Exploring a multi-objective approach for optimal control problems *via* a direct multiple shooting method

Gisela C.V. Ramadas*
Research Center of Mechanical Engineering (CIDEM)
School of Engineering, Polytechnic of Porto, 4200-072 Porto, Portugal
Email: gcv@isep.ipp.pt

Edite M.G.P. Fernandes
ALGORITMI Center
University of Minho, 4710-057 Braga, Portugal
Email:emgpf@dps.uminho.pt

Ana Maria A.C. Rocha
ALGORITMI Center
University of Minho, 4710-057 Braga, Portugal
Email:arocha@dps.uminho.pt

M. Fernanda P. Costa
Centre of Mathematics
University of Minho, 4710-057 Braga, Portugal
Email:mfc@math.uminho.pt

Abstract
This paper explores the use of a multi-objective approach through the implementation of a numerical direct multiple shooting (MS) method to solve optimal control problems (OCP). When a direct MS method is used to solve the OCP, a set of 'continuity constraints' emerges and should be satisfied together with the other algebraic mixed states and control constraints. To minimize the objective function and satisfy all the constraint conditions, the finite-dimensional optimization problem is reformulated as a multi-objective problem with three objectives to be optimized. An illustrative example is included to show that the present methodology is worth pursuing.

Keywords: *Optimal Control, Direct Multiple Shooting, Multi-objective Optimization.*

1 Introduction

In this paper, we consider solving an optimal control problem (OCP) by a direct multiple shooting (MS) method. An OCP is a constrained optimization problem that has a set of dynamic equations as constraints. There are three types of OCP that differ in the formulation of the functional to be optimized. They are equivalent and it is possible to convert a problem in one of the forms into another one. Here, we consider the OCP in the *Mayer form*:

$$\begin{array}{rcl}\min\limits_{\mathbf{u}(t)\in U} J(\mathbf{y}(t),\mathbf{u}(t)) & \equiv & M(T,\mathbf{y}(T)) \\ \text{s.t.} \quad \mathbf{y}'(t) & = & \mathbf{f}(t,\mathbf{y}(t),\mathbf{u}(t))\,,\ t\in[0,T], \\ \mathbf{y}(0) & = & \mathbf{y}_0,\ \mathbf{y}(T)=\mathbf{y}_T, \\ 0 & = & h_e(t,\mathbf{y}(t),\mathbf{u}(t)), e\in E\,,\ t\in[0,T], \\ 0 & \geq & g_j(t,\mathbf{y}(t),\mathbf{u}(t)), j\in F\,,\ t\in[0,T], \end{array} \tag{1}$$

where $\mathbf{y} \in \mathbb{R}^s$ is the vector of state variables, $\mathbf{u} \in U \subset \mathbb{R}^c$ is the vector of control, U represents a class of functions (in particular functions of class C^1 and piecewise constant), $E = \{1,2,\ldots,m\}$ and $F = \{1,2,\ldots,l\}$ [5]. In the problem of Mayer, the functional is not an integral but a function M that depends in general on the dependent variables $\mathbf{y}$ and the final point of the t-domain T. For simplicity, we assume that the initial point of the t-domain is 0.

In the OCP we want to find $\mathbf{u}$ that minimizes the objective functional J subject to the dynamic system of ordinary differential equations (ODE) and the mixed states and control (equality and inequality) constraints.

Methods for solving OCP like (1) can be classified into indirect and direct methods. Indirect methods use the first-order necessary conditions from Pontryagin's maximum principle to reformulate the original problem into a boundary value problem [1]. On the other hand, direct methods solve the OCP directly. Direct methods transform infinite-dimensional OCP into a finite-dimensional optimization problem that can be solved by effective and well-established nonlinear programming (NLP) algorithms. All direct methods discretize the control variables but differ in the way they treat the state variables.

In a direct multiple shooting (MS) method the t-domain is partitioned into smaller subintervals and the system of ODE is integrated in each subinterval independently. Besides the control variables, the new *state start values* for each subinterval are the decision variables of the finite NLP problem [1, 3].

This paper explores the use of a multi-objective optimization (MOO) approach to solve the NLP problem, within a direct MS method for solving an OCP in the *Mayer form*. When a direct MS method is used to solve the OCP, a set of 'continuity constraints' emerges and should be satisfied together with the other algebraic mixed states and control constraints. To minimize the objective function and satisfy all the constraint conditions, the NLP problem is reformulated as a multi-objective problem with three objectives to be optimized. A set of near-optimal solutions is found by using a variant of the NSGA-II [4] to solve the resulting finite-dimensional MOO problem.

The paper is organized as follows. Section 2 introduces the direct MS method for solving the OCP in the *Mayer form* and Section 3 presents the proposed multi-objective formulation. An illustrative example is shown in Section 4.

2 Direct MS method

In a direct MS method, the controls are discretized in the NLP. On a specific grid defined by $0 = t_1 < t_2 < \cdots < t_{N-1} < t_N = T$, where $N-1$ is the total number of subintervals, the control $\mathbf{u}(t)$ is discretized, namely using a piecewise constant: $\mathbf{u}(t) = \mathbf{q}^i$, for $t \in [t_i, t_{i+1}]$ and $i = 1,\ldots,N-1$, so that $\mathbf{u}(t)$ only depends on the control parameters $\mathbf{q} = (\mathbf{q}^1, \mathbf{q}^2, \ldots, \mathbf{q}^{N-1})$. The dynamic system is solved by an ODE solver and the state variables $\mathbf{y}(t)$ are considered as dependent variables $\mathbf{y}(t,\mathbf{q})$.

In a MS method, the discretized controls and *state start values* at the nodes of the grid, $\mathbf{x}^i \in \mathbb{R}^s$, $i = 1,2,\ldots,N-1$, are the decision variables for the NLP solver [1]. After the discretization of the controls, the ODE system is solved on each shooting subinterval $[t_i, t_{i+1}]$ independently. The variables $\mathbf{x}^i$, $i = 1,2,\ldots,N-1$ are the initial values for the state variables for the $N-1$ independent initial value problems on the subintervals $[t_i, t_{i+1}]$:

$$\mathbf{y}'(t) = \mathbf{f}(t, \mathbf{y}(t), \mathbf{q}^i), \text{ for } t \in [t_i, t_{i+1}] \text{ and } \mathbf{y}(t_i) = \mathbf{x}^i.$$

To ensure continuity of the solution trajectories $\mathbf{y}^i(t; \mathbf{x}^i, \mathbf{q}^i)$, the *state start values* $\mathbf{x}^i$ should satisfy the 'continuity conditions': $\mathbf{y}^i(t_{i+1}; \mathbf{x}^i, \mathbf{q}^i) = \mathbf{x}^{i+1}$, $i = 1,\ldots,N-1$, as well as the initial value $\mathbf{x}^1 = \mathbf{y}_0$ and the final state constraints $\mathbf{x}^N = \mathbf{y}_T$.

3 Multi-objective formulation

Our proposal for solving the finite-dimensional NLP problem relies on a MOO formulation of the resulting NLP problem. Besides the objective function M, two other non-negative functions are to be minimized. The function VCC (see (2) below) emerges from the 'continuity constraints' violation (including initial state and final state constraints) and VAC (as shown below in (3)) comes up from the mixed state and control constraints violation. These two violation functions are handled separately. To measure the violation of the 'continuity constraints', initial state and final state constraints, the following non-negative function VCC is defined:

$$VCC(\mathbf{x},\mathbf{q}) = \sum_{l \in L} \sum_{i \in I} \left| y_l^i(t_{i+1}; \mathbf{x}^i, \mathbf{q}^i) - x_l^{i+1} \right| + \sum_{l \in L} \left| x_l^1 - y_{l_0} \right| + \left| x_l^N - y_{l_T} \right| \tag{2}$$

where $L = \{1, 2, \ldots, s\}$, noting that VCC is zero if the constraints are satisfied, otherwise is positive. To evaluate the algebraic equality and inequality constraints violation, a non-negative function VAC is used

$$VAC(\mathbf{x}, \mathbf{q}) \quad = \sum_{j \in J} \sum_{i \in I} \max \left\{0, g_j(\mathbf{y}^i(t; \mathbf{x}^i, \mathbf{q}^i), \mathbf{q}^i)\right\} + \sum_{e \in E} \sum_{i \in I} \left| h_e(\mathbf{y}^i(t; \mathbf{x}^i, \mathbf{q}^i), \mathbf{q}^i) \right|, \tag{3}$$

and similarly, $VAC = 0$ when the corresponding constraints are satisfied, and $VAC > 0$ otherwise.

The MOO problem is stated as follows:

$$\min_{(\mathbf{x}, \mathbf{q})} (VCC(\mathbf{x}, \mathbf{q}), VAC(\mathbf{x}, \mathbf{q}), M(T, \mathbf{y}(T))) \tag{4}$$

where state and control variables satisfy the ODE. The purpose of MOO is to optimize conflicting objectives simultaneously, although no unique solution that can simultaneously optimize all the objectives does exist. The optimal solution of a MOO problem is not a single solution but rather a set of potential solutions with objective function values that cannot be simultaneously improved. They define the set of non-dominated solutions, known as the set of Pareto-optimal solutions. This set is called Pareto-optimal set in the decision space and Pareto-front (PF) in the objective space (see [6] for details concerning MOO).

4 Illustrative example

The NLP problem in the form (4) is solved by the `gamultiobj` function from the MATLAB® (MATLAB is a registered trademark of the MathWorks, Inc.). This function implements a variant of NSGA-II [4]. The example to illustrate the behavior of the present MOO approach is: Find $u(t)$ that minimizes J (with $T = 3$ fixed),

$$\begin{array}{ll}
\min\limits_{u(t)} & J \equiv \displaystyle\int_0^T (y^2(t) + u^2(t))\, dt \\
\text{s.t.} & y'(t) = (1 + y(t))y(t) + u(t),\ t \in [0, T] \\
& y(0) = 0.05,\ y(T) = 0, \\
& |y(t)| \leq 1,\ t \in [0, T] \\
& |u(t)| \leq 1,\ t \in [0, T].
\end{array}$$

Figures 1(a) and 1(b) depict the bi-dimensional representation of the PF projections ($VCC \times VAC$) produced by the solver after phase 1 and then phase 2 respectively. The population size was set to 100. In the phase 1, the problem was solved with a maximum of $5Nvar$ generations and in the phase 2, 126 generations were required to obtain the solution with the accuracy defined by default in the `gamultiobj` function. $Nvar$ denotes the number of decision variables of the problem. Based on $N = 11$, $Nvar = 2N + (N - 1)$. The NLP problem has two state variables and one control variable. State variable $y_1 = y$ and y_2 was added to transform the *Lagrange form* of the problem into the *Mayer form*. The average distance measure of the solutions on the PF was 0.0119522 and the spread measure of the PF was 0.100187. We note that the initial population of phase 2 is the final population of phase 1.

From the PF, we selected the point that has the smallest VCC value. The optimal objective values are $VCC = 3.14072$, $VAC = 0$, $M = 0.649469$. The optimal states trajectory and control for the selected point are shown in Figures 2(a) and 2(b) respectively.

Acknowledgments

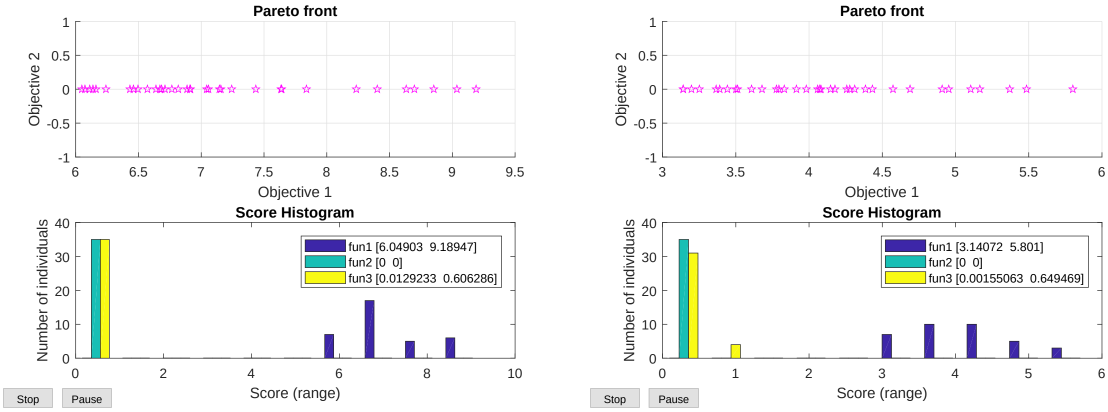
We acknowledge the financial support of CIDEM, R&D unit funded by the FCT - Portuguese Foundation for the Development of Science and Technology, Ministry of Science, Technology and Higher Education, under the Project UID/EMS/0615/2016, and FCT within the Projects Scope: UID/CEC/00319/2019 and UID/MAT/00013/2013.

References

[1] Assassa, F., Marquardt, W.: Dynamic optimization using adaptive direct multiple shooting. Computers and Chemical Engineering, 60, 242–259 (2014)

[2] Biegler, L.: Nonlinear Programming: Concepts, Algorithms, and Applications to Chemical Processes. Mos-SIAM Series on Optimization. Society for Industrial and Applied Mathematics (2010)

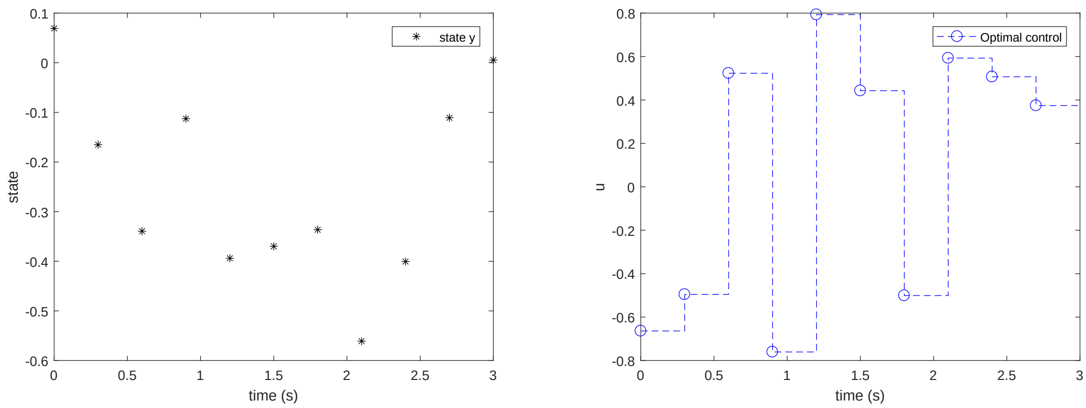
[3] Bock, H.G. and Plitt, K.J.: A multiple shooting algorithm for direct solution of optimal control problems. IFAC Proceedings Volumes, 17(2), 1603–1608 (1984)



(a) PF from phase 1

(b) PF after phase 2

Figure 1: PF after phase 1 and phase 2



(a) State trajectory for $N = 11$

(b) Optimal control for $N = 11$

Figure 2: State trajectory and optimal control

[4] Deb, K., Pratap, A., Agrawal, S., Meyarivan, T.: A fast and elitist multi-objective genetic algorithm: NSGA-II. IEEE Transaction on Evolutionary Computation, 6, 182–197 (2002)

[5] Frego, M.: Numerical Methods for Optimal Control Problems with Applications to Autonomous Vehicles. Ph.D. Thesis, University of Trento (2014)

[6] Miettinen, K.: Nonlinear Multiobjective Optimization, Kluwer Academic Publishers (1999)