Journal of Process Control 24 (2014) 1106–1120

Contents lists available at ScienceDirect

Journal of Process Control

journal homepage: www.elsevier.com/locate/jprocont

Nonlinear model predictive control using trust-region derivative-free optimization

Jon S. Dæhlen[a,*], Gisle Otto Eikrem[b], Tor Arne Johansen[a]

[a] Department of Engineering Cybernetics, Norwegian University of Science and Technology, Trondheim, Norway
[b] Statoil Research Center, Trondheim, Norway

ARTICLE INFO

Article history:
Received 2 September 2013
Received in revised form 23 March 2014
Accepted 11 April 2014
Available online 21 June 2014

Keywords:
Derivative-free optimization
DFO
MPC
NMPC
Model-predictive control
Non-linear model-predictive control
Sub-sea oil separator
Trust-region methods
Derivative-free MPC
Derivative-free NMPC

ABSTRACT

Gradient-based optimization may not be suited if the objective and constraint functions in a nonlinear model predictive control (NMPC) optimization problem are not differentiable. Some well-known derivative-free optimization (DFO)-algorithms are investigated, and a novel warm-start modification to the Wedge DFO-algorithm is proposed. Together with a gradient-based SQP-algorithm these are applied to the NMPC problem and compared in a single-shooting NMPC formulation to a subsea oil–gas separation process. The findings are that DFO is significantly more robust against the numerical issues, compared to a gradient-based SQP tested. Moreover, the warm-start modification reduces the computational complexity.

1. Introduction

In the later years nonlinear model predictive control (NMPC) has attracted much attention, as the use of non-linear models improve the controller performance on highly non-linear systems and allows for operation over wider range.

A common method for solving an NMPC problem is by linearizing the model about the previous optimal trajectory, referred to as the nominal trajectory. This approximate model can be optimized as in linear model predictive control (MPC), and a new linearization is performed in the next time step about the updated nominal trajectory. This however makes the model only valid for small perturbations from its nominal trajectory, and e.g. if the set-point is changed rapidly, then more accurate prediction of larger transients may be desirable. Today it seems that state-of-the-art to overcome this is sequential-quadratic programming (SQP) [13] and interior-point (IP)-methods [8]. Common for both is that they iteratively linearise the model and the constraints until convergence is achieved, which in turn requires the gradient of the model. If the model is known explicitly, an off-line symbolic differentiation can be performed and the gradient and possibly the Hessian can be implemented.

However, this may not always be the case, as the model may not be available in an appropriate programming language, it can be made up of a mix of subroutines from different programming languages or the source can simply be unavailable. This will render the symbolic software hard or impossible to apply. The model may not even be continuous or smooth, as logic operators are common, and the model may not be explicitly available at all as it may be embedded in a numerical simulation software [16]. In these cases it is likely that the most common and intuitive approach is to retrieve the gradient from finite-differences. This method is however known to be sensitive towards numerical issues [19]. When performing the simulation it can be desirable to use a variable-step ordinary differential equation (ODE)-solver. This can speed up the simulation time significantly, however it is known that this tends to induce numerical noise and discontinuities, which possibly can be amplified through finite-differences and thus compromise the performance of a gradient-based NMPC optimization. This motivates for investigating optimization methods not requiring derivatives, namely derivative-free optimization (DFO).

* Corresponding author. Tel.: +47 93256732.
E-mail addresses: jondaeh@gmail.com, jda@icd.no (J.S. Dæhlen).

http://dx.doi.org/10.1016/j.jprocont.2014.04.011
0959-1524/

DFO has been applied in NMPC in several occasions, by use of genetic algorithms (GA) [5,17,1,24,4]. However in these cases the goal has been to overcome extreme non-linearities. Although GA is known to be well-suited towards noise and discontinuities it also tends to have slow convergence rate. Tlili et al. [26] used the Nelder-Mead simplex method for solving a highly non-linear NMPC problem and reported that it was 10 times faster than GA, and Sadrieh and Bahri [23] used parallel computing on a Graphics Processing Unit (GPU) GPU to obtain even faster computation. These did however suggest to try other optimization methods and more practical problems. Koller and Ulbrich [11] reported that a Derivative-free Trust-region method (DFTRM) out-performed the Nelder-Mead method in an optimal-control problem, however the exact nature of the optimization problem was not stated. DFTRM is also known to require less function evaluations than algorithms as Nelder-Mead and GA [16], thus investigating the use of these methods in NMPC aimed at controlling a realistic process seems appropriate.

The main contribution of the paper is an evaluation of the performance and robustness of DFTRM for NMPC, performed by a case-study of a challenging sub-sea oil and gas separation process. Another significant contribution is a novel warm-start procedure that aims to improve computational performance, which is presented and evaluated using the case-study.

The rest of the paper will be structured as follows:

- Section 2 briefly presents the single-shooting formulation of the NMPC problem.
- Section 3 gives a short description of DFTRM, investigates some existing and well-known DFO algorithms, and a novel warm-start procedure is presented.
- The case study of an industrial sub-sea crude-oil separation unit is presented in Section 4. The objectives of the controller is also presented, and the resulting NMPC problem is analysed. The closed-loop system is simulated using DFO algorithms, a gradient-based SQP-algorithm and the warm-start modification. The results are summarized and discussed.
- Section 5 concludes the findings.

For more details on the algorithms, implementation and the simulations, the reader is referred to [7].

2. Nonlinear model predictive control formulation

The single-shooting approach to NMPC is based on parametrizing the objective function of the optimization problem in the input sequence(i.e. the Manipulated variables (MV)) and the current state of the system. This approach is well suited when the prediction model of the plant is a simulation, i.e. the solution to an initial value problem using an ODE-solver to a piecewise constant control input sequence $[u(0), \ldots, u(k-1)]$

$$[x(1), \ldots, x(\hat{K})] = \ell(u(0), \ldots, u(K-1), x(0), K) \tag{1}$$

where K is the discrete length of the control horizon and $x(0)$ is the current state of the plant. The left-hand side of (1) is then the resulting trajectory of the states of the model from time step 1 to $\hat{K}$. Note that if a variable-step solver is used, K is likely to be different from $\hat{K}$. Considering an objective function on the form $J(x(1), \ldots, x(\hat{K}), u(0), \ldots, u(K-1))$, which is common in NMPC, the complete single-shooting NMPC problem can be stated

$$\min_{u\in\mathbb{R}^n} \; J(\ell(u(0),\ldots,u(K-1),x(0),K),u(0),\ldots,u(K-1),x(0)) \tag{2a}$$

$$:= J(u;x(0);K) \tag{2b}$$

$$\text{s.t}$$

$$x_{min} \le \ell(u(0),\ldots,u(K-1),x(0),K) \le x_{max} \tag{2c}$$

$$u_{min} \le u(k) \le u_{max}, k=0,\ldots,K-1 \tag{2d}$$

$$\Delta u_{min} \le \Delta u(k) \le \Delta u_{max}, k=0,\ldots,K-1 \tag{2e}$$

The definition of $J(u;x(0);K)$ is to illustrate that there are only the input blocks that are manipulated variables (MV). Note that because of input blocking, the number of MV will be less than K, however this is not considered in (2) for simplicity of notation. Clearly neither the objective function not the constraints may not be convex. Applying SQP or IP methods will somehow need to make a linear and/or quadratic approximation of the objective function and the constraints to approximate (2) with a Quadratic Programming (QP) problem, thus requiring the gradient of $J(u;x(0);K)$ and the constraints. As mentioned initially, the prediction model may not be differentiable, thus this process may become troublesome.

3. Derivative-free trust-region methods

This section is aimed at giving a working understanding of DFTRM, and a description of the main differences between the algorithms which is used in this study. A more extensive introduction can be found in [6], and in papers referenced in the subsection for each algorithm.

DFO algorithms comes in a vast number of types and varieties. A common characteristic of these methods are that they start with an initial set of samples $Y \subset \mathbb{R}^n$ of the objective function $f: \mathbb{R}^n \mapsto \mathbb{R}^1$ in order to approximate it, and creates new points by trying to figure out where the optimum of $f(x)$ is by looking at the positions of y^i, $i \in [1, m]$ and the resulting function values $f(y^i)$, $i \in [1, m]$. When applied to NMPC, $f(x)$ is typically constructed from $J(u;x(0);K)$ and (2c)–(2e) applied through penalty functions.

Algorithm 1 describes a simplified DFTRM, which is based on making a polynomial model $q(x)$ of the true objective function $f(x)$ around the current iterate x_k, with k being the iteration index.

The algorithm needs enough samples initially to build the model $q(x)$ in the first iteration. There exist several different interpolation techniques, which requires different number samples. A fully determined model requires $m = \frac{1}{2}(n+1)(n+2)$ samples, n being the number of decision variables. Because of the large computational resources required for this, it is a challenge to handle this in many real-time applications.

In step 2 the polynomial model $q(x)$ is made from Y and its corresponding function values by solving a set of equations to find the unknown coefficients in $q(x)$. In this paper, a quadratic model is chosen, since optimality conditions are then simple, and analysis of the optimization problem presented in Section 4 suggests that the objective resembles a quadratic function.