ARTICLE IN PRESS

Available online at www.sciencedirect.com

ScienceDirect

Journal of the Franklin Institute ▮ (▮▮▮▮) ▮▮▮–▮▮▮

Journal of The Franklin Institute

www.elsevier.com/locate/jfranklin

# Low-complexity digital architecture for solving the point location problem in explicit Model Predictive Control

Alberto Oliveri, Christian Gianoglio, Edoardo Ragusa, Marco Storace*

*Department of Electrical, Electronic, Telecommunications Engineering and Naval Architecture (DITEN), University of Genoa, Via Opera Pia 11a, 16145 Genoa, Italy*

Received 5 September 2014; received in revised form 6 March 2015; accepted 7 March 2015

**Abstract**

This paper describes a digital circuit architecture which implements a recently proposed algorithm for the solution of the point location problem in the evaluation of piecewise affine functions. The circuit is suitable for FPGA implementation of explicit Model Predictive Control. The performances of the architecture are tested in a case study through hardware-in-the-loop simulation. Results show that the proposed circuit can be implemented on limited hardware resources also for quite complex, possibly discontinuous control functions, thus representing a good spare solution when other existing circuit architectures (generally faster but more resource-demanding) cannot be deployed.


## 1. Introduction

An increasing interest is devoted in the literature to the fast computation of Model Predictive Control (MPC), in order to use it in embedded control systems with low sampling times. The implicit (traditional) MPC strategy [1] is hardly applied in this context, since an optimization problem must be solved online at each sampling time (an application can be found in [2]). The explicit MPC solution [3] brings all the computation offline by solving the optimization problem for a given set of states; the result is an explicit formulation of the control function as a piecewise affine (PWA) function of the system states, defined over a domain partitioned into convex polytopes. The main limit of the explicit approach is that the number of polytopes the function is

*Corresponding author.
*E-mail addresses:* alberto.oliveri@unige.it (A. Oliveri), marco.storace@unige.it (M. Storace).

http://dx.doi.org/10.1016/j.jfranklin.2015.03.018
0016-0032/

defined on can become very large, if many constraints are imposed to the system and a large prediction horizon is used in the MPC setup. The advantage, instead, is that the online computation is limited to the evaluation of a (possibly discontinuous) PWA function, which is an easier task with respect to the minimization of a cost function. The computation in this case is essentially composed of two steps: (i) solve the point location problem, i.e., locate the polytope containing a given input point and (ii) evaluate the affine expression defined over this polytope.

The bottleneck of the whole computation is the point location problem, whose solution requires an efficient algorithm. The simplest way to solve the point location problem is to search for each polytope sequentially, until the correct one is found (*direct search*). This approach is obviously slow and inefficient, since in the worst case all polytopes must be explored. An alternative approach has been proposed in [4], which is based on a binary search tree. A digital circuit implementing the binary search tree algorithm is described in [5], where the tree is represented by a Finite State Machine (FSM). The main drawback of this approach is that, if the number of polytopes is too high, the FSM becomes very complex, thus preventing the implementation of the circuit in low-cost FPGAs, as may be requested by embedded applications.

More recently, a novel algorithm has been proposed, which employs direct search on a restricted subset of polytopes, determined by means of a fictitious hyper-rectangular domain partition [6]. A tuning parameter allows determining the coarseness of this fictitious partition in order to trade off between computation latency and memory requirements. This approach is therefore more efficient than the classical direct search method and does not require a FSM as for the binary search tree, thus being less resource-demanding. Nevertheless, the computation latency can be much higher with respect to using the binary search tree approach.

An alternative canonical form for continuous PWA functions, which is not prone to the point location problem, is the *lattice* form [7]. Recently, a circuit architecture suitable for FPGA implementation of this form has been proposed [8], which exhibits, in general, very good performances in terms of circuit size and memory occupation, and has a latency comparable with the circuit architecture implementing the binary search tree algorithm. The main drawback of the lattice form is that it cannot be used to represent discontinuous PWA functions.

In this paper we describe a digital circuit architecture implementing the algorithm proposed in [6]. The circuit performances are characterized by changing the tuning parameter and compared with those of other circuit architectures. The circuit is tested through hardware-in-the-loop simulation in the control of a PWA system with MPC technique.

## 2. Description of the algorithm

In this section we briefly recall the main elements, necessary to understand the proposed circuit architecture, of the algorithm described in [6] for the solution of the point location problem. Given (i) a domain $\mathcal{D} \subset \mathbb{R}^n$ partitioned into $P$ convex polytopes $\mathcal{P}_i$, $i = 1, \ldots, P$, such that $\bigcup_i \mathcal{P}_i = \mathcal{D}$ and $P_i \cap P_j = \emptyset$, if $i \neq j$, and (ii) a point $x \in \mathcal{D}$, the point location problem consists in finding the index $k$ such that $x \in \mathcal{P}_k$. Any polytope can be described by a set of inequalities, one for each edge of the polytope:

$$\mathcal{P}_i = \{x \in \mathcal{D} : H_i x \leq K_i\} \tag{1}$$

A PWA function $u : \mathbb{R}^n \to \mathbb{R}$ (e.g., obtained by solving an explicit MPC control problem) is defined as follows:

$$u = F_i x + G_i \quad \text{if } x \in \mathcal{P}_i \tag{2}$$

Please cite this article as: A. Oliveri, et al., Low-complexity digital architecture for solving the point location problem in explicit Model Predictive Control, Journal of the Franklin Institute. (2015), http://dx.doi.org/10.1016/j.jfranklin.2015.03.018