Control Engineering Practice 47 (2016) 60–69

Contents lists available at ScienceDirect

Control Engineering Practice

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

Explicit MPC: Hard constraint satisfaction under low precision arithmetic

Andrea Suardi [a,*], Stefano Longo [b], Eric C. Kerrigan [a,c], George A. Constantinides [a]

[a] Department of Electrical and Electronic Engineering, Imperial College London, London SW7 2AZ, UK
[b] Centre for Automotive Engineering, Cranfield University, Cranfield MK43 0AL, UK
[c] Department of Aeronautics, Imperial College London, London SW7 2AZ, UK

ARTICLE INFO

Article history:
Received 30 April 2015
Received in revised form
5 December 2015
Accepted 11 December 2015
Available online 29 December 2015

Keywords:
Embedded systems
Predictive control
Robust control

ABSTRACT

MPC is becoming increasingly implemented on embedded systems, where low precision computation is preferred either to reduce costs, speedup execution or reduce power consumption. However, in a low precision implementation, constraint satisfaction cannot be guaranteed. To enforce constraint satisfaction under numerical errors, we adopt tools from forward error analysis to compute an error bound on the output of the embedded controller. We treat this error as a state disturbance and use it to inform the design of a constraint-tightening robust controller. The technique is validated via a practical implementation on an FPGA evaluation board.

1. Introduction

Since the widespread use of single- and double-precision floating-point arithmetic in computer architectures, control system designers routinely start with the assumption that computation is performed with infinite numerical precision. The consequence is that the two activities of control system design and its implementation are often decoupled. This is safe for simple and well-understood algorithms. The control engineer worries about high-level issues, such as closed-loop performance, while the software engineer worries about implementation issues, such as code efficiency and timing (Teich, 2012).

In addition to high numerical precision, other factors such as high clock speed and small packaging have become standard features of modern embedded systems processors. Such advances in digital electronics (together with the development of sophisticated algorithms) have facilitated the spread of computationally heavy control schemes in low-cost applications with relatively fast dynamics. The embedded control community has started exploring the hardware design dimension in order to reduce hardware costs and increase execution speed by, for example, implementing algorithms with finite and low precision arithmetic (Constantinides, 2009; Jerez et al., 2014).

It is well-known that low precision, especially if implemented in fixed-point, allows for much simpler circuits and greater computational speeds (Patterson & Hennessy, 1990). All of the above is at the expense of increased numerical errors that cannot and should not be ignored. There is a surprisingly small amount of theory for the design of such computer-based control systems. These issues could be considered as part of the emerging science called cyber-physical systems theory (Wolf, 2009). Cyber-physical systems are integrations of computation with physical processes and therefore would also embrace the problem of control algorithm performance under numerical errors.

Model Predictive Control (MPC) is a powerful control scheme that, due to the necessity of solving an optimization problem every sampling instant, has only recently found application outside the process industry. One of the often ignored drawbacks of MPC, however, is its sensitivity to numerical errors (Hasan, Kerrigan, & Constantinides, 2013). The use of different discretization methods has been proven to be an advantage when working with low precision (Longo, Kerrigan, & Constantinides, 2014). Methods to avoid variable overflow have been proposed by constraining their ranges with carefully selected scaling methods (Jerez, Constantinides, & Kerrigan, 2015). However, for these approaches, stability and constraint satisfaction are not guaranteed and, in practice, the only solution to this problem is extensive simulation analysis.

In this paper, we extend the basic idea presented in Suardi, Longo, Kerrigan, and Constantinides (2014) and we propose a method to guarantee hard constraint satisfaction of an explicit

* Corresponding author.
E-mail addresses: a.suardi@imperial.ac.uk (A. Suardi), s.longo@cranfield.ac.uk (S. Longo), e.kerrigan@imperial.ac.uk (E.C. Kerrigan), g.constantinides@imperial.ac.uk (G.A. Constantinides).

http://dx.doi.org/10.1016/j.conengprac.2015.12.005
0967-0661/

MPC scheme (Bemporad, Morari, Dua, & Pistikopolous, 2002; Kvasnica & Fikar, 2010) when the algorithm is implemented on a platform using finite and low precision arithmetic. Compared to Suardi et al. (2014), this paper adds a detailed algorithmic presentation, unveiling the machinery required for the robust controller design, adds a detailed implementation on a FPGA platform and provides experimental results. Furthermore, this paper uses the design tools presented in Suardi, Kerrigan, and Constantinides (2015) and Suardi (2014). The idea is to quantify the maximum error made by the processor when evaluating the control policy. This is achieved by applying techniques from forward error analysis (Higham, 2002) to the explicit MPC controller. Considering the error as an additive disturbance to the plant dynamics, a controller that is robust to such a disturbance is designed. The resulting controller will therefore be robust against its own finite-precision implementation in a true cyber-physical sense. The proposed method requires the offline solution of an optimization problem, which is non-trivial but possible to automate. The validity of the method has been tested experimentally with a hardware-in-the-loop test ring where the controller has been implemented in a Xilinx Zynq Field-Programmable Gate Array (FPGA) platform. In Section 2 the explicit MPC problem is formulated. In Section 3 the robust controller design methodology is presented. The procedure requires the analytical computation of error bounds, which is described in Section 4. The experimental validation setup is presented in Section 5 and test results of guaranteed robustness and implementation efficiency are discussed in Section 6.

2. Problem setup

Let us assume that we want to find a discrete-time feedback control law

$$u_k := \kappa(x_k), \tag{1}$$

where $\kappa: \mathbb{R}^n \to \mathbb{R}^m$ is designed to stabilize and guarantee some performance for the discretized plant

$$x_{k+1} = Ax_k + Bu_k, \tag{2}$$

where n is the number of states, m the number of inputs, and $A \in \mathbb{R}^{n\times n}$ and $B \in \mathbb{R}^{n\times m}$ are the discretized plant matrices. For simplicity we assume that we want to regulate the system from the current state x_0 to the origin. State and input variables are subject to the constraints

$$x_k \in \mathbb{X}, \quad u_k \in \mathbb{U}, \tag{3}$$

where $\mathbb{X}$ and $\mathbb{U}$ are the polyhedral sets

$$\mathbb{X} = \{x \in \mathbb{R}^n : M_x x \le k_x\}, \tag{4a}$$

$$\mathbb{U} = \{u \in \mathbb{R}^m : M_u u \le k_u\}, \tag{4b}$$

containing the origin in their interior. The constraints on state and input may be physical or chosen by design. The finite horizon constrained linear quadratic regulator problem with horizon N is defined as

$$\min_{u_0,u_1,\ldots,u_{N-1}} \quad x_N' P x_N + \sum_{i=0}^{N-1} (x_i' Q x_i + u_i' R u_i), \tag{5a}$$

$$\text{s.t.} \quad x_{k+1} = Ax_k + Bu_k, \quad (x_0 \text{ given}), \tag{5b}$$

$$x_k \in \mathbb{X}, \quad u_k \in \mathbb{U}, \tag{5c}$$

$$x_N \in \mathbb{X}_f, \tag{5d}$$

$$\forall\, k \in \{0, 1, \ldots, N-1\}, \tag{5e}$$

where $P, Q \in \mathbb{R}^{n\times n}$ are positive semidefinite matrices, $R \in \mathbb{R}^{n\times m}$ is a positive definite matrix, N is the length of the prediction horizon, $\mathbb{X}_f$ is the set in which the terminal state x_N is constrained to lie and, at sample instant k, the state vector $x_k \in \mathbb{R}^n$ is either measured or estimated. Solving (5) yields the optimal open-loop input sequence $u_0^*(x_0), u_1^*(x_0), \ldots, u_{N-1}^*(x_0)$, where, as per standard MPC, the first element u_0^* is applied and the optimization is repeated at every time step in a receding horizon control fashion. MPC relies on a successive solution of the optimization problem in (5). Such an optimization problem can be expressed as a parametric Quadratic Programming (QP) problem for varying parameters x_0 defined as

$$\min_U \quad x_0' Y x_0 + U' H U + x_0' T U \tag{6a}$$

$$\text{s.t.} \quad GU \le W + Ex_0, \quad H > 0 \tag{6b}$$

where the vector $U = [u_0' \; u_1' \ldots u_{N-1}']' \in \mathbb{R}^{Nm}$ is the vector of decision variables and matrices H, T, Y, G, W and E are easily obtained from Q, R and P and by substituting $x_i = A^i x_0 + \sum_{j=0}^{i-1} A^j B u_{i-1-j}$ (Maciejowski, 2002).

When N, n, and m are small, we can compute an MPC feedback controller κ explicitly by solving a multi-parametric optimization problem. In parametric programming, the goal is to find the solution of (6) for a range of parameters values, or equivalently, the closed form solution $x_0 \mapsto U(x_0)$ of (6) for any feasible x_0. This problem could be solved by using the freely available Multi Parametric Toolbox (MPT) (Herceg, Kvasnica, Jones, & Morari, 2013) written for MATLAB® and the Model Predictive Control Toolbox™ embedded into MATLAB®. The resulting κ is a continuous piecewise affine (PWA) function defined over a polyhedral partition of the state space. Therefore, computing (1) requires:

1. the solution of a point location problem to determine in which polytope — defined by a linear inequality $(Hx_k \le k)$ — the current state x_k belongs to
2. the evaluation of a control law of the form

   $$u_k = Fx_k + g \tag{7}$$

   associated with the selected region in step 1.

A variety of algorithms have been proposed to solve the point location problem, since this is the most time-consuming task (Jones, Grieder, & Rakovic, 2006; Monnigmann & Kastsian, 2011; Storace & Poggi, 2011; Tøndel, Johansen, & Bemporad, 2003). Such algorithms range from simple ones (a *sequential* search through the regions of the partition) to more complex ones where the region is found via a binary *search tree*. In either case, the solution of the point location problem and the evaluation of the control law are operations that have to be performed online on the target hardware.

If infinite-precision arithmetic was available, the control action u_k could be computed exactly without introducing any numerical errors, hence complying with the QP problem theoretical guarantees such as constraint satisfaction. However, computing u_k in a processor that works with finite precision (typically any processor) results in the introduction of an error. Such an error is the