MISER3

OPTIMAL CONTROL SOFTWARE

VERSION 3

THEORY AND USER MANUAL

L. S. JENNINGS
M. E. FISHER
K. L. TEO
C. J. GOH

Department of Mathematics
The University of Western Australia
Nedlands, WA 6907, Australia

For sales enquiries please consult the Web site:
`http://www.cado.uwa.edu.au/miser/`
or email the authors at:
`les@maths.uwa.edu.au`

Collaboration is sought and is available depending on the interests of the authors and the novelty of the optimal control problem.

Consulting is available at commercial rates.

L.S. Jennings, M.E. Fisher, K.L. Teo, and C.J. Goh
MISER3 Optimal Control Software: Theory and User Manual.
Version 3

PREFACE

MISER3 — Version 1 (1990)

With the advances of modern computers and the increasing emphasis on optimal design of large scale dynamical systems under scarce availability of resources, optimal control theory has become a useful tool for solving many industrial and management problems. Most real-world problems are much too complex to be solved analytically, and so computational algorithms are a necessary pre-requisite in the solution of real-world problems. As a result, efficient computational methods for solving optimal control problems have been found to be in great demand. Many sophisticated algorithms are now available in the open literature. However, most of these have not been translated into software packages that can be easily used by those not conversant with optimal control theory. For this reason, the software MISER, consisting of two programs: MISER1 and MISER2, was developed by Goh and Teo (1988a).

We now consider that there is an urgent need to rewrite the software MISER for the following reasons.

(1) Both MISER1 and MISER2 are not as user-friendly as we would like and error checking facilities were not available.

(2) The constraint transcription used for handling the continuous state inequality constraints was not satisfactory in the sense that the equality constraints so obtained failed to satisfy the usual constraint qualification. Hence, the convergence of the numerical algorithm was not guaranteed often resulting in oscillation in the numerical computation. Numerical studies have confirmed that the algorithm rarely converged for nontrivial optimal control problems involving continuous state constraints, although it did give good approximate results. Furthermore, it is impossible to numerically overcome the violations of the continuous state constraints, using the above mentioned constraint transcription.

(3) Several theoretical advancements, including a much superior technique for handling the continuous state inequality constraints, have been made in Teo and Jennings (1989). A method of either introducing numerical stability or penalising control variation (Teo and Jennings, 1991) has some interesting applications and is included in the new package.

As for MISER1 and MISER2, the concept of control parametrization was singled out to form the theoretical basis for the development of MISER3, as we believe that this concept is versatile and easily understood.

We have also included in the software package a program called DMISER3 for solving discrete-time optimal control and optimal parameter selection problems. This program is almost identical to MISER3 except that it applies to problems in which the dynamics is described by difference equations.

Although the main aim of this book is to be a user manual, we have, for the sake of completeness, included in Chapter 3, some crucial theoretical results on mathematical programming required for a thorough understanding of the software. For users who are merely concerned with problem solving, an understanding of Chapters 2 and 4 is sufficient for continuous time problems while Chapter 5 contains all material relevant to discrete-time problems. Chapters 4 and 5 also include a mathematical description of the worked examples included in the distribution. The examples in Chapter 4 demonstrate the various extensions that MISER3 has over previous versions, and also demonstrate novel ways of transcribing optimal control problems to the canonical form required by MISER3. For readers whose interests carry beyond the mundane routine of cranking out numerical solutions, the book “A Unified Computational Approach for Optimal Control Problems” by K.L. Teo, C.J. Goh and K.H. Wong, Longman Scientific and Technical, 1991, provides the theoretical basis for the MISER3 package. The material in Chapter 5 of this manual is essentially new, although some is discussed in Teo, Liu and Goh (1990) while much of the chapter is reproduced in Fisher and Jennings (1992).

We wish to thank Dr. R. Womersley of the School of Mathematical Sciences at the University of New South Wales for his valuable comments and suggestions on the earlier drafts of Chapter 3. He has done much to improve the material but is not responsible for any of the remaining shortcomings. The authors would also like to acknowledge Dr. Y. Liu for helpful discussion and programming assistance and Volker Rehbock for help in testing the software.

MISER3 — Version 2 (1997)

Version 2 of the continuous constrained optimal control software incorporates several extensions and new features which the authors had need of as they worked on new classes of optimal control problems. Chapter two outlines these changes, which make the software much larger as an executable, but then workstations seem to have outstripped our needs in terms of central memory provision. In summary:

1. Each control function may be modelled as a piecewise constant or a piecewise linear continuous function. Higher order splines will not be modelled explicitly.
2. Constraints only involving system parameters are separated out so that no costate functions are computed.
3. Linear all-time constraints in control only, are converted internally to linear constraints on the control parameters. This means that no costates are needed.
4. More regularization features are added on top of the bounded variation regularization.
5. The software hooks to the optimization code FFSQP of Zhou and Tits is included.
6. Differential algebraic system equations have been tried, but theoretical problems exist as well as numerical ones. The area needs much more work, in particular the use of the latest DAE software.
7. A condition number option has been added to compute the condition number of the solution of the optimization problem. This should be considered under test, as its limitations are sometimes severe.
8. Methods to parametrize the independent variable (time) interval are incorporated. This is useful for finding approximations to the time of change of a bang-bang control. This will also allow a variable characteristic time for constraints whose characteristic time is less than the objective time horizon.
9. More examples have been added to show the new features.

This manual will now be available electronically only as a set of complete plain TEX files.

Areas of further research:

(a) Refinements to DAE problem management.

(b) Delay differential system dynamics.

(c) Ill-conditioning analysis.

(d) A comparison of methods used by MISER3 with large SQP methods where state equations are discretized as many equality constraints and state values at time points are parameters in the optimization.

(e) Automatic differentiation codes may enable some efficiencies for the user. These are currently being experimented with and analysed for efficiency.

(f) Interfacing with MATLAB and or a GUI for plotting the results and inputting the data of the problem.

Acknowledgements: Many researchers have contributed ideas to version 2 of the software, either directly or indirectly, via bug reporting of version 1, or, by reporting difficulties with a particular optimal control problem and the software. We would like to record our gratitude to all these people.

MISER3 — Version 3 (1999)

Version 3 makes some additions to the continuous case of MISER3. In particular, the case of *multiple characteristic times* for a canonical form is included, but only of a particular form. This involved some re-arranging of the internal workings of the code, so that limits from left and right at internal knots could be handled more easily. Once this was in place it was then possible to compute systems where the *state variables change discontinuously* at some points and the state differential equations change form on the subintervals defined by the *jump points*.

Two more hooks to other optimization software have been incorporated, namely MINOSS and NPSOL from Stanford's System Optimization Laboratory. Like FFSQP, users have to make their own arrangements to get the software. The reason for including hooks to other optimization software is that an optimal control optimization problem is usually ill-conditioned and a problem which cannot be computed with one optimization software may be computable using another. Each optimization code usually uses a different strategy in terms of linearization, merit function, interior or exterior approach for types of contraint, direction of search and line search technique. This allows different paths to an optimum from any given starting point.

Allowance has been made for efficient and accurate numerical derivatives to the functions defining a user problem. While this is not recommended, because of computation speed and accuracy deterioration, it does allow a shorter time between problem formulation and a first solution. Some problems, for example, bio-mechanical systems, with typical dynamics

$$\boldsymbol{M}(\boldsymbol{x})\,\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x},\boldsymbol{u}) \qquad \text{or} \qquad \dot{\boldsymbol{x}} = \boldsymbol{M}^{-1}(\boldsymbol{x})\boldsymbol{f}(\boldsymbol{x},\boldsymbol{u})$$

have very complicated derivatives for the user to compute, involving the gradient of the inverse of a matrix. These derivatives are more easily computed numerically, than computing

$$\frac{\partial\big(\boldsymbol{M}^{-1}\boldsymbol{f}(\boldsymbol{x},\boldsymbol{u})\big)}{\partial\boldsymbol{x}} = -\boldsymbol{M}^{-1}\frac{\partial\boldsymbol{M}}{\partial\boldsymbol{x}}\boldsymbol{M}^{-1}\,\boldsymbol{f}(\boldsymbol{x},\boldsymbol{u}) + \boldsymbol{M}^{-1}\frac{\partial\boldsymbol{f}}{\partial\boldsymbol{x}}.$$

For those difficult optimization cases, where the optimization has to be restarted often, allowance has been made for automatic cold restarts, without user intervention, until either two successive one iteration optimization runs(indicating failure usually) or a clear finish of the optimization. This cannot usefully be employed with MINOSS, as this optimization routine does not return the number of iterations completed. To help with saving results from these runs the latest 99 solution files, restart files, and plotting files are saved.

The version 2 code has been ported to run under MATLAB. While this runs slower than the FORTRAN code it computes small problems satisfactorily. A separate manual can be downloaded from the website `http://www.cado.uwa.edu.au/miser`

The **manual** is now available as a pdf file as well as the plain tex files.

CONTENTS

This table is not automatically produced, so any changes to font size or type or page size may render it inaccurate.

Chapter 1:

INTRODUCTION

The need for efficient general purpose software for solving optimal control problems has become increasingly pressing as the demand for optimal solutions to many engineering, biomedical and management problems has increased.

As opposed to mathematical programming problems which are static in nature, optimal control problems are dynamic. Their objective functionals (or cost functionals) are implicitly dependent on the control functions/parameters through some dynamical processes described by a set of ordinary differential equations. Some practical examples are:

(i) Control of a robotic arm to move it from one point in space to another with the least effort, while at the same time, avoiding interference with existing obstacles.

(ii) Construction of a highway to minimize the construction costs which include cutting, filling, land values, hauling of earth to and away from the construction site, etc., while meeting prescribed constraints on the slope and curvature.

(iii) Transfer of containers from ship to port via large container cranes, during which the swing of the load is to be minimized subject to limited capability on the part of the trolley drive motor and hoist motor.

(iv) Optimization of price and advertising expenses of a marketing firm subject to budget constraints.

(v) Optimization of an athletic movement, especially gymnastic sequences of movements.

One can easily name a multitude of such examples. Note that the independent variable t used in optimal control theory is not always necessarily the time variable. For instance, example (ii) above uses the spatial variable as the independent variable.

Historically, optimal control theory has been a highly specialized field. There exists an extensive list of optimal control literature [Ahmed (1988), Bryson and Ho (1969), Cesari (1983), Craven (1978a, 1995), Fleming and Rishel (1975), and, Pontryagin *et al.* (1962)] which furnishes rigorous necessary conditions for optimality. Unfortunately, except in some fortuitous cases or in the special class of linear quadratic problems, these elegant necessary conditions are difficult to use for generating the optimal solution.

Hitherto, general purpose software packages for solving optimal control problems with complex constraints have been rare. The difficulty arises because, conventionally, different types of constraints are handled differently. Furthermore, for infinite dimensional constraints such as the continuous state constraints, these constraints are often discretized thereby resulting in an unwieldy (in terms of computer memory storage) problem.

Recent research has evolved, from a somewhat different direction, a technique known as control parametrization, which can be used to approximate the optimal control problem by a constrained nonlinear programming problem. The approximate problem is, in turn, solvable by standard mathematical programming packages. As this is only an user manual, we shall not pursue any of the theoretical details here but refer the interested reader to Goh and Teo (1988a), Teo and Goh (1989a), Teo and Jennings (1989, 1991), and especially the book by Teo, Goh and Wong (1991) for an exhaustive and rigorous treatment of the concept of control parametrization.

As the software is meant to be used by non-specialist users who may not have a strong mathematical background nor exposure to the extensive research material, it is designed in such a way that any user with a course in Multivariable Calculus and some knowledge of FORTRAN programming will be able to use it without much difficulty. Nevertheless, users with a substantial background in optimal control theory and mathematical programming techniques will be better able to appreciate the intricacies of the program.

The central idea behind the software MISER3 is the concept of control parametrization [Goh and Teo (1988b), Hicks and Ray (1971), Siresina (1973, 1976), Siresena and Tan (1974), Teo and Clements (1985-6), Teo and Goh (1989a), Teo and Jennings (1989, 1991), Teo and Womersley (1983), Teo, Wong and Clements (1984), Teo and Wu (1984), Wong, Clements and Teo (1985), and, the book by Teo, Goh and Wong (1991)]. Essentially, this procedure approximates the control functions by a linear sum of basis functions, chosen for efficient computation and reality of control approximation. These basis functions are the piecewise constant functions on finite support or the piecewise linear continuous functions on finite support (the witches' hat functions). Thus, by appropriate reformulation, the optimal control problem is transformed into a nonlinearly constrained mathematical programming problem, and hence is solvable by existing efficient software packages for mathematical programming problems. For a discussion of the concept of control parametrization as implemented in the original MISER3 software the reader is referred to Jennings, Fisher, Teo and Goh (1991).

Before rushing into the computation, the user is warned that unless well-posed, not every optimal control problem is solvable. This is particularly so in the event when hard state constraints are present. On the other hand, the algorithm may converge to some local Kuhn-Tucker point, which may not even be a local minima, let alone the global minima. Thus, as is the case for many mathematical programming problems, the dependency on the initial guess can be a crucial one. Often, convergence is achieved when one initial guess is used but not for another. Also different initial guesses may converge to different local optimum points. Thus, a wise initial guess is a great advantage but unfortunately, it is not always easy to guess wisely. Some types of problems allow the formulation of good initial guesses of one problem by solving a more stable simpler problem, see for example Lee, Teo and Jennings (1998) for a robotic example.

Upon successful running of the MISER3 software, users should continue to exercise their intuition, experience and intelligent judgment before accepting the solution obtained. This is important because there may be hidden errors in the analytical formulation of the problem which may still result in seemingly successful execution of the program. In particular, the checking of constraints satisfaction to the desired accuracy is often warranted.

To close this section, we wish to refer the interested reader to Ahmed (1988), Bryson and Ho (1969), Cesari (1983), Craven (1978), Fleming and Rishel (1975), and, Pontryagin *et al.* (1962) for the theoretical aspects of optimal control and to Banks and Burns (1978), Bosarge and Johnson (1970), Dolezal (1981), Gonzalez and Miele (1978), Gruver and Sachs (1981), Mehra and Davis (1972), Miele (1980, 1987), Miele and Wang (1986a, 1986b, 1986c), Miele, Wang and Basapur (1986), Nedeljkovic (1985), Polak (1971), Polak and Mayne (1977), Sakawa (1981), Sakawa and Shindo (1980, 1982), Shindo and Sakawa (1985), Teo and Wong (1987), Teo, Wong and Clements (1984), Teo, Wong and Wu (1986), and, Wong and Teo (1990) for other computational techniques for optimal control problems.

An attempt to list all papers where MISER3 has been used is included in the references.

1.1 A NOTE ON NOTATION

Vectors and matrices are in bold type. A vector is normally a column vector. Let $\boldsymbol{x} = [x_1, \ldots, x_n]^\top$ be an n–dimensional vector, where the superscript $^\top$ denotes the transpose, and let F be a scalar valued function of $\boldsymbol{x}$. Then, the gradient of F with respect to $\boldsymbol{x}$ is a row vector defined by

$$\nabla F = \frac{\partial F}{\partial \boldsymbol{x}} = \left(\frac{\partial F}{\partial x_1}, \ldots, \frac{\partial F}{\partial x_n} \right)$$

If $\boldsymbol{F} = [F_1, \ldots, F_m]^\top$ is an m–dimensional vector valued function of $\boldsymbol{x}$, then the Jacobian matrix of $\boldsymbol{F}$ is represented naturally by

$$\nabla \boldsymbol{F} = \frac{\partial \boldsymbol{F}}{\partial \boldsymbol{x}} = \begin{bmatrix} \frac{\partial F_1}{\partial x_1} & \frac{\partial F_1}{\partial x_2} & \cdots & \frac{\partial F_1}{\partial x_n} \\ \frac{\partial F_2}{\partial x_1} & \frac{\partial F_2}{\partial x_2} & \cdots & \frac{\partial F_2}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial F_m}{\partial x_1} & \frac{\partial F_m}{\partial x_2} & \cdots & \frac{\partial F_m}{\partial x_n} \end{bmatrix}$$

that is, ∇F_i is the i–th row of the Jacobian matrix. There may be times when a gradient with respect to one vector variable of a function of two or more vector variables is needed. This will be indicated by putting a subscript on ∇ as in

$$\nabla_{\boldsymbol{x}} L(\boldsymbol{x}, \boldsymbol{\lambda}) = \frac{\partial L(\boldsymbol{x}, \boldsymbol{\lambda})}{\partial \boldsymbol{x}}.$$

1.2 OTHER METHODS AND SOFTWARE

Another major type of method for optimal control problems is to discretize both the control and state functions. Whether this is thought about as expanding these functions in terms of some basis functions or using the values of these functions at a pre-arranged set of time points, the ODE (and even DAE's) become a large number of equality constraints on the very large number of parameters. All-time state constraints are also simply treated. The ODE can also be treated in its ode form or in its integral form. By taking the costate equations in their integral form these values can be related to the Lagrange multipliers of the ODE constraints. Packages which can be made to do this include GAMS (Brooke and Kendrick (1992)), RIOTS (Schwartz and Polak (1996b)), COOPT (Serban and Petzold (2000)), (Gill *et al.* (2000)), (Petzold *et al.* (1997)).

Boeing has a group working on optimal control, see for example, Betts (2001).

The Chemical Engineering industry (where many of the earlier algorithms originated) has some practitioners developing software, see for example, Biegler *et al.* (2002), Cervantes and Biegler (1998), Logsdon and Biegler (1993), Tanartkit and Biegler (1997) and Tjoa and Biegler (1991).

A comparison of a simple form of these methods and MISER3 can be found in Benyah and Jennings (2001), where the condition numbers involved in the solution process are compared.

Another group headed by Helmut Maurer can be found at
`wwwmath.uni-muenster.de/u/maurer`