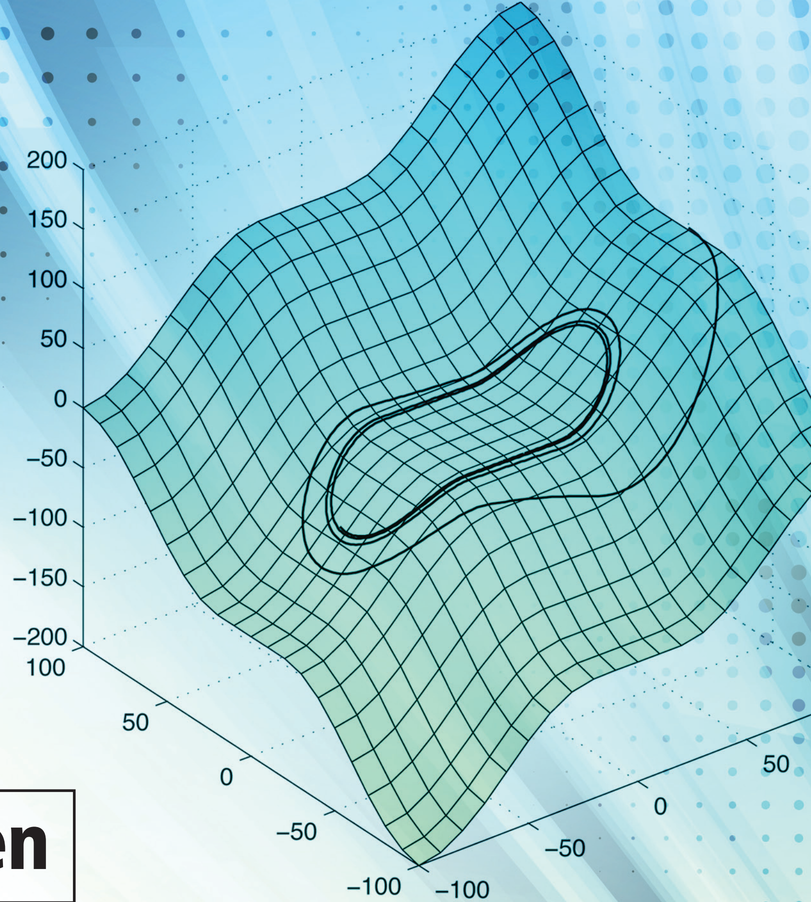

Jan Jantzen

Foundations of Fuzzy Control

A Practical Approach

Second Edition

WILEY

FOUNDATIONS OF FUZZY CONTROL

FOUNDATIONS OF FUZZY CONTROL

A PRACTICAL APPROACH

Second Edition

Jan Jantzen
University of the Aegean at Chios, Greece

WILEY

This edition published in 2013

First Edition published in 2007

Registered office
John Wiley & Sons Ltd, The Atrium, Southern Gate, Chichester, West Sussex, PO19 8SQ, United Kingdom

For details of our global editorial offices, for customer services and for information about how to apply for permission to reuse the copyright material in this book please see our website at www.wiley.com.

Library of Congress Cataloging-in-Publication Data

Jantzen, Jan.
Foundations of fuzzy control : a practical approach / Jan Jantzen.
1 online resource.
Includes bibliographical references and index.
Description based on print version record and CIP data provided by publisher; resource not viewed.
ISBN 978-1-118-53557-8 (MobiPocket) – ISBN 978-1-118-53558-5 (Adobe PDF) –
ISBN 978-1-118-53559-2 (ePub) – ISBN 978-1-118-50622-6 (hardback) 1. Automatic control.
2. Fuzzy systems. 3. Fuzzy automata. I. Title.
TJ213
629.8′312–dc23

2013023628

A catalogue record for this book is available from the British Library.

ISBN: 978-1-118-50622-6

Typeset in 10/12pt Times by Aptara Inc., New Delhi, India

1 2013

In memory of Ebrahim (Abe) Mamdani (1 Jun 1942–22 Jan 2010) and
Lauritz Peter Holmblad (23 Aug 1944–30 Mar 2005)

Figure 1 EH Mamdani (1942–2010)

Contents

The * in the heading denotes that the section can be skipped on the first reading as it contains background material catered for advanced readers of this book.

Foreword

Since the objective of Foundations of Fuzzy Control is to explain why fuzzy controllers behave the way they do, I would like to contribute a historical perspective.

Before the 1960s, a cement kiln operator controlled a cement kiln by looking into its hot end, the burning zone, and watching the smoke leaving the chimney. The operator used a blue glass to protect his eyes. He controlled the fuel/air ratio in order to achieve steady operation of the kiln.

Central control was introduced in the cement industry in the 1960s. PID controllers were installed, mainly for uniform feeding of the raw materials and the fuel. Computers for process supervision and control were introduced in the cement industry in the late 1960s.

During experimental work in the 1970s, the fuel control strategy was programmed as a two-dimensional decision table with an error signal and the change in error as inputs.

The first time we heard about fuzzy logic was at the fourth IFAC/IFIP International Conference on Digital Computer Applications to Process Control, held in Zürich, Switzerland, in 1974. As a postscript to a paper on learning controllers, Seto Assilian and Abe Mamdani proposed fuzzy logic as an alternative approach to human-like controllers.

Experimental work was carried out at the Technical University of Denmark. The theoretical understanding and inspiration in relation to process control was gained mainly from papers written by Lotfi Zadeh and Abe Mamdani, and control experiments were performed in laboratory-scale processes such as, for example, a small heat exchanger. The rule based approach that underlies the decision tables was also inspired by the instructions that we found in a textbook for cement kiln operators, which contained 27 basic rules for manual operation of a cement kiln.

The first experiments using a real cement kiln were carried out at the beginning of 1978 at an FL Smidth cement plant in Denmark. At this stage of the development work, the attitude of the management was sceptical, partly because of the strange name, 'fuzzy'. Other names were suggested, but eventually, with an increasing understanding by the management of the concept, it was decided to stay with the word fuzzy, a decision that has never been regretted since.

In 1980, FL Smidth launched the first commercial computer system for automatic kiln control based on fuzzy logic. To date, hundreds of kilns, mills and other processes have been equipped with high-level fuzzy control by FL Smidth and other suppliers of similar systems.

Jens-Jørgen Østergaard
FL-Soft, Copenhagen

Preface to the Second Edition

This second edition of Foundations of Fuzzy Control includes new chapters on gain scheduling, fuzzy modelling and demonstration examples. Fuzzy gain scheduling is a straightforward extension of the usual PID type fuzzy controllers in the sense that fuzzy rules can interpolate naturally between PID controllers. Broadly speaking, the concept of local fuzzy models is dual to fuzzy gain scheduling. The demonstration chapter includes five larger examples that can be used as teaching modules. Furthermore, the chapter on stability has been extended to include performance. The intent has been to reach farther than mere analysis, that is, to devise a design method that starts from specifications of performance. The book adopts a practical approach, which is reflected in the new subtitle, *A Practical Approach*.

The guiding principle has been to try to reach the bottom of the matter by means of geometry. Thus, the PID controller can be seen as an inner product. Together with viewpoints from adaptive control and the self-organizing controller, this has led to a set of tuning recommendations, where the starting point is a performance specification, namely, the desired settling time (Chapter 7). The tuning recommendations are applied to an unstable chemical reactor tank and for the control of the idle speed in a car engine, in order to test and demonstrate how it works (Chapter 10). Hopefully, the reader will find the second edition of the book even more fundamental and coherent than the first edition owing to the geometric approach.

My students requested more examples and illustrations, and this second edition tries to fulfil that wish. A simulator (Autopilot) was developed to illustrate concepts in nonlinear control, such as equilibria, and the tool can be used as a stand-alone teaching tool. The book contents have been reorganized, and each chapter consists now of two parts, clearly separated by a summary: the first part is intended for an introductory course, and the part after the summary is for an advanced course. The advanced part is also a research guideline for students who wish to write their thesis within fuzzy control.

I teach an introductory course on the Internet using one of the demonstration examples. Access to the course is through the companion website www.wiley.com/go/jantzen, which is devoted to this book. The website also contains downloadable material, such as the MATLAB® programs that produced the figures, lecture slides and error corrections.

Finally, I wish to acknowledge the inspiration and help I have received from Abe Mamdani, especially in connection with the idle speed project (Chapter 10). He died, much too early, in 2010, and he is sadly missed. This second edition is dedicated to him, as well as to Peter Holmblad – two giants in the history of fuzzy control.

Jan Jantzen
University of the Aegean at Chios, Greece

Preface to the First Edition

In summary, this textbook aims to explain the behaviour of fuzzy logic controllers. Under certain conditions a fuzzy controller is equivalent to a proportional-integral-derivative (PID) controller. The equivalence enables the use of analysis methods from linear and nonlinear control theory. In the linear domain, PID tuning methods and stability criteria can be transferred to linear fuzzy controllers. The Nyquist plot shows the robustness of different settings of the fuzzy gain parameters. As a result, a fuzzy controller can be guaranteed to perform as well as any PID controller. In the nonlinear domain, the stability of four standard control surfaces can be analysed by means of describing functions and Nyquist plots. The self-organizing controller (SOC) is shown to be a model reference adaptive controller. There is the possibility that a nonlinear fuzzy PID controller performs better than a linear PID controller, but there is no guarantee. Even though a fuzzy controller is nonlinear in general, and commonly built in a trial and error fashion, we can conclude that control theory does provide tools for explaining the behaviour of fuzzy control systems. Further studies are required, however, to find a design method such that a fuzzy control system exhibits a particular behaviour in accordance with a set of performance specifications.

Fuzzy control is an attempt to make computers understand natural language and behave like a human operator. The first laboratory application (mid-1970s) was a two-input-two-output steam engine controller by Ebrahim (Abe) Mamdani and Seto Assilian, UK, and the first industrial application was a controller for a cement kiln by Holmblad and Østergaard, FL Smidth, Denmark. Today there is a tendency to combine the technology with other techniques. Fuzzy control together with artificial neural networks provide both the natural language interface from fuzzy logic and the learning capabilities of neural networks. Lately hybrid systems, including machine learning and artificial intelligence methods, have increased the potential for intelligent systems.

As a follow-up to the pioneering work by Holmblad and Østergaard, which started at the Technical University of Denmark in the 1970s, I have taught fuzzy control over the Internet to students in more than 20 different countries since 1996. The course is primarily for graduate students, but senior undergraduates and PhD students also take the course. The material, a collection of downloadable lecture notes at 10–30 pages each, formed the basis for this textbook.

A fuzzy controller is in general nonlinear, therefore the design approach is commonly trial and error. The objective of this book is to explain the behaviour of fuzzy logic controllers, in order to reduce the amount of trial and error at the design phase.

Much material has been developed by applied mathematicians, especially with regard to stability analysis. Sophisticated mathematics is often required which unfortunately makes the material inaccessible to most of the students on the Internet course. On the other hand, application-oriented textbooks exist, easily accessible, and with a wide coverage of the area. The design approach is nevertheless still trial and error. The present book is positioned between mathematics and heuristics; it is a blend of control theory and trial and error methods. The key features of the book are summarized in the following four items.

- *Fundamental*. The chapter on fuzzy reasoning presents not only fuzzy logic, but also classical set theory, two-valued logic and two-valued rules of inference. The chapters concerning nonlinear fuzzy control rely on phase plane analysis, describing functions and model reference adaptive control. Thus, the book presents the parts of control theory that are the most likely candidates for a theoretical foundation for fuzzy control, it links fuzzy control concepts back to the established control theory and it presents new views of fuzzy control as a result.
- *Coherent.* The analogy with PID control is the starting point for the analytical treatment of fuzzy control, and it pervades the whole book. Fuzzy controllers can be designed, equivalent to a P controller, a PD controller, a PID controller or a PI controller. The PD control table is equivalent to a phase plane, and the stability of the nonlinear fuzzy controllers can be compared mutually, with their linear approximation acting as a reference. The self-organizing controller is an adaptive PD controller or PI controller. In fact, the title of the book could also have been *Fuzzy PID Control*.
- *Companion web site.*[1] Many figures in the book are programmed in MATLAB® (trademark of The MathWorks, Inc.), and the programs are available on the companion web site. For each such figure, the name of the program that produced the figure is appended in parentheses to the caption of the figure. They can be recognized by the syntax *.m, where the asterisk stands for the name of the program. The list of figures provides a key and an overview of the programs.
- *Companion Internet course*. The course concerns the control of an inverted pendulum problem or, more specifically, rule based control by means of fuzzy logic. The inverted pendulum is rich in content, and is therefore a good didactic vehicle for use in courses around the world. In this course, students design and tune a controller that balances a ball on top of a moving cart. The course is based on a simulator, which runs in the MATLAB® environment, and the case is used throughout the whole course. The course objectives are: to teach the basics of fuzzy control, to show how fuzzy logic is applied and to teach fuzzy controller design. The core means of communication is email, and the didactic method is email tutoring. An introductory course in automatic control is a prerequisite.

The introductory chapter of the book shows the design approach by means of an example. The book then presents set theory and logic as a basis for fuzzy logic and fuzzy reasoning, especially the so-called generalized modus ponens. A block diagram of controller components and a list of design choices lead to the conditions for obtaining a linear fuzzy controller, the prerequisite for the fuzzy PID controller.

[1] www.wiley.com/go/jantzen

The following step is into the nonlinear domain, where everything gets more difficult, but also more interesting. The methods of phase plane analysis, model reference adaptive control and describing functions provide a foundation for the design and fine-tuning of a nonlinear fuzzy PID controller.

The methods are demonstrated in a simulation of the inverted pendulum problem, the case study in the above-mentioned course on the Internet. Finally, a short chapter presents ideas for supervisory control based on experience in the process industry.

The book aims at an audience of senior undergraduates, first-year graduate students and practising control engineers. The book and the course assume that the student has an elementary background in linear differential equations and control theory, corresponding to an introductory course in automatic control. Chapters 1, 2, 3 and 9 can be read with few prerequisites, however. Chapter 4 requires knowledge of PID control and Laplace transforms and Chapters 5, 6 and 7 require more and more background knowledge. Even the simulation study in chapter 8 requires some knowledge of state-space modelling to be fully appreciated. Mathematical shortcuts have been taken to preserve simplicity and avoid formalism.

Sections marked by an asterisk (*) may be skipped on a first reading; they are either very mathematical or very practically oriented, and thus off the main track of the book.

It is of course impossible to cover in one volume the entire spectrum of topic areas. I have drawn the line between fuzzy control and neuro-fuzzy control. The latter encompasses topics such as neural networks, learning and model identification that could be included in a future edition.

Acknowledgements. I am pleased to acknowledge the many helpful suggestions I received from the late Lauritz Peter Holmblad, who acted as external supervisor on Masters projects at the Technical University of Denmark, and Jens-Jørgen Østergaard. They have contributed process knowledge, sound engineering solutions and a historical continuity. Thanks to Peer Martin Larsen, I inherited all the reports from the early days of fuzzy control at the university. I also had the opportunity to browse the archives of Abe Mamdani, then at Queen Mary College, London. I am also pleased to acknowledge the many helpful suggestions from Derek Atherton and Frank Evans, both in the UK, concerning nonlinear control, and in particular state-space analysis and describing functions. Last but not least, former and present students at the university and on the Internet have contributed collectively with ideas and suggestions.

Jan Jantzen
University of the Aegean at Chios

1

Introduction

Fuzzy control uses sentences, in the form of rules, to control a process. The controller can take many inputs, and the advantage of fuzzy control is the ability to include expert knowledge. The interface to the controller is more or less natural language, and that is what distinguishes fuzzy control from other control methods. It is generally a nonlinear controller. There are, however, very few design procedures in the nonlinear domain compared to the linear domain. This book proposes to stay as long as possible in the linear domain, on the solid foundations of linear control theory, before moving into the nonlinear domain with the design. The design method consists accordingly of four steps: design a PID controller, replace it with a linear fuzzy controller, make it nonlinear, and fine-tune the resulting controller. A nonlinear process may have several equilibrium points, and the local behaviour can be different from the behaviour far from an equilibrium, which makes it difficult to control. In order to demonstrate various aspects of nonlinear control, the book uses a simulator of a train car on a hilly track.

Fuzzy controllers appear in consumer products such as washing machines, video cameras, and cars. Industrial applications include cement kilns, underground trains, and robots. A *fuzzy controller* is an *automatic controller*, that is, a self-acting or self-regulating mechanism that controls an object in accordance with a desired behaviour. The object can be, for instance, a robot set to follow a certain path. A fuzzy controller acts or regulates by means of rules in a more or less natural language, based on the distinguishing feature: fuzzy logic. The rules are invented by plant operators or design engineers, and fuzzy control is thus a branch of artificial intelligence.

1.1 What Is Fuzzy Control?

Conventionally, computer programs make rigid *yes* or *no* decisions by means of decision rules based on two-valued logic: true/false, yes/no, or one/zero. An example is an air conditioner with a thermostatic controller that recognizes just two states: above the desired temperature or below the desired temperature. *Fuzzy logic*, on the other hand, allows intermediate truth-values between true and false.

Foundations of Fuzzy Control: A Practical Approach, Second Edition. Jan Jantzen.

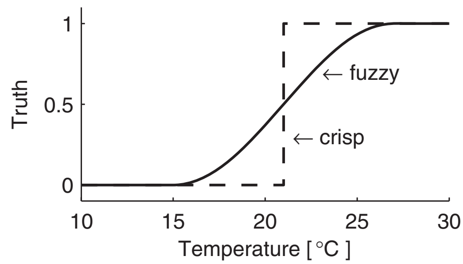

Figure 1.1 A warm room. The crisp air conditioner considers any temperature above 21°C warm. The fuzzy air conditioner considers gradually warmer temperatures. (figwarm.m)

A fuzzy air conditioner may thus recognize 'warm' and 'cold' room temperatures. The rules behind are less precise, for instance:

- *Rule*. If the room temperature is warm and slightly increasing, then increase the cooling.

Many classes or *sets* have *fuzzy* rather than sharp boundaries, and this is the mathematical basis of fuzzy logic: the set of 'warm' temperature measurements is one example of a fuzzy set.

The core of a fuzzy controller is a collection of *linguistic* (*verbal*) rules of the *if–then* form. Several variables may appear in each rule, both on the *if* side and on the *then* side. The rules can bring the reasoning used by computers closer to that of human beings.

In the example of the fuzzy air conditioner, the controller works on the basis of a temperature measurement. The room temperature is just a number, and more information is necessary to decide whether the room is warm. Therefore, the designer must incorporate a human being's perception of warm room temperatures. A straightforward approach is to evaluate beforehand all possible temperature measurements. For example, on a scale from 0 to 1, truly *warm* corresponds to 1 and definitely *not warm* corresponds to 0,

Grades of *warm*	0.0	0.0	0.4	0.9	1.0
Temperature (°C)	10	15	20	25	30

This example uses *discrete* temperature measurements, whereas Figure 1.1 shows the same idea graphically, in the form of a *continuous* mapping of temperature measurements to truth-values. The mapping is arbitrary, that is, based on preference, not mathematical reason.

1.2 Why Fuzzy Control?

If PID control (proportional-integral-derivative control) is inadequate – for example, in the case of higher-order processes, systems with a long deadtime, or systems with oscillatory modes (Åström and Hägglund 2006) – fuzzy control is an option. But first, let us consider why one would not use a fuzzy controller:

- The PID controller is well understood, easy to implement – both in its digital and analogue forms – and it is widely used. By contrast, the fuzzy controller requires some knowledge of fuzzy logic. It also involves building arbitrary membership functions.

- The fuzzy controller is generally nonlinear. It does not have a simple equation like the PID, and it is more difficult to analyse mathematically; approximations are required, and it follows that stability is more difficult to guarantee.
- The fuzzy controller has more tuning parameters than the PID controller. Furthermore, it is difficult to trace the data flow during execution, which makes error correction more difficult.

On the other hand, fuzzy controllers are used in industry with success. There are several possible reasons:

- Since the control strategy consists of *if–then* rules, it is easy for a process operator to read. The rules can be built from a vocabulary containing everyday words such as ‘high’, ‘low’, and ‘increasing’. Process operators can embed their experience directly.
- The fuzzy controller can accommodate many inputs and many outputs. Variables can be combined in an *if–then* rule with the connectives *and* and *or*. Rules are executed in parallel, implying a recommended action from each. The recommendations may be in conflict, but the controller resolves conflicts.

Fuzzy logic enables non-specialists to design control systems, and this may be the main reason for its success.

1.3 Controller Design

Established design methods such as pole placement, optimal control, and frequency response shaping only apply to linear systems, whereas fuzzy control is generally nonlinear. Since our knowledge of the behaviour of nonlinear systems is limited, compared with the situation in the linear domain, this book is based on a design procedure founded on linear control:

1. Design a PID controller.
2. Replace it with a linear fuzzy controller.
3. Make it nonlinear.
4. Fine-tune it.

The idea is to exploit the design methods within PID control and carry them forward to fuzzy control. The design procedure is feasible because it is possible to build a linear fuzzy controller that functions exactly as any PID controller does. The following example introduces the design procedure.

1.4 Introductory Example: Stopping a Car

Assume that we are to design a controller that automatically stops a car in front of a red stop light, as a part of future safety equipment. Figure 1.2 illustrates the situation, and it defines the symbols for the brake force (F), the mass of the car (m), and its position (y). Assume also that we can only apply the brakes, not the accelerator pedal, in order to keep the example simple. Even though the example is simple, it is representative; think of parking a robot in a charging dock, parking a ferry at the quay, or stopping a driver-less metro train at a station.

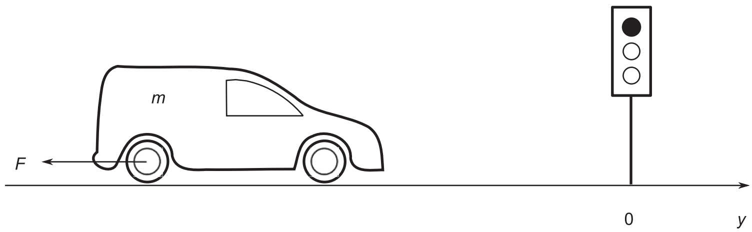

Figure 1.2 Stopping a car. The position y is positive towards the right, with zero at the stop light. The brakes act with a negative force F on the mass m.

Figure 1.3 shows a simulation model in Simulink (trademark of The MathWorks, Inc.). The block diagram includes a limiter block on the brake force, and the model is therefore nonlinear.

Step 1: Design a PID controller

Our first attempt is to try a proportional (P) controller,

$$F = K_p e \tag{1.1}$$

where K_p is the *proportional gain*, which can be adjusted to achieve the best response. The error $e \geq 0$ is the position error measured from the reference point *Ref* to the current position $y \leq 0$, that is,

$$e = Ref - y \tag{1.2}$$

Since y is negative and $Ref = 0$, then e is positive. But K_p is also positive, and the P controller in Equation (1.1) would demand a positive force F – in other words, acceleration by means of the accelerator pedal. The problem definition above ruled out the accelerator pedal, however, and we can conclude that a proportional controller is inadequate.

Our second attempt is to apply a proportional-derivative (PD) controller, since it includes a prediction. The controller is

$$F = K_p (e + T_d \dot{e}) \tag{1.3}$$

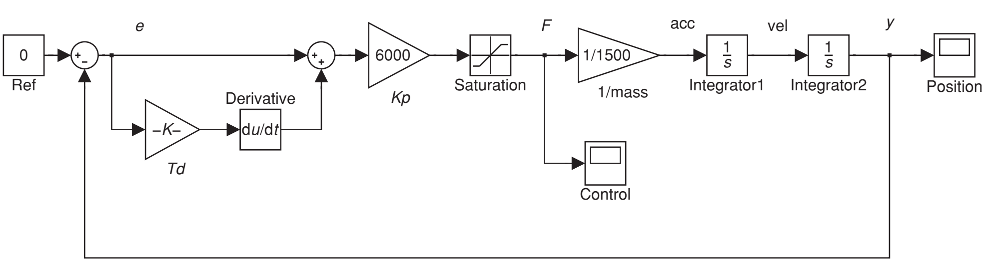

Figure 1.3 Simulink block diagram. A PD controller brakes the car from initial conditions $y(0) = -15$ and $\dot{y}(0) = 10$. (figcarpd.mdl)

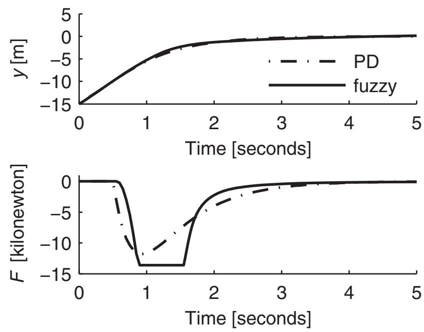

Figure 1.4 Stopping a car. Comparison between a PD controller and a fuzzy controller. (figstopcar.m)

where T_d is the *derivative gain*, which can be adjusted. Now the control signal is proportional to the term $e + T_d\dot{e}$ which is the predicted error T_d seconds ahead of the current error e. Compared to the P controller, the PD controller calls for extra brake force when the velocity is high. Figure 1.4 shows the response and the brake force with

$$K_p = 6000$$
$$T_d = 1$$

During the first 0.5 s, the control signal is zero. Thereafter the derivative action takes over and starts to brake the car. In other words, the controller waits 0.5 s until it kicks in, it quickly increases the braking force, and after about 1 s it relaxes the brake gently. It takes about 5 s to stop the car.

We *tuned* the gains K_p and T_d in order to achieve a good closed loop performance. Hand tuning is possible, but it generally requires patience and a good sense of how the system responds. It is easier to use rules, for example the *Ziegler–Nichols tuning rules*. Although the rules often result in less than optimal settings, they are a good starting point for a manual fine tuning.

Step 2: Replace it with a linear fuzzy controller

A fuzzy controller consists of *if–then* rules describing the action to be taken in various situations. We will consider the situations where the distance to the stop light is long or short, and situations where the car is approaching fast or slowly. The linguistic terms must be specified precisely for a computer to execute the rules.

The following chapters will show how to design a *linear* fuzzy controller, with a performance that is exactly the same as the PD controller in the previous step. It is a design aid, because the PD controller, with its tuning, settles many design choices for the fuzzy controller. One requirement is that the membership functions should be linear.

At the end of this step, we have a fuzzy controller, with a response (not shown) exactly as the PD response in Figure 1.4.

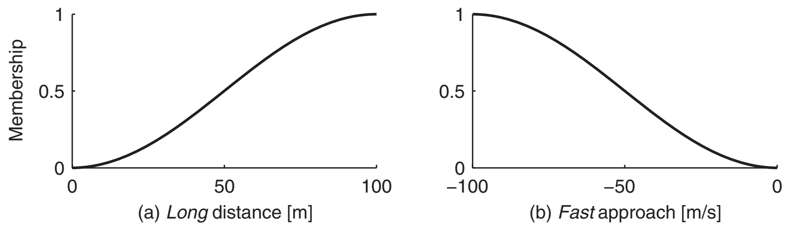

Figure 1.5 Fuzzy membership functions. Curve (a) is related to the distance from the stop light, and curve (b) specifies what is meant by a *fast* approach. The two curves are components of the rule: If distance is long and approach is fast, then brake zero. (figmfcar.m)

Step 3: Make it nonlinear

A complete rule base of all possible input combinations contains four rules:

If distance is long and approach is fast, then brake zero (1.4)

If distance is long and approach is slow, then brake zero (1.5)

If distance is short and approach is fast, then brake hard (1.6)

If distance is short and approach is slow, then brake zero (1.7)

The linguistic terms must be specified precisely for a computer to execute the rules. Figure 1.5 shows how to implement 'long', as in 'distance is long'. It is a fuzzy *membership function*, shaped like the letter *s*. The horizontal axis is the *universe*, which is the interval [0, 100]% of the full range of 15 m. The vertical axis is the *membership grade*, that is, how compatible a distance measurement is with our perception of the term 'long'. For instance, a distance of 15 m (100%) has membership 1, because the distance is definitely long, while half that distance is long to a degree of just 0.5. Note that the horizontal axis corresponds to the previously defined error e, scaled onto a standard range relative to the maximum distance.

The term 'fast', as in 'approach is fast', is another membership function. The horizontal axis is again percentages of full range (10 m/s), but the numbers are negative to emphasize that the distance is decreasing rather than increasing. The horizontal axis corresponds to the previously defined time derivative $\dot{e}$ scaled onto the universe. The -100% corresponds to the maximum speed of 10 m/s. Similarly, the membership function for 'short' is just a mirror image of the membership function 'long', and the membership function 'slow' is just a mirror image of 'fast'.

Turning to the *then*-side of the rules, the term 'zero' means to apply the brake force $F = 0$. The term 'hard' is the full brake force of -100%.

The nonlinear domain is poorly understood in general, and it usually calls for a trial and error design procedure. Nevertheless, the following chapters provide methods such that at least some *analysis* is possible.

Step 4: Fine-tune it

Figure 1.4 shows the response with the nonlinear controller, together with the initial PD response, after adjusting one tuning factor (input gain on the error, GE). The response is close to the initial PD response, but a little faster. The lower plot with the control signals shows the difference: the fuzzy controller waits longer before it kicks in, then it uses all the available brake force, and thereafter it releases the brake quicker than the PD controller.

The behaviour is not necessarily better than PD control. But since the fuzzy controller in step 2 is guaranteed to perform the same way, it is safe to say that the fuzzy controller is at least as good. Whether it performs better after steps 3 and 4 is an open question, but at least the fuzzy controller provides extra options to shape the control signal. This could be important if passenger comfort has a high priority.

Example 1.1 *Tuning by means of process knowledge*

Is it possible to use a mathematical model to find optimal settings for the PD controller?

► *Solution*

Disregarding engine dynamics, skidding, slip, and friction – other than the frictional forces in the brake pads – the force F causes an acceleration a according to Newton's second law of motion $F = ma$. Acceleration is the derivative of the velocity which in turn is the derivative of the position. Thus $a = \ddot{y}$, where the dots are Newton's dot notation for the differentiation operator $\mathrm{d}/\mathrm{d}t$. *We can rewrite the differential equation that governs the motion of the car as*

$$F = m\ddot{y} \Leftrightarrow \ddot{y} = \frac{F}{m} \tag{1.8}$$

For a Volkswagen Caddy Van (diesel, 2-L engine) the mass, without load and including the driver, is approximately 1500 kg. *Assume that the stop light changes to red when the car is* 15 m *(*49 ft*) away at a speed of* 10 m/s *(*36 km/h *or* 23 mph*). We have thus identified the following constants:*

$$\begin{aligned} m &= 1500 \\ y(0) &= -15 \\ \dot{y}(0) &= 10 \end{aligned}$$

Here $y(0)$ means the initial position, that is $y(t)$ at time $t = 0$, and $\dot{y}(0)$ is the initial speed. The force F arises not from the engine, but from an opposite friction force in the brakes, and it is directed in the negative direction. Since the brake is our only means of control, the control signal F is constrained to the interval

$$-13\,600 \le F \le 0 \tag{1.9}$$

This can be seen as follows. According to its data sheet, the car requires at least 27.3 m *to stop when driving at a speed of* 80 km/h. *As all the kinetic energy is converted to work, we have, on the average,*

$$\frac{1}{2} m \left(\dot{y}\right)^2 = F y$$

and thus

$$\begin{aligned}|F| &= \frac{1}{y}\frac{1}{2}m(\dot{y})^2 \\ &= \frac{1}{27.3}\frac{1}{2}1500\left(\frac{80\,000}{3600}\right)^2 \\ &\approx 13\,600\end{aligned}$$

We therefore assume that the anti-lock braking system limits the magnitude of the brake force to 13 600 *N (newton).*

The closed loop characteristic equation *is obtained by inserting Equation (1.3) into Equation (1.8):*

$$\ddot{y} = \frac{K_p\,(e + T_d\dot{e})}{m} = -\frac{K_pT_d}{m}\dot{y} - \frac{K_p}{m}y \tag{1.10}$$

There will be a steady state solution, since insertion of $\ddot{y} = \dot{y} = 0$ *yields the solution* $y = 0$; *this is just a check that a solution in accordance with the problem definition is feasible.*

Disregarding the nonlinearity, the transfer function in the Laplace domain is the forward path gain in the block diagram divided by 1 minus the loop gain (Mason's rule),

$$\begin{aligned}\frac{y(s)}{Ref} &= \frac{K_p\,(1 + T_ds)\,\frac{1}{m}\frac{1}{s^2}}{1 + K_p\,(1 + T_ds)\,\frac{1}{m}\frac{1}{s^2}} \\ &= \frac{\frac{K_p}{m}T_ds + \frac{K_p}{m}}{s^2 + \frac{K_p}{m}T_ds + \frac{K_p}{m}}\end{aligned} \tag{1.11}$$

The denominator is the closed loop characteristic polynomial, *compare Equation (1.10), and it is a second-order polynomial in s. The general transfer function of a second-order system is*

$$T = \frac{\omega_n^2}{s^2 + 2\zeta\omega_ns + \omega_n^2} \tag{1.12}$$

Here ω_n *is the* natural frequency *– the frequency of oscillation without damping – and* ζ *is the* damping ratio. *It is very useful here, because we are looking for the response without overshoot, which is as fast as possible. This is the case when* $\zeta = 1$, *which yields a* critically damped *response. Comparing with Equation (1.11) our damping ratio is*

$$\zeta = \frac{1}{2}\sqrt{\frac{K_p}{m}}T_d$$