

## Control-oriented system identification: an $H_\infty$ approach

Jie Chen and Guoxiang Gu  
*John Wiley & Sons Inc.*  
 2000, 421pp., £54.95  
 ISBN 0 47132 048 X

The objective is to present methods for identification of models from experimental data in the form required for robust control by design by  $H_\infty$  methods; that is in the form of a nominal deterministic plant model plus an explicit deterministic uncertainty model. The book brings together at graduate level many results otherwise scattered in a large number of sources. Prerequisites are fluency in mathematical analysis and operator theory and some familiarity with robust control theory.

The book is written at a fair degree of mathematical sophistication, although helpful insight into the underlying context is frequently provided. The text is concerned with  $H_\infty$  identification and model validation with attention being largely restricted to linear time-invariant systems. What the authors consider to be the mathematical prerequisites are summarised in the short but pleasingly written chapter two; mostly the material there is a tutorial attempt to establish a unifying operator-theoretic setting in which control and identification problems can be represented as mappings between appropriate function spaces.

When the treatment proper begins and  $H_\infty$  identification starts to be discussed, it is first explained very simply in a familiar conventional transfer function setting. There is then a progression covering untuned and tuned linear algorithms, two-stage nonlinear algorithms, nonlinear interpolatory algorithms, identification in the time domain and identification of continuous time systems. Everywhere, plentiful comments are clear and well written. For instance, in the summary of chapter 7: 'The time domain approach developed in this chapter is an identification plan directly mapping time series input-output observations to a stable identified model. This plan is similar to but also different from the frequency domain algorithms presented in earlier chapters. Moti-

vated by classical interpolation theory, it too requires solving an analytic function interpolation problem, seeking to match the power series of the identified model to the available time domain data'.

The remainder of the book proper is taken up with chapters on model validation. On this topic the authors' comment is typical of their style throughout: 'It is intuitively clear...that a model that has not been invalidated by the presently available data, may well be so by future data...and the primary utility of a model validation test is actually to invalidate a model'.

There are appendices on information-based and computational complexity and on LMI (linear matrix inequality) programming. A good set of supporting and forward-pointing

references is supplied. The text is well supported by illustrative examples and exercises.

Unusually and admirably in a book written at an advanced theoretical level, the authors never lose sight of engineering realities. Well-written paragraphs explaining the material or relating it to a wider mathematical or engineering context abound throughout the book and are one of its great strengths.

This is a valuable, pleasingly well-written book that requires a knowledge of the function-analytic mathematics that is routinely taught to engineers at continental universities, but that is more of a rarity in UK engineering courses.

PROF. RON LEIGH  
 Brunel University

## The road to the unified software development process

Ivar Jacobson, revised and updated by Stefan Bylund  
*Cambridge University Press*  
 2000, 358pp., £24.95  
 ISBN 0 52178 774 2

Ivar Jacobson is one of the creators of the Unified Modelling Language (UML). Here we have a collection of his articles from the past decade or so, compiled by a former colleague, Stefan Bylund. The articles are interspersed with recent interviews given by Jacobson, which provide a reflective commentary.

Jacobson reminds us that UML is now 'supported by the whole software industry'. The language forms part of the Unified Software Development Process, and the book tells the story of how the language and the process came to exist. We learn that the roots of UML can be traced back to Jacobson's days at Ericsson in Sweden in the late 1960s. As one of the designers of a precursor to the AXE telephone switch, he pressed for the adoption of object-oriented design methods—in the face of strong opposition. He was vindicated by the subsequent success of the AXE project, and the fact that, as he claims,

the design methods will outlive the product several times. Many features of UML, such as class, collaboration, sequence and state charts, and activity diagrams, have direct counterparts in the methods used for AXE. Another offspring was the CCITT language, SDL, but for various reasons this has not received mainstream adoption as an object-based modelling technique. The Unified Process (which now exists as a commercial offering from the Rational Software Corporation) emerged from an earlier product known as Objectory. This, in turn, was derived from a textbook version called OOSE.

In a 1993 article, appearing in an early chapter of the book, Jacobson made a plea for a 'ceasefire in the methods war', claiming that 27 different object-oriented design methods were in existence. The solution was not for a committee to impose some standard from above (as had been done by the CCITT for SDL) but for an 'ad hoc group of knowledgeable and motivated people who believe that sound agreements are necessary' to be formed. By the end of the book, he has argued that UML—used in conjunction with the tools of the