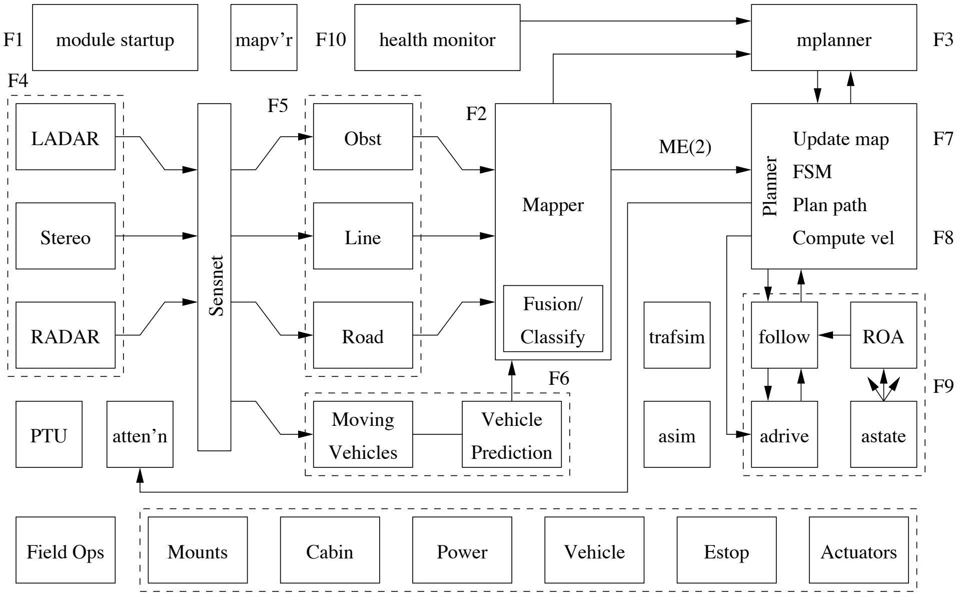


Figure 14.26: Systems architecture for operation of Alice in the 2007 Challenge. The sensing subsystem is responsible for building a representation of the local environment and passing this to the navigation subsystems, which computes and commands the motion of the vehicle. Additional functionality is provided for process and health management, along with data logging and simulation.

tween sets of linked processes [CFG+06]. This approach allows the use of significant amounts of distributed computing for sensor processing and optimization-based planning, as well as providing a very flexible backbone for building autonomous systems and fault tolerant computing systems. This architecture also allows us to include new components in a flexible way, including modules that make use of planning and sensing modules from the Jet Propulsion Laboratory (JPL).

A schematic of the high-level system architecture that we developed for the Urban Challenge is shown in Figure 14.26. This architecture shares the same underlying approach as the software used for the 2005 Grand Challenge, but with three new elements:

Canonical Software Architecture for mission and contingency management. The complexity and dynamic nature of the urban driving problem make centralized goal and contingency management impractical. For the navigation functions of our system, we have developed a decentralized approach where each module only communicates with the modules directly above and below it in the hierarchy. Each module is capable of handling the faults in its own domain, and anything the module is unable to handle is propagated “up the chain” until the correct level has been reached to resolve the fault or conflict. This architecture is described in more detail in Section **??** and builds on previous work at JPL [DRRS00, IRBM05, Ras01].†

RMM: Check to make sure these references are OK

Mapping and Situational Awareness. The sensing subsystem is responsible for maintaining both a detailed geometric model of the vehicle's environment, as well as a higher level representation of the environment around the vehicle, including knowledge of moving obstacles and road features. It associates sensed data with prior information and broadcasts a structured representation of the environment to the navigation subsystem. The mapping module maintains a vectorized representation of static and dynamic sensed obstacles, as well as detected lane lines, stop lines and waypoints. The map uses a 2.5 dimensional representation where the world is projected into a flat 2D plane, but individual elements may have some non-zero height. Each sensed element is tracked over time and when multiple sensors overlap in field of view, the elements are fused to improve robustness to false positives as well as overall accuracy. These methods are described in more detail in Section **??**.

Route, Traffic and Path Planning. The planning subsystem determines desired motion of the system, taking into account the current route network and mission goals, traffic patterns and driving rules and terrain features (including static obstacles). This subsystem is also responsible for predicting motion of moving obstacles, based on sensed data and road information, and for implementing defensive driving techniques. The planning problem is divided into three subproblems (route, traffic, and path planning) and implemented in separate modules. This decomposition was well-suited to implementation by a large development team since modules could be developed and tested using earlier revisions of the code base as well as using simulation environments. Additional details are provided in Section **??**.

14.7 Application Fields

Aerospace

Character of industry

- Often technology drivers: space missions
- A few dozen large companies both civil and military
- Very high safety standards. Has often driven technology
- Early adopter of simulation ans model based engineering
- FACE Future Airborn Capability Environment

Products and markets

- Military and commercial markets
- Commercial jetliners Boeing 787 Airbus 380 private jets, military, helicopters, missiles, satellites, UAVs
- 2014 Boeing delivered 723 2013: 648,Airbus 2013: 628 by the end of World War II USA produced 300 000 plances

Control algorithms and their use

- Early adopter of optimization, Kalman filter, state feedback, gain scheduling, some use of adaptive control, autonomy
- Optimal control emerged from early space flight
- Early experiments in adaptive control
- User interaction through changes of operating modes and setpoints

Special features and impact

- Extreme safety requirements
- The aerospace industry has often pioneered new control methods

Automotive

Character of industry

- Mass market
- Autosar

Products and markets

- 2011: 60M, 2015 over 72M, 2014: 90M dominated by 6 companies who made more than 50M cars
- Many suppliers of subsystem and services

Control algorithms and their use

- Started with engine control to meet California emission standards, grew rapidly
    - Separate ECUs for each function was originally used to reduce complexity
    - A car may have up to 100 ECUs strong effort to reduce the number
    - Standard car XX loops: engine control, cruise control, tranction control
    - collision avoidance
    - autonomous care
- Relatively simple control algorithms, special algorithms for ABS, collission avoidance
- Highly sophisticated systems for driver less cars
- Interesting: Climate control of cars, component manufacturers supply hardware and validated dynamics simulation models. Car manufacturers evaluate consequences of choosing different suppliers

- Simple PID with gain scheduling for ordinary cars, sophisticated software for autonomy
- User only changes modes and reference values set points
- The systems are upgraded when the car is serviced

Special features and impact

- Strong impact on the field because of the large numbers
- Drove semiconductor manufacturers to develop microcontrollers
- Strong collaboration on developing design tools: modelica, AUTOSAR, FMI

Types of ECU include Electronic/engine Control Module (ECM), Powertrain Control Module (PCM), Transmission Control Module (TCM), Brake Control Module (BCM or EBCM), Central Control Module (CCM), Central Timing Module (CTM), General Electronic Module (GEM), Body Control Module (BCM), Suspension Control Module (SCM), control unit, or control module. Taken together, these systems are sometimes referred to as the car's computer. (Technically there is no single computer but multiple ones.) Sometimes one assembly incorporates several of the individual control modules (PCM is often both engine and transmission)[1]

Some modern motor vehicles have up to 80 ECUs.

Power Systems

Special features

- Large integrated systems for generation and distribution of electric power
- Highly integrated system

Character of industry

- Large companies or government monopolies
- In the US there are close to 8000 power plants with 20000 individual generators. There were three integrated networks Eastern, Western and Texas which have recently bee combined. A system with massive parallelism.
- Control of individual stations very similar to process control uses similar equipment.

Products and markets

-

Control algorithms and their use

- Extensive use of optimization

- Coordination of production, frequency and voltage control
- Logic and switching for
- Systems based on logic and sequencing start up and shut down and for maitaining the net operational and protecting the equipment
- Control of individual power generators very similar to process control. There are non-minimum phase dynamics due to the pen-stock dynamics in hydroelectric plants and due to the shrink and swell effect in boilers.
- Massively parallel system proportional gains tells the degree an individual station participates in frequency control

Special features and impact

- Jump and rate emerged in the power industry
- Dramatic change in industry from a centralized highly regulated to distributed systems with smart grids.

Physical Experiments

Special features

- One of a kind: Synchrotrones, particle accelerator, spallation sources and LIGO
- Highly specialized groups of engineers and phycicists
- Nobel prize

Character of industry

- Unique government or international research groups

Products and markets

-

Control algorithms and their use

- Emergence of repetitive control

Special features

-

Impact

-

One hundred years after Albert Einstein predicted the existence of gravitational waves, scientists have finally spotted these elusive ripples in space-time.

In a highly anticipated announcement, physicists with the Advanced Laser Interferometer Gravitational-Wave Observatory (LIGO) revealed on 11 February that their twin detectors have heard the gravitational 'ringing' produced by the collision of two black holes about 400 megaparsecs (1.3 billion light-years) from Earth1, 2.

Ladies and gentlemen, we have detected gravitational waves, David Reitze, the executive director of the LIGO Laboratory, said at a Washington DC press conference. We did it!

Gravitational waves: 6 cosmic questions they can tackle One black hole was about 36 times the mass of the Sun, and the other was about 29 solar masses. As they spiralled inexorably into one another, they merged into a single, more-massive gravitational sink in space-time that weighed 62 solar masses, the LIGO team estimates.

Cruise controllers and other automotive control systems are and cellular phones are mass produced with billions of users.

Lots of things in between.

Process Control

Special features

-

Character of industry

- Nature of industry: Petrochemical, pulp and paper, chemical, pharmaceutical, ... share many properties with power plants.

Products and markets

- Large processes a few thousand actuators per process
- Factory may have tens of process units with a few hundred control loops

Control algorithms and their use

- Cascade, midrange an selectors
- Optimization both steady state and dynamic
- The DCS system
  - The DCCs is like a toolbox which has to be configured
  - Emerged from relay cabinettes and controller cabinets
  - Developed both from industrial users and equipment suppliers
  - Stadardization of communication from plastic tubes to Fieldbus

- Several categories of people interact with the system: process engineers, instrument engineers and operators
- Control loops are retuned systems sometimes reconfigured
- Techniques used: Overview from Jemima

Special features and impact

- Much of the development of PID controllers and their tuning (ZN)
- The Ziegler-Nichols tuning rules
- Model predictive control Cutler and Ramacher
- Maximum likelihood and prediction error methods
- Standards

Process control provides automtion for a variety of industries such as chenicals, oil refining, pulp and paper, chemicals, and pharmaceuric power plants and many others. It enables a few operators to run a complex process from central control rooms. Control and automation for process control is a 100 billion dollar industry. The distributed control system (DCS) is the standard tool to provide control, see Figure **??**. It has facilities for connecting sensors, actuators and algorithms and can be viewed as a tool-box for implementing control systems. The control algorithms are implemented by process and instrument engineers both by company personel and by consultants. Controller parameters can be modified during operation and the system is occasionally reconfigured. Algorithms and languages are standardized by international committees.

A typical installation has thousands of loops, most of them are PID controllers, cascade, selector and midranging control are common as are automatic tuning and model predictive control. A Japanese study some years ago gave the following percentages.

Process control and aerospace were applications where complex control systems emerged at an early stage. In process control it was customary to have one cabinet with analog controller for regulation and a relay cabinet with relay logic for startup, shut down and equipment protection. Valves are commonly used for actuation in process control. It is customary to have a feedback loop with a valve positioner to reduce effects of friction and nonlinearities at the lowest level of the hierarchy, and feedback loops for control of pressure and temperature at the next level. As technology developed the relays were replaced by programmable logic controller PLC and the analog controllers were replaced by distributed control systems DCS. The PLC's and DCS's had very different architecture, but since both were digital devices it was natural that controller functions were introduced in PLC's and logic functions in DCS systems. In process control the SCADA systems appeared ad the standard solution.

the control system for a large chemical process can have thousands of signals and actuators.

Telecommunication

Character of industry

- A few large operators and a few large companies, design highly centralized, lots of engineering users do not influence the system
- Mass market products
- Very large number of cellular phones
    - 1B smartphones in 2013?
    - 6.7 B users in 1914

Products and markets

- Control of individual stations and cellular phones

Control algorithms and their use

- Control of individual stations and cellular phones
- Very large number of cellular phones

Special features and impact

Sensors and Instruments

Consumer devices

-

Miscellaneous

Sensors instruments and consumer devices, radio, TV, appliances

14.8 Summary