Automatica 55 (2015) 226–235

Contents lists available at ScienceDirect

Automatica

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

A dual gradient-projection algorithm for model predictive control in fixed-point arithmetic☆

Panagiotis Patrinos, Alberto Guiggiani, Alberto Bemporad

IMT - Institute for Advanced Studies, Piazza San Ponziano 6, 55100 Lucca, Italy

ARTICLE INFO

Article history:
Received 12 March 2013
Received in revised form
25 February 2015
Accepted 27 February 2015
Available online 31 March 2015

Keywords:
Embedded systems
Convex optimization
Predictive control

ABSTRACT

Although linear Model Predictive Control has gained increasing popularity for controlling dynamical systems subject to constraints, the main barrier that prevents its widespread use in embedded applications is the need to solve a Quadratic Program (QP) in real-time. This paper proposes a dual gradient projection (DGP) algorithm specifically tailored for implementation on fixed-point hardware. A detailed convergence rate analysis is presented in the presence of round-off errors due to fixed-point arithmetic. Based on these results, concrete guidelines are provided for selecting the minimum number of fractional and integer bits that guarantee convergence to a suboptimal solution within a pre-specified tolerance, therefore reducing the cost and power consumption of the hardware device.

1. Introduction

Model Predictive Control (MPC) technology is widely popular in many industrial applications due to explicit performance optimization, and its straightforward handling of constraints on inputs, outputs and states (Bemporad, 2006; Mayne & Rawlings, 2009). An MPC controller relies on solving a Quadratic Program to minimize input efforts and the difference between predicted outputs and desired set-points. The fact that a QP needs to be solved within each sampling period has initially limited the diffusion of MPC technologies to low-bandwidth applications where high computational resources are available, as in the chemical and refinery industries. However, in the last years an increasing interest in embedded MPC solutions is spreading in many other industries, such as automotive and aerospace.

Embedding MPC on a hardware platform poses quite a few challenges, both from a system-theoretic and an optimization point of view. Specifically, the main requirements that make a QP solver suitable for embedded MPC are the following: (a) the algorithm should be simple enough to be implemented on simple hardware platforms; (b) one must be able to compute a bound on its worst-case execution time for computing a (reasonably good) solution; (c) stability and invariance guarantees for the resulting closed-loop system must be provided despite suboptimality and/or infeasibility of the solution; (d) the algorithm should be robust to low precision arithmetic, i.e., the effect of round-off errors should be small, no overflow should occur, and one should be able to determine *a priori* the behavior of the algorithm under such hypotheses.

Ling et al. detailed in Ling, Yue, and Maciejowski (2006) an FPGA implementation of an interior-point method for solving the QP problem, showing that the "MPC-on-a-chip" idea is indeed viable. Later, Knagge et al. proposed an active-set QP solver for ASIC and FPGA (Knagge, Wills, Mills, & Ninness, 2009), and tested it for MPC control of nonlinear systems. A "QP-on-a-chip" controller implemented on FPGA with an iterative linear solver was tested in hardware-in-the-loop experiments in Hartley et al. (2012).

All of the solvers proposed in such contributions require *floating-point* numbers. However, when trying to minimize computational effort, power consumption, and chip size, a great positive impact is given by the choice of *fixed-point* number representation (Kerrigan, Jerez, Longo, & Constantinides, 2012). Nevertheless, this significant improvement in performance comes at the price of a reduced range in which numbers can be represented and round-off errors (Wilkinson, 1994). Because of this, algorithms that perform well in floating-point may perform much worse (even completely wrongly) in fixed-point. Therefore, additional challenges arise when dealing with fixed-point arithmetic, mainly studying round-off error accumulation during algorithm iterations, and establishing bounds on the magnitude of the computed variables to avoid overflows. In Jerez, Constantinides, and Kerrigan (2012) an implementation of a modified interior-point solver in

☆ The material in this paper was presented at the 2013 European Control Conference, July 17–19, 2013, Zurich, Switzerland. This paper was recommended for publication in revised form by Editor Frank Allgöwer.

E-mail addresses: panagiotis.patrinos@imtlucca.it (P. Patrinos), alberto.guiggiani@imtlucca.it (A. Guiggiani), alberto.bemporad@imtlucca.it (A. Bemporad).

http://dx.doi.org/10.1016/j.automatica.2015.03.002
0005-1098/

fixed-point is presented. The authors focus on the solution of the linear system required in each algorithm iteration, and propose a preconditioning technique tailored to prevent overflow errors as well as a detailed analysis of the effects of the round-off error.

Recently, the use of first-order methods, and in particular fast gradient methods developed by Nesterov (2004), has been advocated as a viable candidate for embedded optimization-based control (Bemporad & Patrinos, 2012; Patrinos & Bemporad, 2012; Richter, Jones, & Morari, 2009; Richter, Morari, & Jones, 2011). These methods can compute a suboptimal solution in a finite number of iterations, which can be bounded *a priori*, and they are simple enough (usually requiring only matrix–vector products) for hardware implementation. In particular, the accelerated DGP method proposed in Bemporad and Patrinos (2012) and Patrinos and Bemporad (2012), called GPAD, can be applied to linear MPC problems with general polyhedral constraints and with guaranteed global *primal* convergence rates. In Rubagotti, Patrinos, and Bemporad (2014) results of Patrinos and Bemporad (2012) are exploited to show how GPAD can be used in MPC to provide invariance, stability and performance guarantees in a finite number of iterations for the closed-loop system.

1.1. Contribution

In this work, we propose a DGP method, which can be seen as a simplified (non-accelerated) version of GPAD, specifically tailored for fixed-point implementation. The main contribution of this work is that we provide a detailed convergence rate and asymptotic error analysis in terms of *primal cost and primal feasibility* in the presence of round-off errors due to fixed-point arithmetic, thus addressing successfully the last of the requirements described previously for embedded optimization-based control. In addition to that, we give specific guidelines on the number of fractional bits that certify the convergence to a target suboptimal solution, as well as on the number of integer bits to avoid overflow errors. The machinery we use to perform the analysis is based on the notion of the *inexact oracle* proposed by Devolder, Glineur, and Nesterov (2013). However, directly applying the results of Devolder et al. (2013) to our dual gradient projection method would only provide us with convergence rate estimates about the quality of the *dual and not the primal iterates* of the algorithm.

The reason for limiting the analysis to the non-accelerated version of GPAD is that accelerated methods suffer from error accumulation, as shown in Devolder et al. (2013). In Nedelcu and Necoara (2012) and Nedelcu, Necoara, and Dinh (2013) the authors analyze the convergence rate of inexact gradient augmented Lagrangian methods for constrained MPC, where the source of inexactness comes from suboptimal solution of the so called inner problem. In the present work, the source of inexactness comes from round-off errors due to the fixed-point implementation.

1.2. Structure of the paper

After introducing some notation at the end of this section and motivating the work in Section 2, in Section 3 we give general theoretical results when a gradient projection (GP) algorithm runs with an inexact oracle. In Section 4 an inexact DGP method is applied to a modified version of the dual problem and its convergence rate with respect to primal suboptimality and infeasibility is analyzed. In Section 5, the general results of the proposed inexact DGP method are applied to the case of QP based on a fixed-point implementation. Simulation results and experiments on low-cost hardware boards are presented in Section 6. Finally, conclusions are drawn in Section 7.

The main technical contribution of this paper has appeared in Patrinos, Guiggiani, and Bemporad (2013) without providing the proofs of the theoretical results, that are provided here in full detail.

The notation adopted throughout the paper is standard. Let $\mathbb{R}$, $\mathbb{N}$, $\mathbb{R}^n$, $\mathbb{R}^{m\times n}$ denote the sets of real numbers, nonnegative integers, column real vectors of length n, and m by n real matrices, respectively. The transpose of a matrix $A \in \mathbb{R}^{m\times n}$ is denoted by A'. For any nonnegative integers $k_1 \leq k_2$, the finite set $\{k_1, \ldots, k_2\}$ is denoted by $\mathbb{N}_{[k_1,k_2]}$. For $z \in \mathbb{R}^n$, $\Pi_Z(z)$ denotes its Euclidean projection on the set $Z \subseteq \mathbb{R}^n$, while $[z]_+$ denotes its Euclidean projection on the nonnegative orthant, i.e., the vector whose ith coordinate is $\max\{z_i, 0\}$. For a vector $z \in \mathbb{R}^n$, $\|z\|$ and $\|z\|_\infty$ denote the Euclidean and infinity norm of z respectively, while if $A \in \mathbb{R}^{m\times n}$, $\|A\|$ denotes the spectral norm of A (unless otherwise stated).

2. Motivation

When performing computations on low-cost, low-power embedded devices, the adoption of a fixed-point number representation can have a great positive impact in terms of computational speed. However, this comes at the price of a reduced precision and a reduced range when compared to floating-point representation, leading to the occurrence of round-off and overflow errors.

Suppose that an algorithm is running on a fixed-point hardware with a scaling factor 2^{-p}, where $p \in \mathbb{N}_+$ is the number of fractional bits, and assume that real numbers are represented in fixed-point by rounding to the closest value. Therefore, the resolution (i.e., the smallest representable non-zero magnitude) of a fixed-point number is equal to $2^{-(p+1)}$.

It is obvious that addition and subtraction do not result in any loss of accuracy due to rounding. However, multiplication can suffer from rounding. In specific, multiplying two scalars $\zeta = \gamma\xi$ leads to the fixed-point representation $\mathtt{fi}(\zeta)$ of ζ, with $|\zeta - \mathtt{fi}(\zeta)| \leq 2^{-(p+1)}$.

For $x, y \in \mathbb{R}^n$ let $\mathtt{fi}(x'y) \triangleq \sum_{i=1}^n \mathtt{fi}(x_i y_i)$. Then the round-off error for the inner product of x and y can be bounded as follows:

$$|x'y - \mathtt{fi}(x'y)| \leq 2^{-(p+1)} n. \tag{1}$$

If A is an $m \times n$ matrix and x is an n-vector, then

$$\|Ax - \mathtt{fi}(Ax)\|_\infty \leq 2^{-(p+1)} n. \tag{2}$$

Quadratic Programming algorithms based on Gradient Projection method require, at each iteration, the computation of the gradient for the cost function. In a fixed-point architecture, instead of the exact gradient $\nabla\Phi(\cdot)$, we have access to an approximate formulation $\tilde{\nabla}\Phi(\cdot)$.

Convergence proofs have therefore to be reformulated in order to take into account of this approximation. In addition to that, it is of interest to find direct links between the fixed-point precision and bounds on the gradient error, as well as solution quality. Finally, bounds on the magnitude of all the variables are required such that the number of integer bits can be chosen to avoid the occurrence of overflow errors. All these topics will be covered in the next sections.

3. Inexact gradient projection

Consider the problem

$$\begin{aligned} &\text{minimize} \quad \Phi(y) \\ &\text{subject to } y \in Y, \end{aligned} \tag{3}$$

where Y is a nonempty closed convex subset of $\mathbb{R}^m$, and $\Phi : \mathbb{R}^m \to \mathbb{R}$ is convex, L_Φ-smooth, i.e., there exists a $L_\Phi > 0$ such that

$$\|\nabla\Phi(y) - \nabla\Phi(w)\| \leq L_\Phi \|y - w\|, \quad y, w \in \mathbb{R}^m.$$