Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnLine 50-1 (2017) 11595–11600

Reducing Memory Footprints in Explicit Model Predictive Control using Universal Numbers

Deepak Ingole * **Michal Kvasnica** * **Himeshi De Silva** ** **John Gustafson** **,***

* *Faculty of Chemical and Food Technology, Slovak University of Technology in Bratislava, Slovakia, (e-mail:{deepak.ingole, michal.kvasnica}@stuba.sk).*
** *Department of Computer Science, National University of Singapore, Singapore, (e-mail: himeshi@comp.nus.edu.sg).*
*** *Agency for Science, Technology and Research, Computational Resource Centre, Singapore, (e-mail: john-gustafs@acrc.a-star.edu.sg).*

Abstract: Explicit Model Predictive Control (MPC) is an effective alternative to reduce the on-line computational demand of traditional MPC. The idea of explicit MPC is to pre-compute the optimal MPC feedback law off-line and store it in a form of look-up table which is to be used in on-line phase. One of the main bottlenecks in an implementation of explicit MPC is memory required to store optimal solutions. This limit its applicability to systems with few states, small number of constraints, and short prediction horizons. In this paper, we present a novel way of reducing the memory footprint of explicit MPC solutions. The procedure is based on encoding all data (i.e., the critical regions and the feedback laws) as universal numbers (unums), which can be viewed as a memory-efficient extension of IEEE floating point standard. By doing so, we illustrate that the total memory footprint can be reduced by 80% without losing control accuracy. An additional advantage of proposed approach is, it can be applied on top of existing complexity reduction techniques.

Keywords: Explicit MPC, memory reduction, floating point numbers, universal numbers.

1. INTRODUCTION

Model Predictive Control (MPC) is an advanced control strategy where a Constrained Finite-Time Optimal Control (CFTOC) problem is solved on-line at each sampling interval to obtain the optimal open-loop control sequence. However, only the first input from this sequence is applied to the plant and based on that input, the plant's current state is measured and sent back to the optimizer to compute the next control sequence (Rawlings and Mayne, 2009, Chapter2). A major hurdle in the success of MPC for real-time applications running on embedded platforms is to solve MPC optimization problem within a sample instant. To mitigate the hurdle of on-line computational demand, a multi-parametric programming based approach i.e. *explicit MPC* has been proposed in Bemporad et al. (2002).

In explicit MPC, the optimal control law is pre-computed off-line as a function of all possible initial states. For a large class of MPC problems, such a control law can be shown to take the form of a Piecewise Affine (PWA) function defined over a polyhedral partition in the state space, that maps state measurements onto the optimal control value inputs. Having a pre-computed PWA function at the hand, explicit MPC needs to evaluate the PWA function on-line at each sample instant to compute optimal control actions based on the current state measurement (Bemporad et al., 2002). The major advantages of explicit MPC are: i) by using PWA function, on-line execution-time can be achieved in the range of milli- to microseconds (Oberdieck et al., 2016a), ii) MPC properties like closed-loop stability, feasibility and safety can be verified prior to deploying the control law on hardware, and iii) the PWA function evaluation can be performed on simple embedded hardware like a microcontroller, Programmable Logic Controller (PLC) or Field Programmable Gate Array (FPGA) (Honek et al., 2015).

However, to achieve such a simple and fast implementation, the data related to pre-computed PWA control law needs to be stored on the embedded hardware. Although, this aspect is less addressed in the literature, in fact it plays a crucial role when implementing explicit MPC on embedded devices with low-memory storage capacity and restricted computational power. In the last few years, an effort has been made to reduce the complexity of explicit MPC which is mainly focused on two directions: first, how to make feedback law simple; second, how to reduce the number of bits needed to store data with required accuracy.

The use of a model reduction technique has been proposed in Hovland and Gravdahl (2008) which reduces the

2405-8963 © 2017, IFAC (International Federation of Automatic Control) Hosting by Elsevier Ltd. All rights reserved.
Peer review under responsibility of International Federation of Automatic Control.
10.1016/j.ifacol.2017.08.1518

number of regions but with suboptimal control actions. It needs longer control horizons as compared to the horizons needed by original model to obtain good performance. By approximating the optimal MPC feedback, one can obtain less complex but suboptimal feedback function; see e.g. Bemporad and Filippi (2001), Jones and Morari (2009), Johansen and Grancharova (2003). All these techniques lead to a suboptimal control law. The performance-lossless complexity reduction techniques are represented by Optimal Region Merging (ORM) (Geyer et al., 2008), lattice representation (Wen et al., 2009), saturated region clipping (Kvasnica and Fikar, 2012), partial selection (Kvasnica et al., 2012), or region separation (Kvasnica et al., 2013). A good overview of complexity reduction techniques is demonstrated with the Multi-Parametric Toolbox (MPT) in Kvasnica et al. (2015). In all of the cases one can obtain less complex and performance-lossless explicit MPC solution, but the downside is that these techniques are limited to small systems due to the significant pre-computing effort required to solve non-trivial optimization problems.

A common drawback of all the aforementioned approaches is, the data of simplified MPC feedback law needs to be stored as Floating Point (FP) numbers in the IEEE format, 32-bits for single precision and 64-bits for double precision numbers. The bit size of numbers is thus constant regardless of the values they store. One of the way of reducing bit size of the underlying explicit MPC data was presented in Szücs et al. (2011) where, the authors have used Huffman encoding (Knuth, 1985) to compress some of the data (specifically, integer indices to a set of unique half-spaces). The procedure can thus be viewed as a variable-size encoding. Since there is a one-to-one correspondence between original data and their compressed counterparts, the compressed feedback law exhibits the same properties (e.g. control performance, closed-loop stability and constraint satisfaction) as the original one. An alternative was presented by Suardi et al. (2016) where the authors have proposed use of low precision arithmetic.

This paper aims at reducing the memory footprint of explicit MPC by using *universal numbers* (unums) (Gustafson, 2015) to represent the data associated with the optimal MPC feedback law. This allows one to use an explicit MPC for the number of systems that would otherwise be excluded due to the high complexity of resulting explicit controllers; that is, due to the lack of available memory or powerful computing devices to make point location algorithms faster. A key idea of unums is to store a real number with a variable bit length format using six subfields: sign bit, exponent, fraction, uncertainty bit, exponent size, and fraction size. Basically, unum is a superset of IEEE 754 floating point format (Muller et al., 2009) that tracks whether a number is an exact float or lies in the open interval between two exact floats. Compared to the standard floating point formats; the variable size in unum offers an ability to change its representative range, precision, and the uncertainty bit indicates the exactness of represented value. Thus, unums use fewer bits, obey algebraic laws, and do not require rounding, overflow, and underflow for proper operations (Gustafson, 2015, Chapter 3).

2. PRELIMINARIES AND PROBLEM STATEMENT

2.1 Explicit MPC

Consider the class of discrete time Linear Time-Invariant (LTI) systems

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

where $x \in \mathbb{R}^n$ is the state vector, $u \in \mathbb{R}^l$ is the control input, $A \in \mathbb{R}^{n\times n}$ and $B \in \mathbb{R}^{n\times l}$ and the pair (A, B) is stabilizable. State and input variables are subject to the polytopic constraints $x \in \mathcal{X} \subseteq \mathbb{R}^n$, $u \in \mathcal{U} \subseteq \mathbb{R}^l$ where $\mathcal{X}$ and $\mathcal{U}$ are polyhedral sets containing the origin in their respective interior. The constrained finite time optimal control problem for the LTI system in (1) is

$$\min_{U_N} x_N^T P x_N + \sum_{k=0}^{N-1} x_k^T Q x_k + u_k^T R u_k \tag{2a}$$

$$\begin{aligned}
\text{s.t.}\ & x_{k+1} = Ax_k + Bu_k, && k = 0, \ldots, N-1, && \text{(2b)}\\
& x_k \in \mathcal{X}, && k = 0, \ldots, N-1, && \text{(2c)}\\
& u_k \in \mathcal{U}, && k = 0, \ldots, N-1, && \text{(2d)}\\
& x_N \in \mathcal{X}_f, && && \text{(2e)}\\
& x_0 = x(t), && && \text{(2f)}
\end{aligned}$$

where $Q \in \mathbb{R}^{n\times n}$, $R \in \mathbb{R}^{l\times l}$ and $P \in \mathbb{R}^{n\times n}$ are the weighting matrices, with conditions $Q \succeq 0$ and $P \succeq 0$ to be positive semidefinite, and $R \succ 0$ to be positive definite. We denote by N the prediction horizon, x_{k+1} as the vector of predicted states at time instant k, $U_N = \{u_0, \ldots, u_{N-1}\}$ as the sequence of control actions, x_0 as the initial conditions, and $\mathcal{X}_f$ as the polyhedral constraint set for the terminal state x_N.

It is well known, see, e.g. (Borrelli et al., 2011, Chapter 12), that by using the substitution $x_k = A^k x_0 + \sum_{j=0}^{k-1} \left(A^{k-1-j}B\right) u_j$ the MPC problem (2) can be translated into a multi-parametric quadratic problem (mp-QP) of the form

$$\min_{U_N} U_N^T H U_N + x_0^T F U_N + x_0^T Y x_0 \tag{3a}$$

$$\text{s.t.}\ \ GU_N \leq w + Wx_0, \tag{3b}$$

where matrices $H \in \mathbb{R}^{lN\times lN}, F \in \mathbb{R}^{n\times lN}, Y \in \mathbb{R}^{n\times n}, G \in \mathbb{R}^{q\times lN}, w \in \mathbb{R}^q, W \in \mathbb{R}^{q\times n}$ and q is a number of inequalities.

Furthermore, as demonstrated e.g. by Bemporad et al. (2002), the mp-QP (3) admits a closed-form solution as a function $\kappa : \mathbb{R}^n \to \mathbb{R}^{lN}$ that maps the initial conditions x_0 onto the sequence of optimal control inputs $U_N^\star$, i.e., $U_N^\star = \kappa(x_0)$. Moreover, provided that $H \succ 0$ (which is satisfied once $R \succ 0$, $Q \succeq 0$, $P \succeq 0$), κ is a continuous piecewise affine (PWA) function over polyhedral critical regions:

$$\kappa(x) = \begin{cases} L_1 x + g_1 & \text{if } x \in \mathcal{R}_1 \\ & \vdots \\ L_M x + g_M & \text{if } x \in \mathcal{R}_M \end{cases} \tag{4}$$

where

$$\mathcal{R}_i = \{x \in \mathbb{R}^n \mid Z_i x \leq z_i\} \ \ i = 1, \ldots, M \tag{5}$$

are the polyhedral regions with $Z_i \in \mathbb{R}^{c_i\times n}$, $z_i \in \mathbb{R}^{c_i}$ describing the half-spaces of the i-th region with c_i being the number of half-spaces of the i-th region, and $L_i \in \mathbb{R}^{lN\times n}$,