In this issue we bring you reviews of four books covering diverse aspects and applications of advanced control. The review by Jankovic of Guzzella and Onders' *Introduction to Modeling and Control of Internal Combustion Engine Systems* discusses the numerous issues that arise in controlling modern engine systems. The next review, by Jadbabaie, provides an overview of convex optimization and its applications, the subject of a recent book by Boyd and Vandenberghe. The third review, by Sarangapani, of *Neural Engineering* by Eliasmith and Anderson, discusses the control engineering applications of concepts from this biologically inspired area of research. Finally, the review by Borkar of Yin and Zhang's *Discrete Markov Chains* explains the intricacies of two-time-scale models.

***—Kirsten Morris***

---

***Introduction to Modeling and Control of Internal Combustion Engine Systems*** by L. Guzzella and C.H. Onder, Springer-Verlag, 2004, ISBN: 354022274X, US$79.95. *Reviewed by Mrdjan J. Jankovic.*

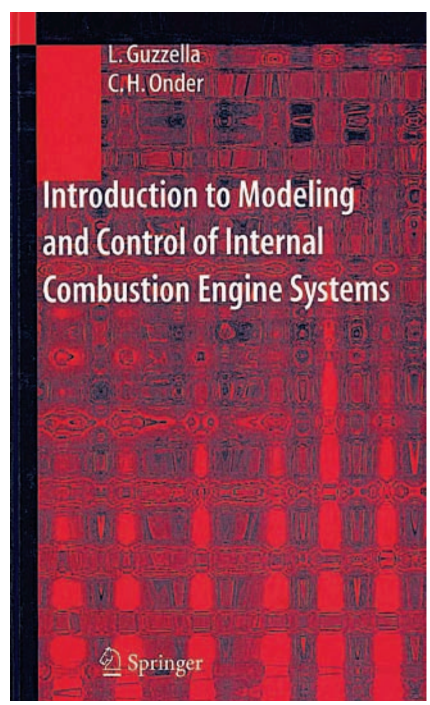
## The Book

The topic of this book is modeling and control of internal combustion engines for automotive applications. With global automotive annual sales of around 60 million vehicles, internal combustion engines impact the world's oil consumption and local air quality. In developed markets such as the United States, government regulations on tailpipe emissions and fuel consumption have been in existence for more than 30 years. Regulations and customer preferences have induced significant developments in engine hardware and exhaust gas after-treatment systems. These advances have been matched by progress in the theory and practice of engine control. For example, achieving tailpipe emission reduction of two orders of magnitude is made possible by catalytic converters that treat the engine exhaust gas. Efficient operation of these after-treatment systems critically depends on accurate management of the in-cylinder air-fuel ratio, which is accomplished by increasingly sophisticated control systems. To facilitate the production implementation of evermore complex control systems, the computing power and memory size of powertrain electronic control units is increasing by about an order of magnitude per decade [4].

On the fuel economy side, various "tuning" devices, such as the exhaust gas recirculation (EGR) system, variable valve timing and lift, and variable displacement (also called displacement-on-demand), have been added to reduce fuel consumption. In its base configuration, a spark ignition (gasoline) engine needs to control only air, fuel, and spark timing to respond to the driver's acceleration demand and meet fairly strict emission standards. In the process of base engine hardware design, parameters such as valve timing—the relative phase between the crankshaft, the main shaft that rotates powered by the engine pistons, and the camshafts, which control opening and closing of the intake and exhaust valves—are fixed at values that provide a reasonable compromise between operating conditions. By allowing these parameters to vary with the operating condition, additional fuel economy and performance (torque and power) improvements can be achieved. Similarly, in a diesel engine, the optimized parameters may include the air-fuel ratio, percent of recirculated exhaust gas, fractions of fuel in multiple fuel-injection pulses, and dwell times between pulses.

The set points for the optimized parameters are not known a priori. These values are obtained after an elaborate and time-consuming mapping, optimization, and calibration (fine-tuning) process. The task of the control system is to regulate the optimized parameters to set points that vary with operating conditions and to remove undesirable transient side effects. Online determinations of the set points and "decoupling" of the effects rely substantially on feedforward control (see [1] for an example). In addition, many set-point regulation control loops in an engine are affected by transport delays, which reduce the effectiveness of the feedback and necessitate the use of a feedforward component. For feedforward compensation, the availability of accurate control-oriented models is essential. This fact provides a solid justification for the material selection by the authors. More than half of *Introduction to Modeling and Control of Internal Combustion Engine Systems* is devoted to engine subsystem modeling.

Despite the rich journal and conference proceedings literature on engine controls, there are few texts that provide a broad, introductory exposition to the subject. The only other textbook that this reviewer is aware of is [2], which presents a broader set of automotive control problems, including not only engine




1066-033X/05/$20.00©2005IEEE

controls but also the driveline and vehicle dynamics. In fact, most of the material in [2] is devoted to the last two topics. Thus, the book by Guzzella and Onder can and does provide a more detailed account of engine modeling and control. However, persons interested in this subject can benefit from reading both texts because there is less overlap than one would expect.

## Contents of the Book

The book is divided into three parts. The first part provides a brief introduction to the engine control problem, while the two remaining parts focus on the modeling of engine and after-treatment subsystems as well as control design.

## Models of Engine Subsystems

The modeling part of the book is divided into two chapters according to the nature of the engine models considered. Chapter 2 considers *mean value models* (MVM), which are control-oriented models that neglect the discrete nature of the engine cycle and consider evolution of variables (states) to be continuous in an average sense over the cycle. In contrast, a *discrete-event model* (DEM) of an engine system explicitly takes into account the discrete engine events. DEMs are considered in chapter 3.

MVMs are the staple of control-oriented engine modeling. Chapter 2 provides an in-depth account of such models for diesel and gasoline engines. Models of air and fuel delivery, torque generation, thermal systems, pollution formation, and after-treatment are discussed in detail. When the discrete nature of the engine cycle plays a crucial role in understanding a phenomenon or an effect, DEMs are more appropriate. Chapter 3 presents DEMs of mean-torque production, air and fuel flows, in-cylinder residual gas, and in-cylinder pressure.

The air system MVMs, which are fairly standard, include the manifold filling dynamics derived from the ideal gas law $PV = mRT$, the throttle (sharp edge orifice) flow model, and the engine volumetric efficiency. Although a formula for the volumetric efficiency is provided by (2.22), in this reviewer's experience, it is more likely that a characterization is obtained based on experimental engine mapping data. The bonus of this section (section 2.3) is a detailed turbocharger model that contains the Moore-Greitzer compressor surge phenomenon. The corresponding DEM models of chapter 3 include a standard DEM for mass airflow into the cylinders (section 3.2.2). For compressed natural gas engines, the effects of a stratified (that is, nonhomogeneous) air-fuel mixture and the back flow of gas into the port caused by late intake valve closing are also included (section 3.2.4).

The fuel system model is focused on the wall-wetting effect, that is, the propensity of injected fuel to stick to the intake port walls (and intake valve surface if the engine is cold) and form puddles that evaporate at a certain rate. Because of puddle formation and evaporation, the mass of fuel that ends up in the cylinder is different from the mass of injected fuel. For accurate air-fuel ratio control, it is important to model these effects and provide compensation. The first principles model in section 2.4 is based on new results by the second author [3]. From the control point of view, it is important to track wall wetting cylinder by cylinder. A DEM of the fuel flow dynamics, based on multiplexing the continuous-time dynamics, is presented in section 3.2.3.

The torque generation section (section 2.5), which provides ample illustration of the problem of engine optimization, shows that the torque depends on most engine variables (such as speed, manifold pressure, spark timing, air-fuel ratio, and EGR fraction). Thermal system models (section 2.6) are relevant because of the desire to achieve quick light-off of catalytic converters after a cold start and maintain their temperature in the desired range.

I found the pollutant formation and abatement sections (sections 2.7 and 2.8) to be particularly informative. Formation of the three main regulated gases, namely, CO, non-methane organic gases (NMOG) (also called hydrocarbons), and nitrogen oxide (NOx), is discussed in detail. The three-way catalysts, combined with the control system that regulates air-fuel ratio at stoichiometry (defined as the ratio of air and fuel with just enough oxygen for complete fuel combustion—equal to about 14.6 for gasoline), achieve emission reduction for all three gases by a factor ranging from 20 to over 100. Diesel engines cannot run with a stoichiometric (or rich) air-fuel ratio because such operation results in high particulate matter emissions and visible smoke. Thus, with conventional three-way catalysts, the reduction of NOx is not achieved due to excess oxygen in the exhaust gas. Instead, a selective catalytic reduction (SCR) system that uses urea to break down NOx can be employed. The book presents a model of an SCR system.

Finally, DEMs of in-cylinder phenomena, the residual gas dynamics, and in-cylinder pressure, can be found in section 3.3. This section shows how to estimate the burned-mass fraction (BMF), as a function of crank-angle degree, from the in-cylinder pressure signal. The BMF is relevant for combustion development since the location of 50% BMF correlates well with the maximum brake torque (MBT) spark timing, the timing that achieves the best fuel consumption.

## Engine Subsystem Control

The second part of the book, chapter 4 and appendices A and B, is devoted to control design. There are several different modes of operation for IC engines during the typical drive. These modes include cranking, cold-start emission reduction (CSER), idle, normal drive, and speed control. Each mode is defined by a different objective or set of objectives. For example, in the CSER mode, the objective is to achieve catalyst light-off in a short time, possibly at the expense of fuel economy. In the drive mode, the engine produces torque to respond to the driver's demand (and concurrently optimizes fuel consumption and emissions), whereas, in the idle mode, the goal is to regulate the engine speed to a desired set point. In each mode, many engine subsystems, which require local or coordinated control to achieve the objective, are active.

In addition, an engine control system has to provide monitoring, diagnostics, and failure mode management functionality that are either required by regulation or needed to ensure capability and availability. Presenting a description of each of these subsystems in a book of this format is not desirable or even feasible. Instead, in Chapter 4 the authors present four engine control design case studies: air-fuel ratio control, multivariable engine speed and air-fuel ratio control, SCR system control, and thermal management. In all four cases, the controller performance is evaluated experimentally.

The air-fuel ratio regulation (section 4.2) is a typical example of an engine control system design. The objective is to maintain the air-fuel ratio at stoichiometry to ensure high efficiency of the catalytic converter. The controller contains feedforward components, specifically, an air-charge estimator, and a compensator for fuel wall-wetting dynamics. Improved emission performance is achieved if the model of catalyst oxygen storage is included in the control system design. The feedback signal is provided by either a switching or a wide-range (continuous) exhaust oxygen sensor. The feedback component is either a conventional PI controller or a more complex controller obtained using $H_\infty$ design.

The second example is $H_\infty$ design for a multivariable control system that uses the throttle, fuel injectors, and spark timing to regulate engine speed in the idle mode as well as the air-fuel ratio. To use spark timing as an actuator, the base timing must be centered away from the MBT point to provide two-sided actuation. Although this setup entails a fuel economy penalty, this practice is standard in the industry because of the importance of high-quality idle speed control. The same caveat applies to fuel if it is used for speed control; with the nominal air-fuel ratio set point at stoichiometry (for best emissions), the fuel cannot increase the torque, and thus engine speed. This point illustrates the nonlinear nature of an internal combustion engine and difficulties that arise in applying linear control techniques.

The last two examples of Chapter 4 are control of the selective reduction system in a diesel engine and a thermal management system. In both cases, the controller has the familiar feedback-feedforward structure.

Two appendices conclude the book. Appendix A provides a broad but not very detailed review of modeling and control of dynamical systems. The material touches on various issues that an engine control designer must take into account. Appendix B presents a control design case study of an idle speed controller, this time as a single-input, single-output system. The subtopics presented are the synthesis of the mathematical model, system identification, model linearization, approximation of the delay with Padé approximation, and design of the control system for the air-path using observer-based pole placement.

## Conclusions

This book presents a selection of topics of high relevance to engine control design. The text investigates modeling of physical and chemical processes and discusses selection of the control architecture and design of the feedback gains for several important control loops. The diversity of the material clearly illustrates the multidisciplinary nature of the subject and the steep learning curve faced by a newcomer to the field.

An internal combustion engine is a difficult system to control. In many cases, the behavior is nonlinear (see, for example, the throttle flow function in figure 2.9). Some of the control variables, such as spark timing, are actually optimized parameters, which means that a deviation in either direction from its optimal set point changes the system output in the same direction. In such a situation, linearization around the set point may not produce a useful model. In addition, in most control loops, there is a significant delay between the control input and the measured output. These facts make the feedforward component a major factor in the control design. In my experience, most of the effort for a typical engine subsystem control design is devoted to the modeling required for the feedforward component. Indeed, a major portion of this book is devoted to engine subsystem modeling, and I believe this selection is right on target.

Overall, the examples are representative of a typical approach to internal combustion engine control design. If I were to pick a weakness in the presentation, I'd point to the $H_\infty$ design, which is not described in much detail and is not included in the control design overview in appendix A. I suspect that only a fraction of people interested in this subject may have had exposure to such advanced control material.

In summary, this book is an essential text for anyone interested in engine control design. It seems appropriate for a graduate-level course, in particular, for students with some control background. According to the authors, the book is intended for students interested in classical and novel engine control systems. I would like to add that engine control practitioners can also learn a lot from this book, especially those practitioners who have expertise in only a subset of engine control problems.

## References


[1] M. Jankovic, S.W. Magner, S. Hsieh, and J. Koncsol, "Transient effects and torque control of engines with variable CAM timing," in *Proc. American Control Conf.*, Chicago, IL, June 2000, pp. 50–54.

[2] U. Kiencke and L. Nielsen, *Automotive Control Systems for Engine, Driveline, and Vehicle*. Berlin-Heidelberg: Springer-Verlag, 2000.

[3] K. Locatelli, C.H. Onder, and H. Geering, "A rapidly tunable wall-wetting model for SI engine control," in *Proc. SAE World Congress and Exhibition,* 2004, paper 2004-01-1469.

[4] T. Ueda and A. Ohata, "Trends of future powertrain development and evolution of powertrain control systems," in *Proc. SAE World Congress and Exhibition,* 2004, paper 2004-21-0063.


***Mrdjan Jankovic*** received his B.S. from Belgrade University (1986) and master's and doctoral degrees from Washington University in St. Louis (1989 and 1992). He held postdoctoral positions with Washington University and the University of California, Santa Barbara. He joined Ford Research Laboratory in 1995, where he is a technical leader in the Powertrain Controls R&A Department. His responsibilities include project management, mentoring and supervision of technical staff, and direct technical contribution to development of engine control systems. His research interests include automotive engine optimization and control, nonlinear control, and time-delay systems. He is a coauthor of *Constructive Nonlinear Control* (Springer-Verlag, 1997) and holds more than 20 U.S. patents. He has served on the editorial board of *IEEE Transactions on Control Systems Technology* since 1997. He received the Ford Research Technical Achievement Award (2001) and IEEE TCST Outstanding Paper Award (2002). He is a Fellow of the IEEE.

---

***Convex Optimization*** by Stephen Boyd and Lieven Vandenberghe, Cambridge University Press, 2004, 716 pp., ISBN: 0-521-83378-7, US$65.00. *Reviewed by Ali Jadbabaie.*

## Development of Convex Optimization

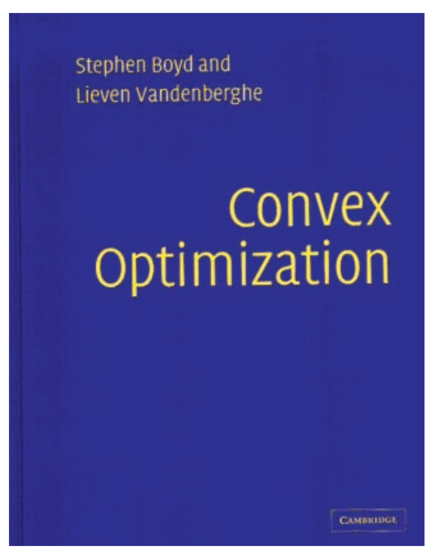
One of the most interesting developments in systems and control theory in the early 1990s was the observation that various problems related to analysis and synthesis of robust and nonlinear control systems could be formulated as optimization problems involving a linear function over a set of matrix inequalities. A second, related development was the realization that such problems could be solved efficiently. Even now, after about 15 years, one can hardly open up a control theory journal and not see a reference to a linear matrix inequality (LMI). As an example of the ubiquity of LMIs in control theory, consider the most basic property of a linear time-invariant (LTI) system, that is, its stability. In 1890, A.M. Lyapunov showed, in his doctoral thesis, that the system of differential equations $\dot{x} = Ax$ is stable (all trajectories converge to zero), if and only if there exists a positive-definite matrix $P$ such that $A^T P + PA \preceq 0$. This condition was perhaps the first LMI [1].

In the 1960s, V.A. Yakubovich wrote extensively about the role of matrix inequalities in control theory. Later, Jan Willems, who actually coined the term LMI in his seminal 1971 paper [5], speculated about their potential role in control theory. Specifically, he mentions that LMIs might be exploited in computational algorithms.

LMIs belong to a special class of optimization problems known as convex optimization. In a convex optimization problem, a linear objective function is minimized (or maximized) subject to a set of linear equality constraints and convex inequality constraints. While the inequalities in an LMI problem are linear functions of matrix variables (hence the name LMI), such problems are inherently nonlinear. This nonlinearity is due to the fact that the condition for a matrix to be negative is a nonlinear function of the entries of the matrix, namely, that the eigenvalues are all negative or, equivalently, the nested principal minors are all negative. Obviously, both of these conditions are messy nonlinear functions of the matrix entries. So how could such a complicated nonlinear problem be solved?

The conventional wisdom in the 1960s and 1970s suggested that an optimization problem is easy if it is linear [linear objective with linear constraints, for example, a linear program (LP)] and hard if it is nonlinear. This view was reinforced by the fact that large scale LPs were being solved easily. Furthermore, it was known how to exploit sparsity and structure to solve very large problems, thanks to Dantzig's simplex algorithm (which had favorable practical complexity) as well as the work of Leonid Kantorovich in the Soviet Union (for which he won the 1975 Nobel Prize in economics). Dantzig's simplex method performed quite well in practice, although certain rare instances took a long time to run.

Until recently, the conventional wisdom in the optimization community



suggested that an optimization problem is "easy" if it is linear (linear objective with linear constraints, that is, linear program, or an LP) and "hard" if it is nonlinear. This view was reinforced due to the fact that LPs with hundreds and thousands of variables can be solved very efficiently, since we know how to exploit sparsity and structure to solve large problems. This was thanks to Dantzig's simplex algorithm that had very favorable practical complexity as well as the work of Leonid Kantorovich in the Soviet Union (for which he won the 1975 Nobel Prize in economics). The simplex method performed quite well in practice, except in rare instances. It took some time to realize that perhaps the easy-hard division is not along linearity or nonlinearity. As Rockefellar stated in 1993, "the great watershed in optimization isn't between linearity and nonlinearity, but between convexity and non-convexity" [4].

A key turning point in changing the conventional wisdom was the 1975 discovery by N.Z. Shor in the Soviet Union of the ellipsoid algorithm and the 1979 discovery by Leonid Khachian that the ellipsoid algorithm solves LPs in polynomial time. In contrast, Dantzig's simplex algorithm had a worst-case exponential complexity. This development was such big news that it became a page 1 story in the *New York Times*. Although the ellipsoid algorithm provides a polynomial time complexity guarantee, its practical performance in terms of running times was worse than the simplex algorithm.

The 1984 discovery of a polynomial-time interior-point algorithm by Karmarkar changed the balance, since Karmarkar's algorithm is efficient in practice as well as in theory. His interior-point algorithm was named for the fact that it moved through the interior of the feasible region to reach the optimal solution, even though the optimal solution is known to be on the boundary.

Simply put, Karmarkar's algorithm converts a constrained optimization problem into an equivalent unconstrained one, in which the cost function is augmented by adding a barrier function whose value goes to infinity when approaching the boundary of the feasible set. As a result, the effect of constraints appears in the objective function, and the problem is effectively rendered unconstrained. All that remains to be done is to set the derivative to zero using, for example, Newton's algorithm to find the minimum of the modified objective function. However, things are not as easy as they seem: to solve this problem efficiently with a theoretical polynomial time complexity, one needs to have an a priori bound on the number of Newton steps needed to get arbitrarily close to the solution. The difficulty is that the classical complexity analysis of Newton's method depends on constants that are functions of the third derivative; these constants are often coordinate dependent and thus are difficult to estimate.

Four years later, a major advance was achieved by Nesterov and Nemirovsky, who realized that Karmarkar's analysis can be modified and extended to a much larger class of convex optimization problems, including but not limited to what we now call LMI problems or semidefinite programs (SDPs) [2], [3].

The barrier function method of converting constrained optimization problems to unconstrained optimization problems was not new on its own (in fact the idea goes back to the 1960s), but here is where Nesterov and Nemirovsky's brilliant idea comes into play: they show that certain logarithmic barrier functions for convex constraint sets (such as the one used by Karmarkar) possess a nice property called *self-concordance*. Roughly speaking, self-concordance means that the the third derivative of the barrier function can be bounded by a function of its curvature or second derivative in a coordinate-independent fashion. As a result, Nesterov and Nemirovsky were able to provide a complexity estimate for the barrier method that was independent of the chosen coordinates, and the number of Newton steps could be bounded a priori. In their seminal papers [2], [3], Nesterov and Nemrovsky show that it is possible to construct a self-concordant barrier for any convex optimization problem; however, not all such barriers are computable. There is, nevertheless, a large set of convex optimization problems for which such barriers exist. These problems can be solved in polynomial time with any given desired degree of accuracy.

This background is only half of the story; the other half is identifying solvable convex optimization problems, which are not always obvious. Sometimes the problem is not convex but, with a change of variables, can be transformed into a convex problem. This point is precisely where the book at hand becomes an invaluable resource.

In *Convex Optimization*, the authors introduce several classes of optimization problems that are amenable to efficient numerical solution with interior point algorithms. One such class is that of conic problems in which a linear objective function is minimized subject to a set of inequality constraints that represent the intersection of an affine space (such as a plane) and a cone (such as the positive orthant).

Examples of conic problems include linear programs (where the cone is the positive orthant), SDPs (where the cone is the set of positive-definite matrices; LMI feasibility problems are of this category), and second-order cone programs (SOCPs) (where the cone is the Lorentz or ice cream cone). All of these problems, which involve optimization problems over self-dual cones, can be treated more or less in the same fashion. As a result, it is possible to develop a duality theory that closely mirrors that of linear programs.

While SDPs and LMIs are familiar in the controls community, SOCP problems are not as well known. A good example of an SOCP, which also demonstrates the difficulty in recognizing convex problems, is a linear program with random constraints. This problem appears to be a linear program in which each row of the constraint matrix is a Gaussian random vector, and each inequality constraint needs to be satisfied with some probability. To an untrained eye, the difficulty of the problem would be the same, irrespective of the value of probabilities. It turns out, however, that when the constraints are required to be satisfied with probability above 0.5, the problem is convex, and an SOCP. If, however, the probability of a constraint being satisfied is below 0.5, the problem is no longer convex and becomes very difficult to solve. Another interesting example of an SOCP is robust linear programming, in which each row of the constraint matrix is an uncertain vector in an ellipsoid and the goal is to minimize the worst-case cost.

Finally, geometric programs (GPs) are additional optimization problems that can be solved efficiently. In a GP, the constraints and objective are posynomials or multivariate polynomials, with positive coefficients whose domains are positive real numbers. In constrast to the convex programs discussed earlier, GPs are not convex in their natural form. However, a simple change of variable can be used to transform a GP into a convex problem. The study of simple geometric programs goes back to the 1960s in the chemical engineering literature, while new applications range from information theory to transistor sizing in RF circuits. The aforementioned examples are just a few samples from the interesting examples of convex optimization covered in the book.

## The Book

*Convex Optimization* by Boyd and Vandenberghe can be used as a graduate-level textbook for students across various disciplines of engineering and applied sciences as well as practitioners in industry. The book is a result of over 15 years of research by the authors and their students in formulating, finding, and solving convex optimization problems in diverse areas of engineering and applied sciences. The authors have used various drafts of this book in graduate-level courses on optimization theory since 1995. As a result, the current manuscript is well written and easy to follow. While familiarity with classical optimization is helpful, it is by no means a requirement for reading the book, since the manuscript is self-contained with a useful set of appendices. That being said, a good working knowledge of linear algebra, advanced calculus, basic probability theory, analysis, and basic topology (norms, open sets, and convergence) is required.

There are three major sections in the book dedicated to theory, applications, and algorithms for convex optimization problems. A semester-long graduate level course can easily cover all three sections. The first five to six weeks or the first half of the course can be used to cover the theory, with the second half divided equally between applications and algorithms. While a diverse set of applications, ranging from statistics, geometry, approximation, and estimation problems, are presented as individual chapters, additional applications appear as exercises. One notable absence on the application side are applications of convex optimization in control theory. The third and last part of the book presents algorithms for solving convex optimization problems along with complexity analysis of the algorithms.

## Contents of the Book

The book consists of three sections including 11 chapters, as well as three appendices and an extensive bibliography. Section I, which covers the theory portion of the book, contains five chapters.

- Chapter 1 provides an introduction to the topic, sets up the notation, and provides a brief history of the topic in its bibliography.
- Chapters 2 and 3 provide a modern overview of convex analysis. Convex sets are defined in chapter 2. Convex functions and operations that preserve convexity are discussed in chapter 3.
- Chapter 4 is the core of the text, where LPs, SDPs, SOCPs, and GPs are introduced.
- Chapter 5 presents a unified theory of duality for conic optimization problems. Several interpretations of duality, ranging from economics to mechanics, are also presented.

The second section, which includes chapters 6–8, covers applications. Application areas such as convex geometry, statistics, estimation, and approximation problems are presented.

- Chapter 6 includes several interesting approximation and fitting problems, ranging from norm approximations, least norm problems, reconstruction and smoothing, function fitting and interpolation as well as approximation. This chapter should be of interest to the signal processing community.
- Chapter 7 covers statistical estimation, experiment design, maximum likelihood problems, robust detection, and hypothesis testing.
- Chapter 8 contains a variety of geometric problems, such as Euclidean distance and angle problems, classification problems, and placement and facility location problems.

The third section of the book deals with algorithms for solving convex optimization problems.

- Chapter 9 contains algorithms for unconstrained optimization problems. Topics covered include classical analysis of gradient algorithms for unconstrained problems as well as classical and modern analysis of Newton's algorithm for such problems. The key idea is the demonstration that the quality of the gradient algorithm depends on the chosen coordinates, whereas Newton's method is coordinate independent. Self-concordance turns out to be the crucial property that allows a coordinate-independent analysis of Newton's algorithm. This property is the main reason for the success of barrier-based interior point algorithms.

- Chapter 10 extends the analysis of chapter 9 to the case of equality constrained convex optimization problems.
- The final chapter consists of a detailed analysis of interior point algorithms for convex optimization problems with linear equality constraints and convex inequality constraints using barrier and primal-dual methods.

## Conclusions

*Convex Optimization* is a great segway into the interesting world of theory, applications, and algorithms for convex programs that can be solved efficiently. The text is suitable for a graduate-level course in any engineering department. I have used an earlier draft of the book in a graduate level optimization course at the University of Pennsylvania. The book is also useful for practicing engineers since it presents the state of the art in theory, applications, and algorithms.

The key goal of the book is to help the reader identify and solve convex optimization problems in diverse disciplines of engineering and applied sciences. In my opinion, the authors have achieved this goal. Another feature of this text is that the authors have posted a copy of the book on their Web site, which also includes slides for teaching a course based on the text. While the book does not focus on LMIs for control applications, the text provides the reader with an understanding of how LMI algorithms work.

Finally, I believe the authors have done a great job in providing a rigorous but comprehensible explanation of the success and efficiency of convex optimization problems. Even though convex optimization is more or less a technology, there is still a long way to go for the technology to mature. While linear programs and least squares problems can be solved for thousands of variables, other forms of convex optimization such as SDPs and SOCPs are solvable for hundreds of variables but not thousands. Exploiting structure and sparsity as well as parallelization are subjects of ongoing research.

## References


[1] S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan, *Linear Matrix Inequalities in Systems and Control Theory* (Studies in Applied Math., vol. 15). Philadelphia, PA: SIAM, 1994.

[2] Yu. Nesterov and A. Nemirovsky, "A general approach to polynomial-time algorithm design for convex programing," Centr. Econ. and Math. Inst., USSR Academy of Sciences, Moscow, USSR, Tech. rep., 1988.

[3] Yu. Nesterov and A. Nemirovsky, *Interior Point Polynomial Methods in Convex Programing* (Studies in Applied Math., vol. 13). Philadelphia, PA: SIAM, 1994.

[4] R.T. Rockafellar, "Lagrange multipliers and optimality," *SIAM Rev.,* vol. 35, no. 2, pp. 183–238, 1993.

[5] J. Willems, "Least squares stationary optimal control and the algebraic Riccati equatio," *IEEE Trans. Automat. Contr.*, vol. 18, pp. 184–186, 1971.

[6] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge Univ. Press, 2004 [Online]. Available: http://www.stanford.edu/~boyd/cvxbook.html

[7] V.A. Yakubovich, "The solution of certain matrix inequalities in automatic control theory," *Soviet Math. Dokl.*, vol. 3. pp. 620–623, 1962.



***Ali Jadbabaie*** received his Ph.D. from the Control and Dynamical Systems Department in 2001 from Caltech. After spending a year as a postdoctoral associate at Yale, he joined the Department of Electrical and Systems Engineering at the University of Pennsylvania, where he is an assistant professor. His research interests are in cooperative control of networked systems, optimization theory, and optimization-based control.


---

***Neural Engineering: Computation, Representation, and Dynamics in Neurobiological Systems*** by Chris Eliasmith and Charles H. Anderson, MIT Press, Cambridge, 2003, ISBN: 0-262-05071-4, US$49.95. *Reviewed by Jagannathan Sarangapani.*

## Computational Neuroscience

Computational neuroscience has only recently been established as a scientific discipline in its own right. Since its inception, computational neuroscience has been dedicated to the modeling and simulation of biological neural systems, while neuroscience per se focuses on improving our understanding of how the brain and spinal cord work.

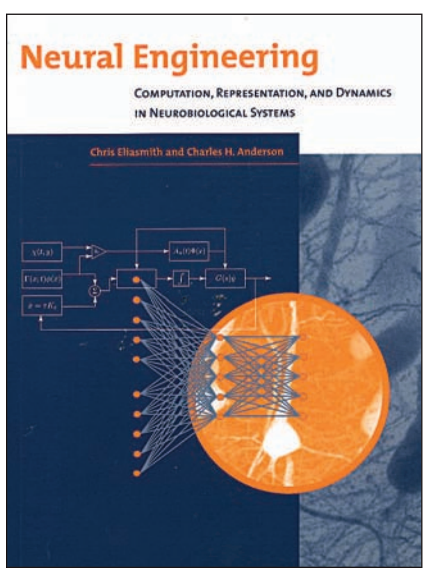
Computational neuroscience and engineering scientists are interested in how to model a single neuron and its information processing capability, characterize neural networks as time-varying control structures, and apply techniques to generate large-scale realistic simulations of networks of neurons so that a specific biological behavior of the brain can be emulated. Novel models of biological systems such as the locomotor, vestibular, and working memory systems are essential components for understanding the brain.

Due to the increasing importance of computational neuroscience in engineering, a need has developed for a textbook for senior undergraduate and beginning graduate students. *Neural Engineering* introduces the theoretical foundations of neuroscience with a focus on the nature of information processing in the brain. In particular, this book encapsulates recent advances in computational neuroscience and extends ideas about neural coding, neural computation, physiology, communication theory, control theory, representation, dynamics, and statistical learning concepts.

With the advent of novel computational neuroscience concepts, control system engineers involved in the design and implementation of intelligent control systems for industrial processes can now use the neural engineering concepts discussed in this book. Control system engineers take pride in modern feedback control concepts that have been responsible for major successes in aerospace, automotive, defense, and industrial systems. However, the complexity of today's systems, such as formation flying of aircraft and multi-vehicle cooperative control, has placed severe demands on existing control design techniques. More stringent performance requirements in both speed and accuracy in the face of system uncertainties and unknown environments have challenged the limits of modern feedback control. Operating a complex system in multiple regimes requires that the controller be intelligent with adaptive and learning capabilities in the presence of unknown disturbances, unmodeled dynamics, and unstructured uncertainties [1].

The application of neural networks to closed-loop control systems has only recently been rigorously studied. Of particular importance to intelligent-control engineers is the neural network universal approximation property. Approximation-based intelligent controllers are based on approximating the unknown nonlinear dynamics of the complex system. These controllers include adaptive and robust control techniques as well as learning control, neural network, and fuzzy logic techniques. Intelligent controllers based on artificial neural networks can approximate the nonlinear dynamics of industrial systems without assuming linearity in the unknown parameters (LIP), which is often required in adaptive control. Some forms of friction, for instance, are not linear in the parameters, while many nonlinear industrial processes do not satisfy this assumption [2]. Moreover, the LIP assumption requires a regression matrix for the nonlinear system that usually involves tedious computations.

Whereas robust control techniques require a bounding function for the unknown system dynamics, neural network controllers can learn the unknown dynamics online. The passivity property of neural networks provides robustness to disturbances and unmodeled dynamics. To design a neural network controller, however, users must select a suitable architecture, network size, weight-tuning schemes for learning, and neuron-activation functions, which require a good understanding of the neuron information-processing capabilities, neuron transformations, and associated neural network models from a neurobiological perspective [1], [2].




## Neural Engineering

*Neural Engineering* introduces neuron models that are suitable for exploring information processing in large brain-like networks. The book also introduces several neural network architectures and discusses their relevance for information processing in the brain along with models of higher-order cognitive functions. An additional feature is the availability of MATLAB programs for exploring the models described in the book. An accompanying webpage includes programs for download.

*Neural Engineering* addresses the gap between the low-level spiked neuron models and a population of neurons represented as high-level cognitive models. The book covers the principles and methods of modeling and understanding diverse neural systems, while presenting a coherent picture of neural function from single cells to complex networks. A unified framework is presented in terms of models, assumptions, and techniques for understanding neural systems. The authors review this emerging field in historical and philosophical overviews and in stimulating summaries of recent results.

Chapter 1 introduces the single neuron, the neural system, and neural transformations. Since the central goal of *Neural Engineering* is to provide a general framework for constructing neurobiological simulations, the authors describe the methodology in three steps, specifically, system description, design specifications, and implementation.

Biomedical and pattern recognition engineers will find chapter 2 useful. This chapter is devoted to representing scalar magnitudes of signals and systems in the nervous system by using engineering and biological representations. To represent neurobiological systems and the effect of noise on the transmission of a signal from a neuron to its neighbor, a noise term is added to the transmitted firing rate to introduce random variation into the neuron's activity. The proposed framework and empirical results are shown on examples involving horizontal eye position, arm movements, and the semicircular canal of the vestibular system.

Chapter 3 represents a population of neurons using a nonlinear mapping of input signals since visual images, auditory streams, patterns of movements, and tactile sensations are a nonlinear function of multiple variables such as light intensity and spatial location. For instrumentation and biomedical engineers, such representations are of interest since advanced instrumentation can be designed and developed.

The goal of chapter 4 is to represent time-varying signals by spiking neurons using a spiking version of the leaky integrate-and-fire (LIF) model. The LIF neuron is best understood as a passive RC circuit coupled with an active spike (delta function) generator. This passive RC circuit is represented mathematically as a first-order differential equation. For a neural network engineering researcher, the LIF model incorporates physiological parameters, including membrane capacitance, membrane resistance, and absolute refractory period. The LIF model, which is a good approximation of most neurons operating over a normal range, incorporates the important nonlinearity of neural spike.

Next, the most interesting and important discussion included in this chapter is the information transmission capacity per frequency channel, which is measured in bits per spike or bits per second. This issue may be of interest to engineers in selecting an activation function. Next, the canonical model, which incorporates a family of models that effectively capture the behaviors of an entire class of neurons, can be studied by using the $\theta$-neuron. This $\theta$-neuron model captures the qualitative features of spiking neurons and compares favorably with the LIF neurons. Control researchers have developed-neuron based neural network controllers to approximate nonlinear functions over compact sets with fewer hidden layer neurons than feedforward neural network architectures.

Chapter 5 extends neural representation by means of encoding and decoding. Nonlinear encoding is defined as the process of representing a neurobiological process mathematically as a function mapping from one space to the other. By contrast, decoding is defined as the process of obtaining an inverse of the function mapping. Population and temporal representations of neural populations are also united in this chapter by viewing them as time-varying vectors. The addition of noise is included, and the effect of distortion is analyzed. This chapter is relevant to engineers working in the pattern recognition area.

While the first few chapters focus on representations of neurobiological systems, neurobiological transformations are characterized in chapter 6. The neurobiological communication channel between two neuron populations is used to explain the concept of a neurobiological transformation. A communication channel example illustrates how one can model a simple biological process by using a one-layer artificial neural network consisting of weights, biases, neuron-activation functions, and inputs. In addition, the chapter explains that the effect of noise should be taken into account at a very early stage of modeling since many biological processes are sensitive to noise. Once a suitable model is developed, the book presents how to analytically determine neural network weights to accurately represent a biological process as a nonlinear mapping of inputs, weights, biases, and outputs. For an engineer, this neurobiological characterization explains how to model physical systems using artificial neural networks and ensures that artificial neural networks can approximate a nonlinear mapping with a suitable set of weights. In addition, this framework indicates that finding weights by analytical means is generally far less computationally intensive than running large training regimes.

By contrast, computational intelligent control engineers typically use an offline training phase to identify suitable neural network weights, and, once these weights are selected, the weights are not tuned online. It is often difficult to identify training sets of input and output signals for constructing an unknown nonlinear mapping. On the other hand, to overcome the need for training sets, it is typically assumed that a linear representation of the unknown nonlinear mapping is known beforehand, and the linear representation is used to identify suitable neural network weights. However, the weights that are determined analytically by linear means can be used as first guesses for nonlinear transformations generated by neural networks since complex behavior exhibited by neural systems is unlikely to be fully explained by a linear transformations alone. Chapter 7 illustrates this point by using the response of neurons in the visual pathway. However, the concept of a multilayer neural network and how to select weight-tuning schemes to approximate a given transformation are not covered.

Chapter 7 also characterizes neural representations and transformations by comparing the standard algebraic notion of a complete orthogonal basis with the concept of an overcomplete basis. In an orthogonal basis, the dot product of any two vectors is zero. In contrast, an overcomplete basis does not require linearly independent vectors, and thus an overcomplete basis is redundant. Descriptions employing overcomplete bases are not as succinct as those employing complete, or orthogonal, bases. However, in the uncertain and noisy world of physical systems, such as neurobiological systems, this redundancy is often valuable for error correction and the efficient use of available resources.

These results are of interest to control researchers since one-layer neural networks with basis functions or multilayer neural networks without basis functions are normally used in an ad hoc manner for designing controllers. The selection of basis functions for a single-layer neural network to achieve a given approximation accuracy is currently not well understood. Moreover, overcomplete basis techniques have not yet been applied in neural network control. I am hopeful that neural engineering can shed some light in the selection of basis functions for a one-layer artificial neural network since a one-layer neural network is computationally efficient and viable for controlling real-world systems.

Neuroscientists and engineers will be interested in studying the neurodynamic models presented in chapter 8. In this chapter, linear system techniques are used to represent neurodynamics. The neural control diagram is introduced next, where the dynamics of a generic population of neurons are represented by a block diagram with inputs and outputs. Using Laplace transforms, a relationship is derived between the dynamics and input matrices of the generic neuron population. However, nonlinear neurodynamics and their stability properties are not sufficiently discussed. Instead in this chapter, the three fundamental principles of neural engineering are quantified, namely:

- Principle 1. Neural representations are defined by a combination of nonlinear encoding and weighted linear decoding.
- Principle 2. Transformations of neural representations are functions of many variables and are determined using an alternatively weighted linear decoding.
- Principle 3. Neural representations can be considered as state variables in system theory, and thus neurobiological dynamics can be analyzed using control theory.

Chapter 9 discusses statistical inference, which is essential for explaining the behavior of animals in a noisy and uncertain world. This chapter outlines the ability of neurobiological systems to implement the transformations needed to support complex statistical inference in the presence of noisy information. For example, parts of the brain are represented using a Kalman filter, which combines ideas from statistical inference and control theory. In addition, this chapter includes preliminary results using Hebbian learning rules for transmitting a signal between two networks of neurons over a simple communication channel. In fact, it is interesting to note that Hebbian learning rules and modified Kalman filter update equations have been used to tune artificial neural network weights in the context of controller design [1]. Finally, potential dissertation topics for neuroscience students are discussed.

## Target Audience

*Neural Engineering* attempts to generate a coherent understanding of neurobiological systems from a systems neuroscience perspective that caters to a wide audience ranging from physiologists to physicists. This book is intended primarily for neuroscientists interested in learning about methods for characterizing the neurobiological systems that are studied experimentally. In-depth understanding of the development of computational models and a simulation environment for large-scale neural models is included. The book's target audience also includes engineers, physicists, and computer scientists interested in learning how quantitative tools relate to the brain. A good discussion on biological systems using familiar tools such as linear algebra, signal processing, control theory, and statistical inference is covered.

The book includes curriculum and programming tools for a graduate-level course in computational neuroscience. Although the book does not include problem sets, solutions, course notes, or examples, these materials along with a code library written in MATLAB are available in [3]. Novel simulation models of commonly modeled biological systems, for instance, locomotor, vestibular, and working memory systems included in [3], provide readers with the means to compare the framework described in this book with that of other methods.

The book covers introductory mathematical analysis of neurobiological systems and thus will be useful as a reference text for engineers. By contrast, the book can be used as a graduate text for students in neuroscience. In my opinion, since sufficient details are not included on neural dynamics, stability issues, and how the brain identifies and tunes the network weights, control system engineers may not be keen on using the book as a textbook for a neural network control systems course. Another weak point is the lack of problems at the end of each chapter.

Despite the lack of a sufficient mathematical treatment of neural engineering concepts and the relevance of neural engineering to real-world control applications, I found this book to be interesting and useful for engineering students. Books such as [4]–[8] require extensive background on physiology and anatomy and thus are useful only to neuroscientists. By contrast, *Neural Engineering* provided me with ideas on how to present neural networks and neurocontrol to control students.

In conclusion, I recommend this book to advanced engineering students, to active researchers in neuroscience, and to those who may be interested in moving into this field. This book is small enough to be manageable in a semester-long course but large enough to contain an abundance of material. However, since the book does not provide detailed coverage of learning paradigms, stability analysis

[2], and control design, I recommend that the book be considered only as a reference text for undergraduate and graduate engineering students.

## References


[1] K.S. Narendra and K.S. Parthasarathy, “Identification and control of dynamical systems using neural networks,” *IEEE Trans. Neural Networks*, vol. 1, pp. 4–27, 1990.

[2] F.L. Lewis, S. Jagannathan, and A. Yesilderek, *Neural Network Control of Robot Manipulators and Nonlinear Systems*. London, UK: Taylor and Francis, 1999.

[3] http://compneuro.uwaterloo.ca/

[4] W.Y. Lytton, *From Computer to Brain*. New York: Springer Verlag, 2002.

[5] M.I. Posner, *Cognitive Neuroscience of Attention*. New York: Guilford, 2004.

[6] E. Gordon, *Integrative Neuroscience*. London, UK: Taylor and Francis, 2000.

[7] S. Zornetzer, *An Introduction to Neural and Electronic Networks,* 2nd ed. San Mateo, CA: Morgan, Kaufmann, 1995.

[8] G.M. Ederman, Ed., *The Brain*. New Brunswick, NJ: Transactions, Oct. 2000.



***Jagannathan Sarangapani*** is a professor in the Department of Electrical and Computer Engineering at the University of Missouri-Rolla. His research interests include embedded neural network and adaptive control, computer, communication, and sensor networks, diagnostics and prognostics, and autonomous systems.

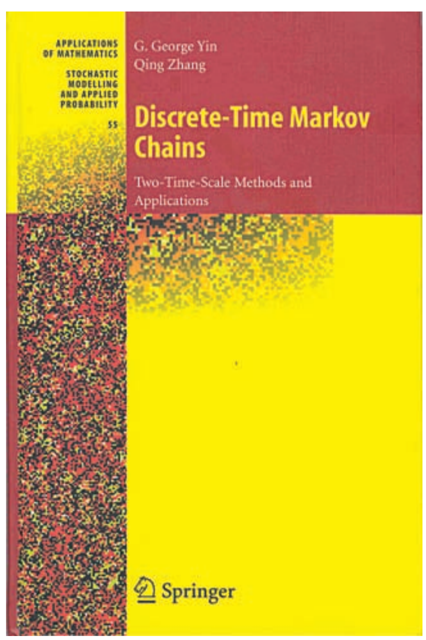

---

***Discrete-Time Markov Chains*** by G. George Yin and Qing Zhang, Springer, New York, 2005, ISBN: 0-387-21948-X, US$79.95. *Reviewed by Vivek S. Borkar.*

## Markov Chains and Two Time Scales

Discrete-time Markov chains are the basic building blocks for understanding random dynamic phenomena, in preparation for more complex situations. Not surprisingly, there have been many textbook-level treatments of discrete-time Markov chains [2], [3]. The book by Yin and Zhang, despite its title, is not one of these. Rather, the scope of this book is better captured by its subtitle *Two-Time-Scale Methods and Applications*. As this subtitle suggests, the book is a monograph on singular perturbation theory involving Markov chains with two time scales.

Perturbation theory of dynamical systems deals with situations in which the dynamics can be viewed as a small perturbation of another underlying dynamics by an additional component weighted by a small parameter $\epsilon > 0$. As $\epsilon \downarrow 0$, the dynamics reduce to the underlying dynamics. The perturbation is said to be regular if there is no drastic qualitative change in the dynamics during the passage to the limit $\epsilon \downarrow 0$. If there is such a change, then the perturbation is singular. The precise meaning of this condition depends on the context. For example, the perturbation of an ordinary differential equation is singular if, in the limit, the dimension of its state space suddenly drops by one or more. Thus the perturbation of $\dot{x}(t) = f(x(t))$ to $\dot{x}(t) = f(x(t)) + \epsilon g(x(t))$ with a nice $g(\cdot)$ is regular, whereas the perturbation of the differential-algebraic system




$$\dot{x}(t) = f(x(t), y(t)),$$
$$g(x(t), y(t)) = 0,$$

to

$$\dot{x}(t) = f(x(t), y(t)),$$
$$\epsilon \dot{y}(t) = g(x(t), y(t))$$

is singular. This distinction is due to the fact that, as $\epsilon \to 0$, a higher dimensional differential equation changes into a lower dimensional algebraic-differential system.

As another example, consider a discrete-time Markov chain with the parameterized transition matrix $P + \epsilon Q$, which is irreducible for $\epsilon > 0$. That is, for all $\epsilon > 0$, the state space does not split into two or more disjoint classes for which the probability of moving from one to the other is zero. This situation can be considered a regular perturbation of the Markov chain with transition matrix $P$ if the latter is also irreducible. If not, the perturbation is singular since the state space decomposes in the limit into disjoint parts unreachable from one another.

The singular case is usually more difficult and also more interesting to analyze. Singular perturbations based on two time scales involve separation of the dynamics into fast and slow modes. One major task of singular perturbation analysis is to characterize the limiting dynamics as $\epsilon \downarrow 0$. The limiting dynamics are typically simpler in some crucial aspect than the original (for example, due to lower dimension) and are therefore easier to analyze. The limiting system can then be a valid approximation to the perturbed system for small $\epsilon > 0$. A harder problem is to quantify the errors in this approximation. Asymptotic expansions associated with the limiting procedure are used for this purpose.

Yin and Zhang have made many significant contributions to this area, and also to the corresponding issues for continuous-time Markov chains. Their contributions to the latter domain have already appeared as a monograph [5]. This book presents its discrete-time counterpart. At the other end of the spectrum, one has continuous-time and state Markov processes with two time scales studied in [4]. Between them, the three monographs cover much of the ground in singular perturbation theory for two time-scale Markov processes. Even so, there is a difference between this book and some of the mainstream singular perturbation theory of discrete-time Markov chains typified by the recent survey [1]. The latter also deals with averaging and asymptotic expansions in powers of $\epsilon$ for various averages related to the singularly perturbed Markov chains. The present book looks at a different type of expansion, an example of which is the probability distribution of the chain after $k$ steps given by

$$p_k^\epsilon \approx \sum_{n=1}^{N} \epsilon^n \varphi_n(\epsilon k) + \sum_{m=1}^{N} \epsilon^m \psi_n(k).$$

This distribution has the region of validity, $0 \leq k \leq T/\epsilon$ for some $T > 0$. A crucial feature is the two-fold role of the perturbation parameter $\epsilon$, which defines the scale of the perturbation because, in the perturbation $P + \epsilon Q$ of the transition matrix $P$ introduced earlier, the perturbation matrix $Q$ is weighted by $\epsilon$. In addition, $\epsilon$ is also used for scaling the time axis by considering time steps $\epsilon k, k \geq 0$, over the time interval $0 \leq k \leq T/\epsilon$, which increases to infinity as $\epsilon$ decreases. The first sum on the right is the outer expansion aimed at giving a good approximation at times away from zero. The second sum is the initial layer correction that ensures correct initial conditions. The expansion is thus at a process level and not only for aggregated averages.

This theory is appealing for applications that involve a dynamic law that switches randomly between many alternatives according to a singularly perturbed Markov chain. The dynamic law may be prescribed, for example, by the driving vector field of an ordinary differential equation or the drift and diffusion coefficients of a stochastic differential equation. This formulation has been the main aim of the authors' research and of this book. In fact, the book naturally partitions into two parts. The first six chapters, along with chapter 14, is the theoretical foundation for the book. The rest of the book deals with applications to filtering, stability, and control of stochastic systems with singularly perturbed Markov switching.

## Contents

The book has 14 chapters, of which three are background material, four are devoted to developing the theory, and the remaining seven deal with specific applications.

Chapter 1 builds up motivation by means of examples from areas such as manufacturing and communications. This chapter also gives an overview of the book and a brief literature survey to place the book in perspective. Chapters 2 and 14 deal with the mathematical background. The former deals with basic material such as Markov processes and martingales, which are usually covered in an advanced probability course. This material provides the essential framework in which the results of the book can be stated and proved. Chapter 14, in turn, covers more advanced and specialized topics from probability and control theory such as weak convergence and the Hamilton-Jacobi-Bellman equation. These topics are used elsewhere in the book in various specific applications.

Of the theoretical chapters, chapter 3 develops the asymptotic expansions mentioned above and derives error bounds. Chapter 4 extends this theory to analyze the limit behavior of associated occupation measures. This analysis is done first for the irreducible case and then for the general case. In particular, the chapter proves a functional central limit theorem for interpolations of suitably normalized occupation measures. Chapter 5 derives some exponential moment bounds. Chapter 6 recapitulates the contributions of these chapters and provides some extensions.

This material thus far is preparation for chapters 7–13, each of which deals with a specific application. Chapter 7 looks at deterministic and stochastic difference equations, where the dynamics are modulated by a singularly perturbed Markov chain as described above. This chapter establishes stability and the associated bounds on mean hitting times for these systems in terms of a stochastic Lyapunov condition on the limiting system. Chapter 8 considers the filtering problem for a linear stochastic system with the coefficients of state and observation dynamics modulated by a singularly perturbed Markov chain. It is shown that the exact filter for the limiting system is near optimal for the original system. The chapter also considers approximations of the Wonham filter for the Markov chain when the chain itself is partially observed.

Chapter 9 considers Markov decision processes with singular perturbations. The main result shows that the value function of the limiting process provides a good approximation to the value function of the original problem. Also, the corresponding optimal policy for the limiting problem is shown to be near optimal for the original problem. An example from manufacturing is given, where the objective is to optimize the scheduling of maintenance and repair of machines subject to random breakdown.

Chapter 10 establishes analogous results on approximation of value functions and near-optimal controls for the control of a linear stochastic system with coefficients switching according to a singularly perturbed Markov chain, with quadratic cost. Continuing in a similar vein, chapter 11 considers the mean-variance control problem for portfolio optimization with - Markov-modulated coefficients. This chapter uses the limiting problem and standard methods to establish near optimality of controls. Both discrete-time and continuous-time problems are considered. Chapter 12, in turn, derives more results in this spirit for a control problem arising from production planning. Chapter 13 considers a different class of problems, namely, least-mean-squares stochastic approximation algorithms associated with an irreducible Markov chain whose transition probabilities are modulated by another singularly perturbed Markov chain. A switching diffusion limit is obtained for interpolated iterates. The results are extended to hidden Markov models.

## Summary

As should be clear from the foregoing, this book is not a text, not even a graduate text. Rather, the book is a research monograph based largely on the authors' own work. The book offers little by way of a general background in discrete-time Markov chains. Rather, the book focuses entirely on two-time-scale analysis for discrete-time Markov chains and thus complements but does not supersede traditional works such as [1]. The book does, however, fill an important niche in the literature on singularly perturbed Markov chains.

A strong point of the book is the numerical examples spread throughout to illustrate theoretical results. I also liked the way each chapter is organized. The main results are stated without proof first, whereas the proofs are given later at the end of the chapter. This format will be of great help to researchers in other areas who want to know and use the results without going into the details of the proofs. Another strong point of this book is the fact that many of the results are established for the most general case that allows for transient states in the unperturbed Markov chain.

The kind of problems analyzed here occur frequently in areas such as manufacturing and communications. Hence, the book will be useful to applied probabilists and engineers who deal with such systems. Other than this, the book's primary audience is other researchers in singularly perturbed Markov chains.

## References


[1] K.E. Avrachenkov, J. Filar, and M. Haviv, "Singular perturbations of Markov chains and decision processes," in *Handbook of Markov Decision Processes*, E.A. Feinberg and A. Shwartz, Eds. Norwell, MA: Kluwer, 2002, pp. 113–150.

[2] P. Brémaud, *Markov Chains*. New York: Springer, 1999.

[3] J.R. Norris, *Markov Chains*. Cambridge, UK: Cambridge Univ. Press, 1999.

[4] Y. Kabanov and S. Pergamenshchikov, *Two-Scale Stochastic Systems*. Berlin: Springer, 2003.

[5] G. Yin and Q. Zhang, *Continuous-Time Markov Chains and Applications: A Singular Perturbations Approach*. New York: Springer, 1998.



***Vivek S. Borkar*** received his B.Tech. in electrical engineering from I.I.T., Mumbai, in 1976, his M.S. in systems and control from Case Western Reserve University in 1977, and his Ph.D. in electrical engineering and computer science from the University of California, Berkeley, in 1980. He was with the TIFR Centre, Bangalore (1982–1989) and the Indian Institute of Science, Bangalore (1989–1999), before joining the Tata Institute of Fundamental Research, Mumbai. He is a Fellow of the IEEE, Indian Academy of Sciences, Indian National Science Academy, and Indian National Academy of Engineering. His research interests are in stochastic optimization and control and their applications.


Today, self-correcting mechanical systems surround us. So it's hard for us to appreciate the impact of Wiener's ideas in the mid-20th century, when he pointed out the similarity between machines with sensory systems that collected information to fine-tune their behavior and biological systems—like human beings—that did the same thing. Cybernetics—Wiener's theory of "control and communication in the animal and the machine"—made him a cultural figure prominent enough to be featured in *Time* magazine cover stories.

From "No Mean Mathematician," by Mark Williams (*Technology Review*, June 2005), a review of the book *Dark Hero of the Information Age: In Search of Norbert Wiener, the Father of Cybernetics*