10

A General Lattice Representation for Explicit Model Predictive Control

William Sun [1] and Thomas Winn [2]
[1]*School of Business Administration, Duquesne University*
[2]*Emerson Process Management, Power & Water Solutions*
USA

1. Introduction

Model predictive control (MPC) is one of the most successful techniques to control multi-variable constraint systems. The MPC uses a mathematical model to predict the future effects of control inputs to system behaviors. The optimal control policy is obtained by solving an optimization problem that minimizes/maximizes a performance objective subject to inputs and outputs constraints over a future time horizon (Morari & Lee, 1999, Mayne et al., 2000, Dua et al, 2008). The MPC control laws are open-loop optimal with respect to the corresponding objective function. However, the conventional MPC may require intensive online computation due to the repetitive solutions of an optimization problem. This limits its applications to large and slowly varying systems. In addition, the MPC control strategy is hard to validate, especially for safety critical system (Grancharova & Johansen, 2005, Pistikopoulos, et al. 2001, Alessio & Bemporad 2009).

In 2002, Bemporad, Morari, Dua & Pistikopolous introduced the concept of explicit MPC (eMPC). The eMPC reformulates the online optimization in a MPC into a mutli-parametric linear/quadratic program (mpLP/mpQP). The optimal control action is calculated off-line as a continuous piecewise-affine (PWA) function of the state and reference vectors (Saffer et al, 2004, Bemporad et al. 2002a, 2002b). The eMPC has several advantages: i) The online computational time can be reduced to the microsecond-millisecond range. It makes the eMPC attractive for fast systems; ii) The MPC functionality is achieved in an easily verifiable way. The eMPC control policies can be validated before real online operations; iii) The eMPC solutions can be implemented with low-cost hardware (Johansen et al., 2007, Wen & Ma, 2008a, 2008b). The eMPC is then promising to be used in embedded systems, small/medium process systems, where the control systems should not be more expensive than the process systems. The eMPC found successful applications in many areas, e.g. AC-DC converters (Beccuti et al. 2009), autonomous vehicle steering, air separation unit (Grancharova et al. 2004), active valve train, hybrid separation, air condition (Pistikopoulos, et al. 2001), biomedical systems and drug delivery systems (Dua et al, 2004), scheduling (Ryu, & Pistikopoulos, 2007), spacecraft attitude control (Hegrenas et al, 2000)and crude distillation unit (Pannocchia et al, 2007).

The eMPC is essentially a strategy of trading time for space. A continuous PWA control map is calculated offline, and stored in memory for online usage (Pannocchia et al, 2007). The efficiency of eMPC method depends critically on finding an efficient representation model for eMPC controllers. However, it is not easy to develop a general model set to represent continuous PWA functions. The PWA representations usually have two conflicting criteria: description and evaluation complexities. The description complexity deals with the number of parameters in the representation model, while the evaluation complexity specifies the time for online calculation of function values.

When an eMPC controller is executed, one needs to solve a point-location problem with two steps: i) identify which polyhedral region the measured state lies in; ii) compute the control action using the corresponding affine control law. The simplest point-location solver is the sequential search (SS) algorithm, which is implemented by substituting a given state variable into the constraint inequalities of different regions. This method requires to store all the polyhedral regions and affine functions individually. Due to the combinatorial nature, the number of polyhedral regions in an eMPC control law can grow exponentially with the size of the optimal control problem (Wen et al., 2005a). Hence, the online function evaluation is computationally expensive when an eMPC control consists of a large number of regions or is defined over a complicated domain partition (Wen et al., 2005b).

Many researchers developed alternative ways to represent the eMPC solutions with appropriate data structures. In 2001, Borrelli, Baotić, Bemporad & Morari propose a search algorithm based on the convexity of the piecewise affine value function. This convex value function (CVF) algorithm reduces the storage space significantly. In some cases, this method might be time consuming because it requires a kind of sequential search. A binary search tree (BST) algorithm is proposed by Tondel, Johansen & Bemporad (2003) on the basis of the geometric structure of the polyhedral partition. This method can deal with the fully general PWA functions, including the discontinuous ones defined on overlapping regions or holes. In this scheme, the auxiliary hyper-planes are introduced, which may subdivide existing regions. This might lead to a worst case combinatorial number of subdivided regions. Then the following search procedure has to consider an additional increase in the number of regions, which may imply a prohibitive pre-processing time or online memory requirements. In 2007, Christophersen, Kvasnica, Jones & Morari developed an efficient search tree algorithm by utilizing the concept of bounding boxes and interval trees. This bounding box tree (BBT) algorithm can deal with the PWA functions defined over a large number of polyhedral regions. But the storage demand and online search time are still linear in the number of polyhedral regions in the original PWA functions. In Geyer et al, 2008, an efficient approach was developed to reduce the number of partitions in eMPC controllers by optimally merging the polyhedral regions where the affine gain is the same. All these algorithms are successful in dealing with the PWA functions defined over a large number of polyhedral regions. However, their storage demands and/or online evaluation computation are dependent on the number of polyhedral regions in the original PWA functions. These algorithms do not utilize the global structure information in eMPC controllers. Then their efficiency may be reduced substantially for some large-scale and complicated eMPC solutions.

The PWA approximation technique presents another efficient way to deal with the computation and description complexities of eMPC solutions. In Johansen, Petersen & Slupphaug (2002), an approach is proposed, which calculates the sub-optimal solutions by predetermining a small number of sampling data when the active set or input is allowed to change on the horizon. An alternative sub-optimal approach was developed in Bemporad & Filippi (2003) where small slacks are introduced on the optimality conditions and the mp-QP is used for the relaxed problem. In 2003, an algorithm is suggested that can determine a suboptimal explicit MPC control on a hypercubic partition (Johansen & Grancharova, 2003). In this partition, the domain is divided into a set of hypercubes separated by orthogonal hyperplanes. In 2006, Jones, Grieder & Rakovic interpret the PWA value function as weighted power diagrams (extended Voronoi diagrams). By using the standard Voronoi search methods, the online evaluation time is solved in logarithmic time (Jones, Grieder, & Rakovic, 2006; Spjotvold, Rakovic, Tondel, & Johansen, 2006). Dynamic programming can also be used to calculate the approximate explicit MPC laws (Bertsekas & Tsitsiklis, 1998; Lincoln & Rantzer, 2002, 2006). The main idea of these approaches is to find the sub-optimal solutions with known error bounds. The prescribed bounds can achieve a good trade-off between the computation complexity and accuracy. These approximation algorithms are very efficient regarding the storage and online calculation time. However, the approximate PWA functions usually have different domain partitions from the original explicit MPC laws. This deviation may hinder the controller performance and closed-loop stability.

The established representation and approximation algorithms have found many successful applications in a variety of fields. However, they can only evaluate the control actions for discrete measured states. None of them can provide the exact analytical expression of the PWA control laws. An analytical expression will ease the process of closed-loop performance analysis, online controller tuning and hardware implementations. The analytic expression also provides the flexibility of tailoring the PWA controllers to some specific applications, e.g. to develop different sub-optimal controllers in different zones (a union of polyhedral regions), and to smooth the PWA controllers at region boundaries or vertices (Wen et al. 2009a). In addition, the canonical PWA (CPWA) theory shows that the continuous PWA functions often consist of many redundant parameters. A global and compact analytical expression can significantly increase the computation and description complexity of eMPC solutions (Wen, et al., 2005a). An ideal representation algorithm should describe and evaluate the simplified MPC solutions after removing the redundant parameters.

In 1977 Chua & Kang proposed the first canonical representation for continuous PWA functions. A canonical PWA (CPWA) function is the sum of an affine function and one or more absolute values of affine functions. All continuous PWA functions of one variable can be expressed in the canonical form. However, if the number of variables is greater than one, only a subset of PWA functions have the CPWA representations (Chua & Deng, 1988). In 1994, Lin, Xu & Unbehauen proposed a generalized canonical representation obtained by nesting several CPWA functions. Such a representation is available for any continuous PWA function provided that the nesting level is sufficiently high. The investigations (Lin & Unbehauen, 1995; Li, et al. 2001, Julian et al., 1999) showed that for a continuous PWA function, the nesting level does not exceed the number of its variables. However, the nested absolute value functions often have implicit functional forms and are defined over complicated boundary configurations. In 2005, Wen, Wang & Li proposed a basis function CPWA (BPWA) representation theorem. It is shown that any continuous PWA function of n variables can be expressed by a BPWA function, which is formulated as the sum of a suitable number of the maximum/minimum of n+1 affine functions.

The class of lattice PWA functions is a different way to represent a continuous PWA function (Tarela & Martínez, 1999, Chikkula, et al., 1998, Ovchinnikov, 2002, Necoara et al. 2008, Boom & Schutter 2002, Wen et al, 2005c, Wen & Wang, 2005d). The lattice representation model describes a PWA function in term of its local affine functions and the order of the values of all the affine functions in each region. From theoretical point of view, the lattice PWA function has a universal representation capability for any continuous PWA function. According to the BPWA representation theorem, any BPWA function can be equivalently transformed into a lattice PWA function (Wen et al. 2005a, 2006). Then the well-developed methods to analyze and control the class of CPWA functions can be extended to that of the lattice PWA functions. From a practical point of view, it is of great significance that a lattice PWA function can be easily constructed, provided that we know the local affine functions and their polyhedral partition of the domain (Wen & Ma, 2007, Wen et al, 2009a, 2009b). Since these information on affine functions and partitions is provided in the solutions of both mp-LP and mp-QP, the lattice PWA function presents an ideal way to represent the eMPC solutions.

In this paper, we propose a general lattice representation for continuous eMPC solutions obtained by the multi-parametric program. The main advantage of a lattice expression is that it is a global and compact representation, which automatically removes the redundant parameters in an eMPC solution. The lattice representation can save a significant amount of online computation and storage when dealing with the eMPC solutions that have many polyhedral regions with equal affine control laws. Three benchmark MPC problems are illustrated to demonstrate that the proposed lattice eMPC control have a lower description complexity, comparable evaluation and preprocessing complexities, when compared to the traditional eMPC solutions without global description models.

The rest of this paper is organized as follows. Section II introduces the main features of PWA functions and eMPC problems. The lattice PWA function and representation theorem are presented in Section III. Section IV is the main part of this paper. It presents the complexity reduction theorem of lattice eMPC solutions, the lattice representation algorithm and its complexity analysis. Numerical simulation results are shown in Section V, and Section VI provides the concluding remarks.

2. PWA functions and eMPC solutions

2.1 PWA function

Definition 1. *In $\Re^n$, let $\Omega = \cup_{i=1}^{M} R_i$ be a compact set, which is partitioned into M convex polyhedrons called regions $R_i, i = 1, \ldots, M$. Then a nonlinear function $p(x) : \Omega \mapsto \Re^m$ is defined as a PWA function if*

$$p(x) = F_i x + g_i, \quad \forall x \in R_i \tag{1}$$

with $F_i \in \Re^{m\times(n+1)}, \beta_i \in \Re^m$. A PWA function is continuous if

$$F_i x + g_i = F_j x + g_j, \quad \forall x \in B_{i,j} \tag{2}$$

where $B_{i,j} = R_i \cap R_j$ is defined as boundaries and $i, j \in [1, \cdots, M]$. Specially, when $m = 1$, $p(x)$ is called as a scalar PWA function, i.e.

$$p(x) = \ell(x|\alpha_k, \beta_k) = \alpha_k^T x + \beta_k, \quad \forall x \in R_i \tag{3}$$

with $\alpha_k \in \Re^n, \beta_k \in \Re$ and $1 \le k \le M$. For convenience of statement, we simply denote $\ell(x|\alpha_k, \beta_k)$ as $\ell_k(x)$.

In Definition 1, each region R_i is a polyhedron defined by a set of inequality

$$R_i = \{x \in \Re^n | H_i x \leq K_i\} \tag{4}$$

where H_i, K_i are matrices of proper sizes with $i = 1, \cdots, M$. Geometrically, a boundary $B_{i,j}$ is a real set of an $(n-1)$-dimensional hyperplane.

2.2 Explicit MPC

Consider the linear time invariant system

$$\begin{cases} x(t+1) = Ax(t) + Bu(t) \\ \quad\; y(t) = Cx(t) + Du(t) \end{cases} \tag{5}$$

which fulfills the following constraints

$$x_{min} \leq x(t) \leq x_{max},\; y_{min} \leq y(t) \leq y_{max},\; u_{min} \leq u(t) \leq u_{max}, \delta u_{min} \leq \delta u(t) \leq \delta u_{max}, \tag{6}$$

at all time instants $t \geq 0$. In (5)-(6), $x(t) \in \Re^n$ is state variable, $u(t) \in \Re^m$, $y(t) \in \Re^{n_y}$ are control input and system output, respectively. A, B, C and D are matrices of appropriate dimensions, i.e. $A \in \Re^{n \times n}, B \in \Re^{n \times m}$, $C \in \Re^{n_y \times n}$ and $D \in \Re^{n_y \times m}$. It is assumed that (A, B) is a controllable pair. δu_{min} and δu_{max} are rate constraints. They restrict the variation of two consecutive control inputs ($\delta u(t) = u(t) - u(t-1)$) to be within of prescribed bounds. The system is called as a single-input system when $m = 1$, and a multi-input system when $m \geq 2$. Assume that a full measurement of the state $x(t)$ is available at current time t. The MPC solves the following standard semi-infinite horizon optimal control problem:

$$J^*(x(t)) = \min_{U=\{u(t)^T, \cdots, u(t+N-1)^T\}^T} V_N(x(t+N)) + \sum_{k=0}^{N-1} V_k\big(x(t+k|t), u(t+k)\big) \tag{7}$$

subject to

$$\begin{cases} x_{min} \leq x(t+k|t) \leq x_{max}, \quad k = 1, \cdots, N_y, \\ y_{min} \leq y(t+k|t) \leq y_{max}, \quad k = 1, \cdots, N_y, \\ u_{min} \leq u(t+k|t) \leq u_{max}, \quad k = 1, \cdots, N_c, \\ \delta u_{min} \leq \delta u(t+k|t) \leq \delta u_{max}, k = 1, \cdots, N_c, \\ x(t) = x(t|t) \\ x(t+k+1|t) = Ax(t+k|t) + Bu(t+k), \quad k \geq 0 \\ y(t+k+1|t) = Cx(t+k|t) + Du(t+k), \quad k \geq 0 \\ u(t+k) = Kx(t+k|t), \quad N_u \leq k < N_y \end{cases} \tag{8}$$

at each time t, where $x(t+k|t)$ denotes the the predicted state vector at time $t+k$. It is obtained by applying the input sequence $u(t), \cdots, u(t+k-1)$ to system (5). In (7), K is the feedback gain, N_u, N_y, N_c are the input, output and constraint horizons, respectively. Normally, we have $N_u \leq N_y$ and $N_c \leq N_y - 1$.

The stage cost function is defined as

$$V_i\big(x(t+i|t),u(t+i)\big) = ||Qx(t+k|t)||_p + ||Ru(t+k)||_p \tag{9}$$

$$V_N\big(x(t+N_y|t)\big) = ||Px(t+N_y|t)||_p \tag{10}$$

where $||\cdot||$ denoted a kind of norm and $p \in \{1,2,+\infty\}$, P,Q and R are weighting matrices of proper sizes. V_N is the terminal penalty function. In this paper, it is assumed that the parameters P,Q,R are chosen in such a way that problem (7) generates a feasible and stabilizing control law when applied in a receding horizon fashion and $J^*(x)$ is a polyhedral piecewise affine/quatratic Lyapunov function.

At each time t, the MPC control law $u(t)$ is the first item in the optimal solution $u^*(t)$, i.e.

$$u(t) = u^*(t) \tag{11}$$

where $u^*(t) = \{u^*(t),\cdots,u^*(t+N_c-1)\}$. Apply $u(t)$ as input to problem (5) and repeat the optimization (7) at time $t+1$ using the new state $x(t+1)$. This control strategy is also referred to as moving or receding horizon.

By some algebraic manipulations, the MPC problem can be formulated as a parametric Linear Program (pLP) for $p \in \{1,+\infty\}$

$$\begin{aligned} u^*(x) &= \min_u \Upsilon^T u \\ s.t.\ \ Gu &\leq W + Ex \end{aligned} \tag{12}$$

or a parametric quadratic Program (pQP) for $p = 2$

$$\begin{aligned} u^*(x) &= \min_u \Upsilon^T u + \frac{1}{2}u^T H u \\ s.t.\ \ Gu &\leq W + Ex \end{aligned} \tag{13}$$

See (Bemporad et al. 2002) for details on the computation of the matrices G, W, E, H and Υ in (12) and (13). By solving the pLP/pQP, the optimal control input $u^*(x)$ is computed for each feasible value of the state x. The features of MPC controllers and value functions are summarized in the following lemma.

Lemma 1. Kvasnica et al., 2004 *Consider the multi-parametric programming of (12) and (13). The solution $u^*(x) : \Re^n \mapsto \Re^m$ is a continuous and piecewise affine*

$$u^*(x) = F_i x + g_i,\ \ \forall x \in R_i \tag{14}$$

where $R_i, i = 1,\cdots,M$ is the polyhedral regions. The optimal cost $J^*(x(t))$ is continuous, convex, and piecewise quadratic ($p = 2$) or piecewise affine ($p \in \{1,\infty\}$).

3. Lattice representation of scalar eMPC solutions

3.1 Lattice PWA Function

Let $\Phi = [\phi_1,\cdots,\phi_M]^T$ be an $M \times (n+1)$ matrix and $\Psi = [\psi_{ij}]$ a $M \times M$ zero-one matrix. A lattice piecewise-affine function $P(x|\Phi,\Psi)$ may be formed as follows,

$$P(x|\Phi,\Psi) = \min_{1\le i\le M}\left\{\max_{\substack{1\le j\le M\\ \psi_{ij}=1}}\{\ell_j(x)\}\right\},\ \forall x \in \Re^n. \tag{15}$$

Note that $P(x|\Phi,\Psi)$ is equal to one of $\ell_1(x),\cdots,\ell_M(x)$ for any $x \in \Re^n$. $P(x|\Phi,\Psi)$ is indeed a continuous PWA function whose local affine functions are just $\ell_j(x), 1 \le j \le M$. The parameter vectors of these affine functions are exactly the row vectors of Φ. Hence the matrix Φ is called a parameter matrix. The matrix Ψ is defined as a structure matrix, if its elements are calculated as

$$\psi_{ij} = \begin{cases} 1 & if \quad \ell_i(x) \ge \ell_j(x) \\ 0 & else \end{cases} \tag{16}$$

with $x \in R_i$ and $1 \le i,j \le M$. Similarly, a dual structure matrix $\hat{\Psi} = [\hat{\psi}_{ij}]^{M\times M}$ is defined by

$$\hat{\psi}_{ij} = \begin{cases} 1 & if \quad \ell(x|\phi_i) \le \ell(x|\phi_j) \\ 0 & else \end{cases} \tag{17}$$

with $x \in R_i$ and $1 \le i,j \le M$.

Lemma 2. $^{\textit{Wen et al., 2007}}$ *Given any n-dimensional continuous PWA function $p(x)$, there must exist a lattice PWA function $P(x|\Phi,\Psi)$ such that*

$$p(x) = P(x|\Phi,\Psi), \forall x \in \Re^n \tag{18}$$

where Φ,Ψ are parameter and structure matrices, respectively.

It is shown in Lemma 2 that a continuous PWA function can be fully specified by a parameter matrix Φ and a structure matrix Ψ. This provides a systematic way to represent the eMPC solutions. The lattice PWA function contains only the operators of min, max and vector multiplication. It is an ideal model structure from the online calculation point of view.
Example 1: The realization of a lattice PWA function can be made more clear using a simple example. Let $p(x)$ be a 1-dimensional PWA function with 4 affine segments,

$$p(x) = \begin{cases} \ell_1(x) = 1, & \forall x \in R_1 = [-2,-1] \\ \ell_2(x) = -x, & \forall x \in R_1 = (-1,0] \\ \ell_3(x) = x, & \forall x \in R_2 = (0,1] \\ \ell_4(x) = -x+2, & \forall x \in R_3 = (1,2] \end{cases} \tag{19}$$

where the plot of $p(x)$ is depicted in Fig. 1. It is easy to see that

$$\begin{cases} \ell_4(x) > \ell_2(x) > \ell_1(x) > \ell_3(x), & \forall x \in R_1 \\ \ell_4(x) > \ell_1(x) > \ell_2(x) > \ell_3(x), & \forall x \in R_2 \\ \ell_4(x) > \ell_1(x) > \ell_3(x) > \ell_2(x), & \forall x \in R_3 \\ \ell_3(x) > \ell_1(x) > \ell_4(x) > \ell_2(x), & \forall x \in R_4 \end{cases} \tag{20}$$

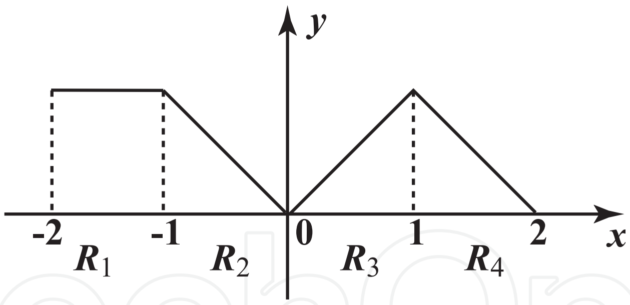


Fig. 1. Plot of 1-dimensional PWA function $p(x)$

Then the structure matrix is written as $\Psi = \begin{bmatrix} 1\,0\,1\,0 \\ 0\,1\,1\,0 \\ 0\,1\,1\,0 \\ 0\,1\,0\,1 \end{bmatrix}$. It follows from (5) that the parameter matrix is $\Phi = \begin{bmatrix} 0 & 1 \\ -1 & 0 \\ 1 & 0 \\ -1 & 2 \end{bmatrix}$. Finally, the lattice PWA function is formulated as

$$p(x) = P(x|\Phi, \Psi) = \min\{\max\{\ell_1, \ell_3\}, \max\{\ell_2, \ell_3\}, \max\{\ell_2, \ell_3\}, \max\{\ell_2, \ell_4\}\} \tag{21}$$

It is obvious that the lattice PWA function in (21) can be further simplified. The simplification algorithm will be discussed in the subsequent sections.

3.2 Lattice representation theorem of eMPC solutions

Lemma 3. *Assume that R_i, R_j are two n-dimensional convex polytopes, where $\ell_i(x), \ell_j(x)$ are their local affine functions with $i, j \in \{1, \cdots, M\}$. Then the structure matrix $\Psi = [\psi_{ij}]^{M\times M}$ can be calculated as follows:*

$$\psi_{ij} = \begin{cases} 1 & if \quad \ell_i(v_k) \geq \ell_j(v_k), 1 \leq k \leq K_i \\ 0 & if \quad \ell_i(v_k) < \ell_j(v_k), k \in \{1, \cdots, K_i\} \end{cases} \tag{22}$$

where v_k are the vertices of R_i with $1 \leq k \leq K_i$ and $K_i \in \mathbb{Z}^+$ is the number of vertices of R_i.

Proof. Since R_i is an n-dimensional polytope, it can be described by its vertices $v_1, \cdots, v_{K_i}$

$$R_i = \{x \in \Re^n | x = \sum_{k=1}^{K_i} \lambda_k v_k, 0 \leq \lambda_k \leq 1, \sum_{i=1}^{K_i} \lambda_k = 1\} \tag{23}$$

Then for any $x \in R_i$, we have

$$\ell_i(x) = \ell_i\left(\sum_{k=1}^{K_i} \lambda_k v_k\right) = \sum_{k=1}^{K_i} \lambda_k \ell_i(v_k) \tag{24}$$

$$\ell_j(x) = \ell_j\left(\sum_{k=1}^{K_i} \lambda_k v_k\right) = \sum_{k=1}^{K_i} \lambda_k \ell_j(v_k) \tag{25}$$

If $\ell_i(v_k) \geq \ell_j(v_k), \forall 1 \leq k \leq K_i$, then $\ell_i(x) \geq \ell_j(x)$ holds for all $x \in R_i$. It follows from (3) that $\psi_{ij} = 1$.
Similarly, if there exists any $k \in \{1, \cdots, K_i\}$ such that $\ell_i(v_k) < \ell_j(v_k)$, then $\ell_i(x)$ and $\ell_j(x)$ will intersect together with an $(n-1)$-dimensional hyperplane as the common boundary. This implies that $\psi_{ij} = 0$.
Using the same procedure stated above, all the elements in the structure matrix Ψ can be calculated, and this completes the proof of Lemma 3. □

Lemma 3 shows that the order of the affine function values in a convex polytope can be specified by the order of the function values at the polytope vertices. This presents a constructive way to realize the structure matrix of a given PWA function.

Theorem 1. *Any continuous eMPC solution can be represented by a lattice PWA function.*

Proof. According to Bemporad et al. 2002, an eMPC solution is presented in the form of conventional PWA representation, which lists all the parameters of the affine functions and regions in a table. Each region is a convex polytope defined by a set of inequalities. It follows from Lemma 2 that an explicit solution to MPC can be realized by a structure matrix and a parameter matrix. These two matrices specify a lattice PWA function. Then any eMPC solution can be described by a lattice PWA function. This completes the proof of Theorem 1. □

4. Simplification of scalar lattice PWA representation

4.1 Super-region

Definition 2. *Given a PWA function $p(x) : \Omega \mapsto \Re^m$ with M regions, i.e. $\Omega = \cup_{i=1}^{M} R_i$. Let*

$$\Gamma_i = \left\{ j \in \{1, \cdots, M\} | \alpha_i x + \beta_i = \alpha_j x + \beta_j, \forall x \in \Omega \right\} \tag{26}$$

be a finite set with $\tilde{M}$ components. Then the set $\Pi \subseteq \Omega$ is defined as a super-region, if $\Pi = \cup_{k=1}^{\tilde{M}} R_k$ and $k \in \Gamma_i$ with $i \in \{1, \cdots, M\}$.

A super-regions is defined as a union of polyhedral regions with same affine function. It can be non-convex or even not connected. If a PWA function have many regions with the same local functions, the number of super-regions is much less than that of regions.
The concept of super-region can be clarified by an 1-dimensional PWA function shown in Fig. 2. The PWA function $p(x)$ is defined over a compact set $\Omega = AE$. The domain is partitioned into 4 regions, i.e. $\Omega = \cup_{i=1}^{4} R_i$. Each region R_i is a convex polyhedron defined by two inequalities, e.g. $R_2 = BC = \{x \in AF | x \geq x_B, x \leq x_C\}$, where x_B, x_C are the coordinates of points B, C. In Ω, there are 3 boundaries, e.g. B, C and D. Note that $p(x) = [p_1(x), p_2(x)]^T$, where

$$p_j(x) = \alpha_{i,j}^T x + \beta_{i,j} \quad \forall x \in R_i$$

with $\alpha_{i,j}, \beta_{i,j} \in \Re$, $i = 1, \cdots, 4$, $j = 1, 2$. We can get $F_i = \begin{bmatrix} \alpha_{i,1}^T \\ \alpha_{i,2}^T \end{bmatrix}$ and $g_i = \begin{bmatrix} \beta_{i,1} \\ \beta_{i,2} \end{bmatrix}$.
It follows from the plot of $p_1(x)$ that $\alpha_{1,1}^T x + \beta_{1,1} = \alpha_{4,1}^T x + \beta_{4,1}, \forall x \in AE$. Then $\Pi_1 = R_1 \cup R_2 = AB \cup DE$ is defined as a super-region. It is evident that Π_1 is not convex, because it

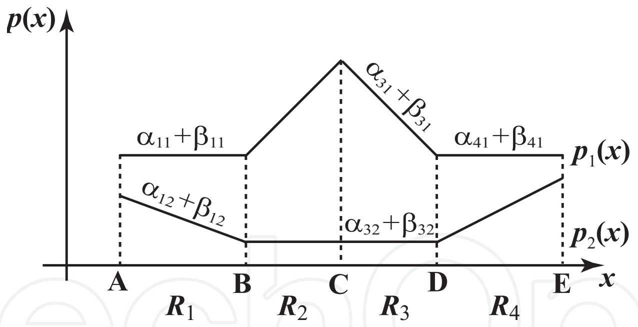


Fig. 2. Plot of a 1-dimensional vector PWA function $p(x) = [p_1(x), p_2(x)]^T$.

is composed of two disconnected line intersections. Similarly, $\Pi_2 = R_2 \cup R_3 = BD$ defines another super-region of $p_2(x)$.

4.2 Row vector simplification lemma

Lemma 4. *Assume that $P(x|\Phi, \Psi) : D \subset \Re^n \mapsto \Re$ is a PWA function with M linear segments. Let φ_i, φ_j be rows of the structure matrix. If the pointwise inequation $\varphi_i - \varphi_j \leq 0$ holds for any $i, j \in \{1, \cdots, M\}$, there exist a simplified structure matrix $\tilde{\Psi} \in \Re^{(M-1)\times M}$, such that*

$$P(x|\Phi, \Psi) = P(x|\Phi, \tilde{\Psi}) \tag{27}$$

where $\Psi \in \Re^{M\times M} = [\varphi_1, \cdots, \varphi_M]^T$ and $\tilde{\Psi} = [\varphi_1, \cdots, \varphi_{j-1}, \varphi_{j+1}, \cdots, \varphi_M]^T$.

Proof. Denote $I_i \in \Re^M$ as the index set of the local affine functions, whose values are smaller than the i-th affine function in its active region, i.e.

$$I_i = \{k | \ell_k(x) \leq \ell_i(x), \forall x \in R_i\} \tag{28}$$

with $i, k \in \{1, \cdots, M\}$. Since $\varphi_i - \varphi_j \leq 0$ holds for any pointwise inequality, we can get $I_i \subseteq I_j$. It directly follows that $\{\ell_p(x)\} \subseteq \{\ell_q(x)\}$ with $p \in I_i, q \in I_j$.
Therefore, it leads that

$$\max_{p\in I_i}\{\ell_p(x)\} \leq \max_{q\in I_j}\{\ell_q(x)\} \tag{29}$$

This implies that

$$\min\big\{\max_{p\in I_i}\{\ell_p(x)\}, \max_{q\in I_j}\{\ell_q(x)\}\big\} = \max_{p\in I_i}\{\ell_p(x)\} \tag{30}$$

Then we finally have

$$P(x|\Phi, \Psi) = \min_{1\leq i\leq M}\big\{\max_{k\in I_i}\{\ell_k(x)\}\big\} = \min_{\substack{1\leq i\leq M\\ i\neq j}}\big\{\max_{k\in I_i}\{\ell_k(x)\}\big\} = P(x|\Phi, \tilde{\Psi}) \tag{31}$$

Here we can see that the j-th row of structure matrix Ψ can be deleted without affecting the function values of $P(x|\Phi, \Psi)$. This completes the proof of Lemma 4. □

Since Lemma 4 can be used recursively, a much simplified structure matrix is obtained by deleting all the redundant rows. A single row in $\tilde{\Psi}$ corresponds to a super region, which is defined as an aggregation of several affine regions. Being a mergence of many convex polytopes, a super region can be concave or even disconnected. Then the number of super regions can be much smaller than that of regions (Wen, 2006).

4.3 Column vector simplification lemma

Lemma 5. *Assume that $P(x|\Phi, \Psi) : D \subset \Re^n \mapsto \Re$ is a PWA function with M linear segments. Denote $\Psi = [\psi_{ij}]^{M\times M}$ and $\hat{\Psi} = [\hat{\psi}_{ij}]^{M\times M}$ as the primary and dual structure matrix. Then the following results hold.*

1. *Given any $i, j, k \in \{1, \cdots, M\}$, if $k, j \in I_i$ and $\hat{\psi}_{jk} = 1$, then $\psi_{ij} = 0$, where I_i is the same as defined in (13);*
2. *If $\psi_{ij} = 0, \forall 1 \leq j \leq M$, then there exist a simplified structure matrix $\tilde{\Psi} \in \Re^{(M-1)\times M}$ and parameter matrix $\tilde{\Phi} \in \Re^{(M-1)\times(n+1)}$, such that*

$$P(x|\Phi, \Psi) = P(x|\tilde{\Phi}, \tilde{\Psi}) \tag{32}$$

where $\Phi \in \Re^{M\times(n+1)} = [\phi_1, \cdots, \phi_M]^T$, $\tilde{\Phi} \in \Re^{(M-1)\times(n+1)} = [\phi_1, \cdots, \phi_{j-1}, \phi_{j+1}, \cdots, \phi_M]^T$, and $\Psi, \tilde{\Psi}$ are the same as defined in Lemma 4.

Proof. According to (17), if $\hat{\psi}_{jk} = 1$, we have

$$\ell_j(x) \leq \ell_k(x), \forall x \in R_j \tag{33}$$

which implies that $\ell_j(x)$ is inactive in its own region, i.e.

$$\max\{\ell_j(x), \ell_k(x)\} = \ell_k(x), \forall x \in R_j \tag{34}$$

Note that $k, j \in I_i$ and I_i is the index set of $\ell_i(x)$. We can get

$$\max_{p\in I_i}\{\ell_p(x)\} = \max_{\substack{p\in I_i\\ p\neq j}}\{\ell_p(x)\}, \forall x \in D \tag{35}$$

This implies that $\psi_{ij} = 0$.
In addition, if $\psi_{ij} = 0$ holds for any $1 \leq j \leq M$, then $\ell_j(x)$ will be totally covered by other affine functions throughout the whole domain. Therefore, the j-th column of the structure matrix and j-th row of the parameter matrix can be deleted. This means that $P(x|\Phi, \Psi) = P(x|\tilde{\Phi}, \tilde{\Psi})$. It should be noted that the matrix $\tilde{\Psi}$ corresponds to a simpler lattice PWA function than Ψ even without a deletion of row vectors. A lattice PWA function with less terms in max operators is produced if some elements in the structure matrix are changed from one to zero. This completes the proof of Lemma 5. □

The significance of Lemma 5 is that it can differentiate the inactive regions from the active ones in a given PWA function. The inactive regions can then be removed from the analytic expression because they do not contribute to the PWA function values. The active regions are also referred to as the lattice regions, which define the number of columns in the structure matrix $\tilde{\Psi}$.
Lemma 5 presents an efficient and constructive method to reduce the complexity of a lattice PWA function. Recalling that an eMPC controller $u(x) \in \Re$ of a single input system is a continuous scalar PWA function, in which many polyhedral regions have same feedback

gains. It implies that the number of super-regions is usually much smaller than that of polyhedral regions. The complexity reduction algorithm of Lemma 5 can produce a very compact representation of the scalar eMPC solutions.

Example 2: In order to clarify the simplification procedure, we consider the lattice PWA function of (21) derived in Example 1.
Denoting $\Psi = [\varphi_1 \;\; \varphi_2 \;\; \varphi_3 \;\; \varphi_4]^T$, we can get $\varphi_2 - \varphi_3 = [0 \;\; 0 \;\; 0 \;\; 0] \leq 0$, where $'' \leq ''$ is the pointwise inequality. It follows from Lemma 4 that the third row vector φ_3 can be removed. Then the structure matrix is simplified as

$$\Psi = \begin{bmatrix} 1\,0\,1\,0 \\ 0\,1\,1\,0 \\ 0\,1\,0\,1 \end{bmatrix}. \tag{36}$$

Furthermore, by using (17), we can obtain the dual structure matrix

$$\hat{\Psi} = \begin{bmatrix} 1\,1\,0\,1 \\ 1\,1\,0\,1 \\ 1\,0\,1\,1 \\ 1\,0\,1\,1 \end{bmatrix}. \tag{37}$$

According to (36), we have $I_i = \{k, j\}$ with $k = 4, j = 2$ and $i = 3$. Using (37), we further have $\hat{\psi}_{jk} = \hat{\psi}_{24} = 1$. Then it follows from Lemma 5 that the item of ψ_{32} can be put to zero. The final structure matrix is written as

$$\tilde{\Psi} = \begin{bmatrix} 1\,0\,1\,0 \\ 0\,1\,1\,0 \\ 0\,0\,0\,1 \end{bmatrix} \tag{38}$$

The corresponding lattice PWA function is

$$p(x) = \min\left\{ \max\{\ell_1, \ell_3\}, \max\{\ell_2, \ell_3\}, \ell_4 \right\} \tag{39}$$

4.4 Lattice PWA representation theorem

Theorem 2. *Let $P(x) : \Omega \mapsto \Re$ be a continuous scalar PWA function with $\hat{M}$ super-regions. There must exist a positive integer $\bar{M} \leq \hat{M}$, a parameter matrix $\Phi \in \Re^{\hat{M}\times(n+1)}$, a structure matrix $\Psi = [\psi_{ij}]^{\bar{M}\times\hat{M}}$ and a lattice PWA function*

$$L(x|\Phi, \Psi) = \min_{1\leq i\leq \bar{M}} \left\{ \max_{\substack{1\leq j\leq \hat{M} \\ \psi_{ij}=1}} \{\ell_j(x)\} \right\} \tag{40}$$

such that

$$P(x) = L(x|\Phi, \Psi), \forall x \in \Omega \tag{41}$$

where ψ_{ij} is a boolean variable, $\Phi = [\phi_1, \cdots, \phi_{\hat{M}}]^T$, $\phi_j = [\alpha_j^T, \beta_j]$, $\alpha_j \in \Re^n, \beta_j \in \Re$ with $1 \leq i \leq \bar{M}, 1 \leq j \leq \hat{M}$.

Theorem 2 shows that the class of lattice PWA functions provides a universal model set for continuous scalar PWA functions. The complexity of a lattice PWA function is specified by the number of super-regions instead of that of regions. Then the lattice PWA functions may present a more compact representation than the PWA models without global analytical descriptions.

The scalar lattice representation theorem can be generalized to describe a vector eMPC solution $u(x) : \Omega \mapsto \Re^m$. The main idea is to represent each component scalar eMPC feedback law individually.

Theorem 3. *Let $u(x) = [u_1(x), \cdots, u_m(x)]^T$ be a continuous vector eMPC solution with $x \in \Omega$. There must exist m lattice PWA functions $L(x|\Phi_i, \Psi_i)$ such that*

$$u_i(x) = L(x|\Phi_i, \Psi_i) \quad \forall x \in \Omega \tag{42}$$

where Φ_i, Ψ_i are parameter and structure matrices and $i = 1, \cdots, m$.

The vector lattice representation theorem is valid for continuous PWA functions. It is proved in (Spjotvold et al, 2007, Bemporad et al. 2002) that an eMPC controller is continuous from a strictly convex mpQP problem. The continuity property is further generalized to general convex mpQP problems (Spjotvold et al, 2006). An eMPC problem with a linear cost function may have discontinuous solutions because of the degeneracy of critical regions. It is proved in (Bemporad et al, 2002) that there always exists a polyhedral partition even for degenerate critical regions, such that the eMPC control is continuous. Recalling that the mpLP problems are essentially special realizations of convex mpQP problems. It is proved constructively in (Spjotvold et al, 2006) that a continuous eMPC solution can be found for LP-based MPC problems by using a minimum norm method. Therefore, the set of continuous PWA functions can cover a wide class of eMPC solutions by utilizing appropriate multi-parametric program solvers.

The continuity of eMPC controllers can be easily verified by checking the function values at the vertices of different regions. This function has been implicitly implemented in the lattice PWA representation algorithm (Wen et al., 2009a). Therefore, the lattice representation can automatically separate the continuous eMPC solutions from the discontinuous ones. In addition, the discontinuity in eMPC controls are often caused by the overlapping of critical regions (Bemporad et al, 2002). The mpt-toolbox (Kvasnica et al., 2004) has a function to detect the existence of overlapping regions. It presents another efficient way to verify the continuity of eMPC controls.

The vector lattice representation can be extended to discontinuous eMPC solutions. A discontinuous eMPC solution is usually decomposed into a set of continuous PWA functions. Recalling that each continuous PWA function has a vector lattice representation. Then the discontinuous eMPC solutions can be represented by a set of lattice PWA functions and a switch logic. The switch logic may be implemented as a binary search tree (Tondel et al. 2003) or bounding box search tree (Christophersen et al, 2007). Further research is under investigation to generalize the lattice representation method to discontinuous PWA functions.

The lattice representation has a quadratic complexity for both online evaluation and memory storage. When the eMPC solutions consist of a large number of super-regions, e.g. the eMPC problems has a large input constraint set, the BST or BBT algorithms may have a lower online computational complexity.

4.5 Representation algorithm

In Kvasnica et al. 2004, a Multi-Parametric Toolbox (MPT) for computing optimal feedback controllers of constrained linear and piecewise affine systems is developed. The toolbox offers a broad spectrum of algorithms to calculate the eMPC solutions. The proposed lattice PWA representation algorithm can be easily embedded into the MPT toolbox and provide a better performance in term of online calculation and memory space requirements.
The main steps of the representation algorithm are summarized as follows.

1. Calculate the eMPC solution using the MPT toolbox. Record the local affine functions, constrained inequalities and vertices of each region;
2. Calculate the values of each affine function at each vertex;
3. Calculate the structure matrix using Lemma 3;
4. Delete the redundant row vectors in structure matrix using Lemma 4;
5. Delete the redundant elements in structure and parameter matrices using Lemma 5;
6. Get the lattice PWA expression of an eMPC solution.

It should be noted that the multi-parametric solver may return a PWA solution that is discontinuous, even for problems where continuous PWA solution exists. Then the lattice representation algorithm is feasible for the continuous PWA solutions obtained from the multi-parametric solver.

4.6 Complexity analysis

Let $u(x) = [u_1(x), \cdots, u_m(x)]^T$ be a vector PWA function with M polydedral regions and $x \in \Re^n$. Denote $\tilde{M}_k$ as the number of lattice regions in $u_k(x)$ and $\hat{M}_k$ the number of super regions with $1 \leq k \leq m$.

4.6.1 Storage

The lattice representation requires the storage of a $(n+1) \times \sum_{k=1}^{m} \tilde{M}_k$ parameter matrix and a $\sum_{k=1}^{m} \hat{M}_k \times \tilde{M}_k$ structure matrix. The total memory needed is $O\big((n+1)\sum_{k=1}^{m} \tilde{M}_k\big)$ real numbers and $O\big(\sum_{k=1}^{m} \hat{M}_k \tilde{M}_k\big)$ binary numbers. The structure matrix is usually very sparse. The actual required memory space can be significantly smaller than the worst estimation through the use of appropriate sparse storage techniques.

4.6.2 Online complexity

For a given state variable, the online evaluation of a lattice PWA control law consists of 3 steps. The first step is to calculate the function value of $\tilde{M}_k$ affine functions. This requires $n\tilde{M}_k$ multiplication and $n\tilde{M}_k$ sums. In the second step, we need to calculate the maximum of $\bar{M}_k$ function values, where $\bar{M}_k$ is the number of affine functions with $\psi_{ij} = 1$ in the structure matrix. Note that $\bar{M}_k \leq \tilde{M}_k$. In the worse case, this step requires $(\tilde{M}_k - 1) \times \tilde{M}_k$ by considering the $\tilde{M}_k$ maximization terms. The last step is to calculate the minimum of $\hat{M}_k$ real numbers. It needs $\hat{M}_k - 1$ comparisons. Therefore, the total online complexity is $O\big(\sum_{k=1}^{m}(2n\tilde{M} + (\tilde{M}_k - 1)\tilde{M}_k + (\hat{M}_k - 1)\big)$. It follows from Wen et al. (2009a) that $n \ll \tilde{M}_k$ and $\hat{M}_k < \tilde{M}_k$ holds for any $k \in \{1, \cdots, m\}$. The online complexity can be roughly approximated by $O\big(\sum_{k=1}^{m} \tilde{M}_k^2\big)$. It should be noted that the structure matrix is usually sparse. The estimate of online calculation is very conservative in most cases. Then the average online calculation complexity can be considerably lower than the worse case estimate.

4.6.3 Preprocessing

The preprocessing phase for the lattice representation is composed of two steps. The first step is to calculate the eMPC control law using multi-parameter program. The second one is to represent the eMPC law with a lattice PWA function. It is very difficult to present a close-form solution of the off-line complexity. However, it was observed in extensive trials that the representation step takes significantly less time than the initial computation of an eMPC controller. Therefore, the proposed lattice representation method can be applied to any system for which an explicit controller is feasible.

5. Numerical examples

Two examples are illustrated in this section. All the simulations are run in Matlab 2007a on a 2.0 GHz Core 2 CPU with 1 GB RAM.
Example 3: Consider the double integrator

$$y(t) = \frac{1}{s^2} u(t) \tag{43}$$

Its equivalent discrete-time state-space representation

$$\begin{cases} x(t+1) = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} x(t) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(t) \\ y(t) = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} \end{bmatrix} x(t) \end{cases}$$

is obtained by setting

$$\ddot{y}(t) \approx \frac{\dot{y}(t+T) - \dot{y}(t)}{T} \tag{44}$$

$$\dot{y}(t) \approx \frac{y(t+T) - y(t)}{T} \tag{45}$$

with $T = 1\,s$. The problem of regulate the system to the origin is formulated as an optimization problem, which minimizes the following performance measure

$$\sum_{k=0}^{1} \left\| \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} x_{k+1} \right\|_\infty + |0.8u_k| \tag{46}$$

subject to the input constraints $-1 \leq u_k \leq 1, k = 0, 1.$ and the state constraints $-10 \leq x_k \leq 10, k = 1, 2..$ where $N_y = 2, N_u = 2, Q = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}, R = 0.8$. The solution of this problem is a continuous PWA function, whose surface plot is visualized in Fig. 3(a). According to Bemporad et al. 2002, the eMPC solution is given in Table 1.

Region #	Region	Controller
1	$\begin{bmatrix} 1.00 & 2.00 \\ 0.00 & 1.00 \\ -1.00 & -1.00 \\ 0.80 & -3.20 \\ 1.00 & 1.00 \\ -1.00 & -3.00 \end{bmatrix} x \leq \begin{bmatrix} 11.00 \\ 11.00 \\ 10.00 \\ -2.40 \\ 10.00 \\ -2.00 \end{bmatrix}$	-1.00
2	$\begin{bmatrix} 0.80 & 3.20 \\ -1.00 & -2.00 \\ -1.00 & -1.00 \\ 1.00 & 1.00 \\ 0.00 & -1.00 \\ 1.00 & 3.00 \end{bmatrix} x \leq \begin{bmatrix} -2.40 \\ 11.00 \\ 10.00 \\ 10.00 \\ 11.00 \\ -2.00 \end{bmatrix}$	1.00
3	$\begin{bmatrix} 0.53 & 2.13 \\ 0.67 & 0.67 \\ -1.00 & -1.00 \\ -0.33 & -1.33 \end{bmatrix} x \leq \begin{bmatrix} 0.00 \\ 0.00 \\ 10.00 \\ 1.00 \end{bmatrix}$	$[-0.33, -1.33]x$
4	$\begin{bmatrix} -0.80 & -3.20 \\ 1.00 & 3.00 \\ -1.00 & -1.00 \end{bmatrix} x \leq \begin{bmatrix} 0.00 \\ 0.00 \\ 10.00 \end{bmatrix}$	0
5	$\begin{bmatrix} -1.00 & -1.00 \\ 0.50 & 0.50 \\ -0.80 & -2.40 \\ 0.50 & 1.50 \end{bmatrix} x \leq \begin{bmatrix} 10.00 \\ 0.00 \\ 0.00 \\ 1.00 \end{bmatrix}$	$[-0.50, -1.50]x$
6	$\begin{bmatrix} 0.80 & 3.20 \\ -1.00 & -3.00 \\ 1.00 & 1.00 \end{bmatrix} x \leq \begin{bmatrix} 0.00 \\ 0.00 \\ 10.00 \end{bmatrix}$	0
7	$\begin{bmatrix} 1.00 & 1.00 \\ -0.50 & -0.50 \\ 0.80 & 2.40 \\ -0.50 & -1.50 \end{bmatrix} x \leq \begin{bmatrix} 10.00 \\ 0.00 \\ 0.00 \\ 1.00 \end{bmatrix}$	$[-0.50, -1.50]x$
8	$\begin{bmatrix} -0.53 & -2.13 \\ -0.67 & -0.67 \\ 1.00 & 1.00 \\ 0.33 & 1.33 \end{bmatrix} x \leq \begin{bmatrix} 0.00 \\ 0.00 \\ 10.00 \\ 1.00 \end{bmatrix}$	$[-0.33, -1.33]x$

Table 1. Conventional Representation of the MPC Solution

Using Lemma 3, we can get

$$\Psi = \begin{bmatrix} 1\,0\,1\,0\,1\,0\,1\,1 \\ 1\,1\,0\,1\,0\,1\,0\,0 \\ 1\,0\,1\,0\,1\,0\,1\,1 \\ 1\,0\,1\,1\,0\,1\,0\,1 \\ 1\,0\,1\,0\,1\,0\,1\,1 \\ 1\,0\,0\,1\,1\,1\,1\,0 \\ 1\,0\,0\,1\,1\,1\,1\,0 \\ 1\,0\,1\,1\,0\,1\,0\,1 \end{bmatrix} \tag{47}$$

By applying Lemma 4 and 5, we can further get a simplified structure matrix

$$\tilde{\Psi} = \begin{bmatrix} 1\,0\,1\,0\,1 \\ 0\,1\,0\,0\,0 \\ 0\,0\,1\,1\,0 \\ 0\,0\,0\,1\,1 \end{bmatrix} \tag{48}$$

The corresponding parameter matrix is

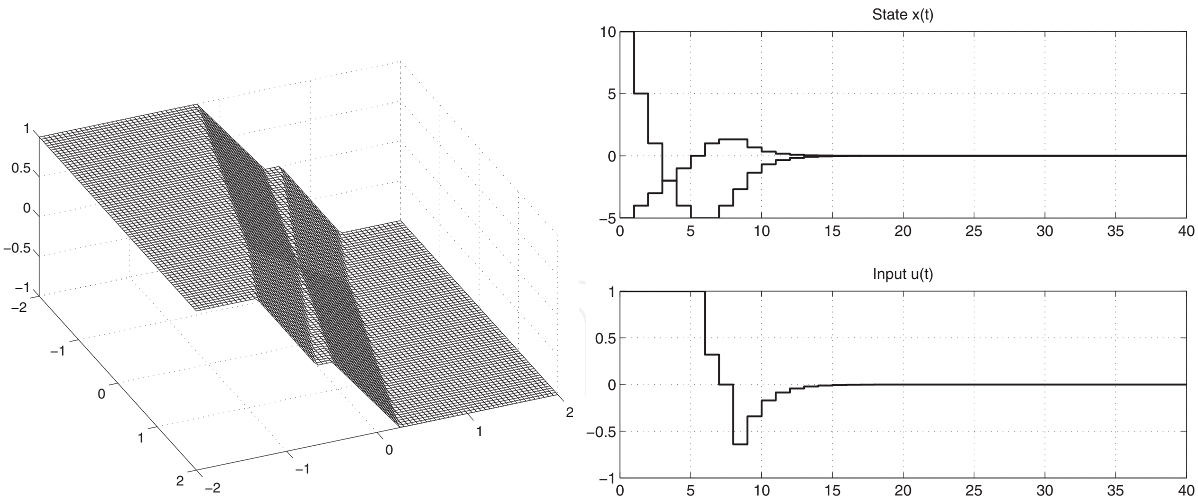
$$\tilde{\Phi} = \begin{bmatrix} 0 & 0 & -1.00 \\ 0 & 0 & 1.00 \\ -0.33 & -1.33 & 0 \\ 0 & 0 & 0 \\ -0.50 & -1.50 & 0 \end{bmatrix} \tag{49}$$

Therefore, the analytical expression of the MPC control law is written as

$$u(x) = \min\{1, \max\{-1, -0.33x_1 - 1.33x_2, -0.50x_1 - 1.50x_2\}, \tag{50}$$
$$\max\{0, -0.33x_1 - 1.33x_2\}, \max\{(0, -0.50x_1 - 1.50x_2\}\}$$

Here it is easy to see that the online MPC optimization is reduced to a lattice PWA function evaluation problem. Same as Bemporad et al. (2002), we consider the starting point $x(0) = [10, -5]^T$. This point is substituted into (50), and the corresponding control action is $u(x) = 1$, which is obtained without any optimization calculations and table searching procedures. The closed-loop response is shown in Fig. 3(b), which is exactly the same with the results from online optimization in Bemporad et al. 2002. The required memory in the analytical expression is to store a structure matrix $\tilde{\Psi} \in \Re^{4\times5}$ and a parameter matrix $\tilde{\Phi} \in \Re^{5\times3}$. The total memory is 35, which is much smaller than the memory space used in Table 1. The online computation requires 7 comparison operations, 8 multiplications and 5 summations. It is evident that the lattice PWA MPC control law performs better in term of online calculation and memory requirements than the conventional eMPC solution.

Example 4: This example is to demonstrate the performance of lattice representation for eMPC solutions from parametric quadratic program. The 2-norm is used in the stage cost function.



(a) Surface plot

(b) Close loop response

Fig. 3. Lattice eMPC Solution

Consider the following state space representation:

$$\begin{cases} x(t+1) = \begin{bmatrix} 0.7326 & -0.0861 \\ 0.1722 & 0.9909 \end{bmatrix} x(t) + \begin{bmatrix} 0.0609 \\ 0.0064 \end{bmatrix} u(t) \\ y(t) = \begin{bmatrix} 0 & 1.4142 \end{bmatrix} x(t) \end{cases} \tag{51}$$

The constraints on input are $-2 \leq u(t) \leq 2$. The corresponding optimization problem for regulating to the origin is written as follows:

$$\begin{aligned} &\min_{u_t,u_t+1} x^T_{t+2|t} P x_{t+2|t} + \sum_{k=0}^{1} \left[x^T_{t+k|t} Q x_{t+k|t} + R u^2_{t+k} \right] \\ &s.t. -2 \leq u(t+k) \leq 2, \qquad k = 0, 1 \\ &\quad x_{t|t} = x(t) \end{aligned} \tag{52}$$

where $P \in \Re^{2\times 2}$ is the solution of the Lyapunov equation $P = A^T P A + Q, Q = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, R = 0.01, N_u = N_y = N_c = 2$. According to Bemporad et al. (2000), the eMPC solution is provided in Table 2.
Using Lemma 3, we can get the following structure matrix.

$$\Psi = \begin{bmatrix} 1&0&0&1&0&1&1 \\ 0&1&1&1&0&1&0 \\ 0&1&1&1&0&1&0 \\ 0&0&0&1&1&1&1 \\ 0&0&0&1&1&1&1 \\ 1&0&0&1&0&1&1 \\ 1&0&0&1&0&1&1 \end{bmatrix} \tag{53}$$

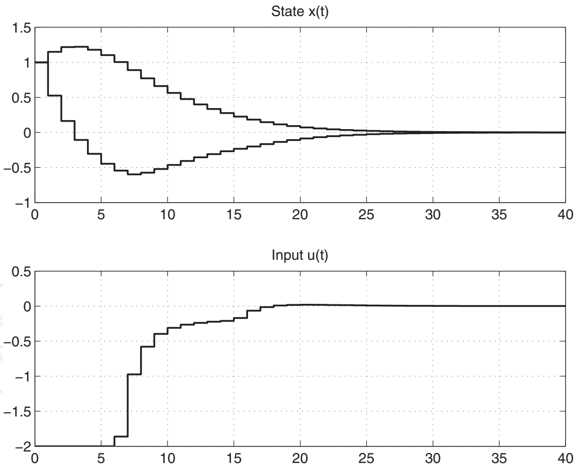


Fig. 4. Closed-loop response of the lattice PWA feedback law

By applying Lemma 4 and 5, we can further get the much simplified structure matrix by deleting all the redundant rows and columns

$$\tilde{\Psi} = \begin{bmatrix} 0\,1\,0\,0\,0 \\ 0\,0\,1\,1\,0 \\ 1\,0\,1\,0\,1 \end{bmatrix} \tag{54}$$

According to the third column of Table 2, the corresponding parameter matrix is

$$\tilde{\Phi} = \begin{bmatrix} -5.9220 & -6.8883 & 0 \\ 0 & 0 & 2 \\ 0 & 0 & -2 \\ -6.4159 & -4.6953 & 0.6423 \\ -6.4159 & -4.6953 & -0.6423 \end{bmatrix} \tag{55}$$

Finally, the analytical expression of the eMPC solution is written as

$$\begin{aligned} u(x) = \min\{&2, \max\{-2, -6.4159x_1 - 4.6953x_2 + 0.6423\}, \\ &\max\{-2, -5.9220x_1 - 6.8883x_2, -6.4159x_1 - 4.6953x_2 - 0.6423\}\} \end{aligned} \tag{56}$$

The closed-loop response of the states is depicted in Fig. 4, which is the same as the one obtained from the online optimization in Bemporad et al. (2000). The required memory in the analytical expression is to store a structure matrix $\tilde{\Psi} \in \Re^{3\times5}$ and a parameter matrix $\tilde{\Phi} \in \Re^{5\times3}$. It is evident that the lattice representation requires much smaller memory space than Table 2. Therefore, the lattice PWA representation of an eMPC solution saves both memory space and online calculation requirements.

Now we discuss the scalability of the lattice representation algorithm. Here this system is solved using different prediction horizons. The simulation results are summarized in Table 3, where N denotes the prediction horizon, τ_{mpt} is the time in seconds to compute the original PWA control, τ_{lat} is the computation time in seconds to build the lattice representation. Note that both representation algorithms give the same lattice eMPC solutions and $\hat{\tau}_{lat} < \tau_{lat}$ hold for all prediction horizons.

No.	Region	Controller
1	$\begin{bmatrix} -5.9220 & -6.8883 \\ 5.9229 & 6.8883 \\ -1.5379 & 6.8296 \\ 1.5379 & -6.8296 \end{bmatrix} x \leq \begin{bmatrix} 2.0000 \\ 2.0000 \\ 2.0000 \\ 2.0000 \end{bmatrix}$	$-[5.9220, 6.8883]x$
2,4	$\begin{bmatrix} -3.4155 & 4.6452 \\ 0.1044 & 0.1215 \\ 0.1259 & 0.0922 \end{bmatrix} x \leq \begin{bmatrix} 2.6341 \\ -0.0353 \\ -0.0267 \end{bmatrix}$	2.0000
3	$\begin{bmatrix} 0.0679 & -0.0924 \\ 0.1259 & 0.0922 \end{bmatrix} x \leq \begin{bmatrix} -0.0524 \\ -0.0519 \end{bmatrix}$	2.0000
5	$\begin{bmatrix} -0.1259 & -0.0922 \\ -0.0679 & 0.0924 \end{bmatrix} x \leq \begin{bmatrix} -0.0519 \\ -0.0524 \end{bmatrix}$	-2.0000
6	$\begin{bmatrix} -6.4159 & -4.6953 \\ -0.0275 & 0.1220 \\ 6.4159 & 4.6953 \end{bmatrix} x \leq \begin{bmatrix} 1.3577 \\ -0.0357 \\ 2.6423 \end{bmatrix}$	$-[6.4159, 4.6953]x$ $+0.6423$
7,8	$\begin{bmatrix} 3.4155 & -4.6452 \\ -0.1044 & -0.1215 \\ -0.1259 & -0.0922 \end{bmatrix} x \leq \begin{bmatrix} 2.6341 \\ -0.0353 \\ -0.0267 \end{bmatrix}$	-2.0000
9	$\begin{bmatrix} 6.4159 & 4.6953 \\ 0.0275 & -0.1220 \\ -6.4159 & -4.6953 \end{bmatrix} x \leq \begin{bmatrix} 1.3577 \\ -0.0357 \\ 2.6423 \end{bmatrix}$	$-[6.4159, 4.6953]x$ -0.6423

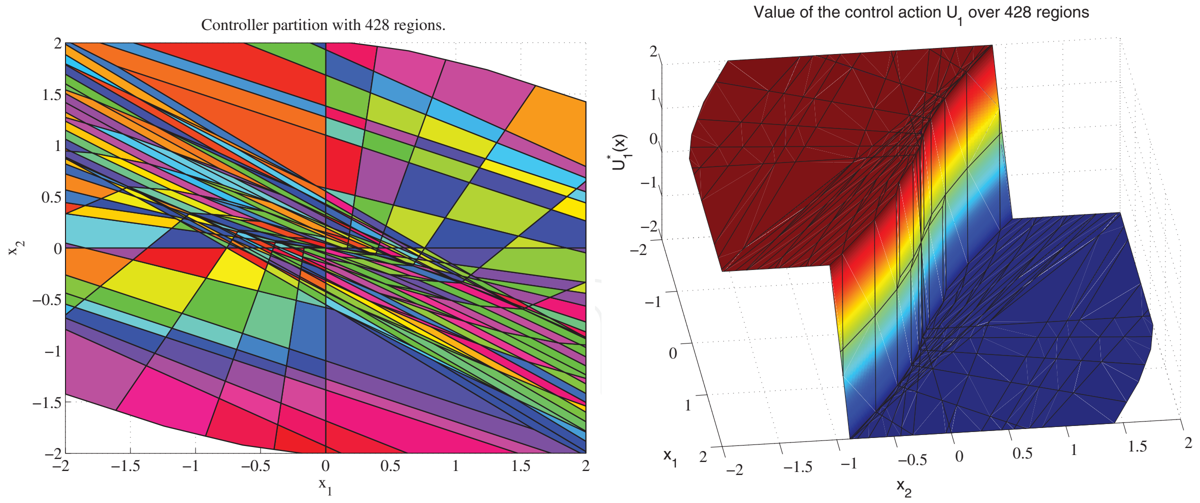
Table 2. Conventional Representation of the eMPC Solution

According to Table 3, the number of affine regions in the eMPC control law increased considerably with the length of prediction horizons. However, all the eMPC laws can be represented by a single lattice PWA function with 19 lattice regions and 12 super-regions. For example, when $N = 28$, the original PWA controller consists of 894 polyhedral regions. But there are only 19 unique affine functions. Therefore, the complexity of the lattice representation is very robust to the length of prediction horizons.

Fig. 5(a) shows the domain partition of the eMPC control when $N = 8$. The corresponding surface plot is visualized in Fig. 5(b). The lattice representation requires to store a 19×3 parameter matrix and a 12×19 structure matrix. The total memory needed is 285. This implies a significant saving in the storage by considering that there are 894 affine functions in the original PWA controller. In the worst case, the online-calculation complexity is $O(234)$. This means a high computation efficiency because the complexity of the competing methods in Table 3 are all specified by the original MPC solution with 894 polyhedral regions.

In this example, the off-line computation is mainly determined by the calculation of the original PWA controllers. For the complex cases with long prediction horizons, the time for building the lattice presentation is one order magnitude less than that for calculating the initial controllers. This makes negligible the additional off-line computation cost for a lattice representation when $N \geq 20$.

Note that the more complicated eMPC laws generally have more polyhedral regions with equal affine control laws. For an instance, when $N = 28$, 875 regions can be merged into one of 19 regions. By comparison, the number of such regions is only 409 when $N = 12$. Therefore, better performance can be anticipated from the lattice representation regarding the



(a) Domain partition

(b) Surface plot

Fig. 5. eMPC solution when $N = 8$

storage, on-line calculation and off-line preprocessing cost when dealing with the complex MPC solutions with large number of polyhedral regions.

N	M	$\tilde{M}$	$\hat{M}$	τ_{mpt}	τ_{lat}
8	254	19	12	11.73	2.06
12	428	19	12	31.19	5.46
16	697	19	12	80.59	15.77
20	863	19	12	174.66	26.23
24	892	19	12	296.05	28.25
28	894	19	12	429.55	28.39

Table 3. Performance of lattice representation for an eMPC solution with different prediction horizons

Example 5: Consider the following linear system with three states and two inputs (Christophersen et al, 2007)

$$x(t+1) = \begin{bmatrix} 0.7 & -0.1 & 0 \\ 0.2 & -0.5 & 0.1 \\ 0 & 0.1 & 0.1 \end{bmatrix} x(t) + \begin{bmatrix} 0.1 & 0 \\ 0.1 & 1 \\ 0.1 & 0 \end{bmatrix} u(t). \tag{57}$$

The system is subject to input constraints $-1 \leq u_i(t) \leq 1$ with $i = 1, 2$, and state constraints $-20 \leq x_i(t) \leq 20$ with $i = 1, 2, 3$. The constrained finite time optimal control problem is solved with $p = 1, N_y = 8$, $Q = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$, $R = \begin{bmatrix} 0.1 & 0 \\ 0 & 0.1 \end{bmatrix}$ and $P = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$. The MPT toolbox is used to generate the eMPC control law $u(x) = [u_1(x), u_2(x)]^T$ with 813.4 seconds. The vector eMPC control is defined over 2497 regions.

Two scalar lattice PWA functions are used to describe $u(x)$, which requires 2 structure matrices $\Psi_1 \in \Re^{18\times21}, \Psi_2 \in \Re^{3\times5}$ and two parameter matrices $\Phi_1 \in \Re^{21\times4}, \Phi_1 \in \Re^{5\times4}$. The total memory required is 497 in the worst case. In addition, the online computational complexity is calculated as 616. The average evaluation time of the lattice eMPC is 3.01×10^{-4} seconds by randomly sampling $20,000$ states in the feasible region. In the preprocessing stage, it takes 11.4 seconds to build the lattice PWA function.

Table 4 lists the comparison results of SS algorithm, CVF algorithm (Christophersen et al, 2007), BST algorithm (Tondel et al. 2003), BBT algorithm (Christophersen et al, 2007) and the proposed lattice representation (LR) algorithm. The LR algorithm has the lowest description complexity. It saves more than 90% memory storage compared with the most efficient competing algorithm. The BST algorithm has the lowest online computational complexity. But it requires more than 5 millions linear programs to construct the binary search tree. The pre-processing time is 3504.2 seconds, which is much longer than the time for building the eMPC control. The SS, BBT and BST algorithms require shorter pre-processing time than the LR algorithm. However, the former algorithms have much higher evaluation and description complexities than the latter. In summary, the LR algorithm offers a significant reduction in memory storage. Therefore, the LR method has a better overall performance than other competing methods by considering description, online and off-line computational complexities.

In order to test the scalability of the LR algorithm, we solve system (57) with different horizons. The simulation results are summarized in Table 5, where $N, M, \tau_{mpt}, \hat{\tau}_{lat}$ are defined as above in Example 4. According to Table 5, the number of affine regions in the eMPC control law increased considerably with horizon N. However, the complexity of the vector lattice representation is very robust to the prediction horizons. The description and evaluation complexities of lattice representation keep constant, although the number of regions increase from 1600 ($N = 12$) to 3189 ($N = 20$) in the original eMPC controls. In all cases, the preprocessing time for building the lattice representations is negligible compared with the time for the eMPC solutions using MPT toolbox. Therefore, the LR algorithm is promising for large-scale eMPC problems with long prediction horizons.

	ϕ	ψ	τ
SS Algorithm	81661	106295	0
CVF Algorithm	9988	14982	0
BST Algorithm	21368	110	3504.2
BBT Algorithm	30104	923	10.0
LR Algorithm	497	616	11.4

Table 4. Comparison of Description, Evaluation and Preprocessing Complexities

N	M	$\bar{M}_{u1}$	$\hat{M}_{u1}$	$\bar{M}_{u2}$	$\hat{M}_{u2}$	τ_{mpt}	τ_{lat}
4	1510	14	17	3	5	86.0	6.0
8	2497	18	21	3	5	813.4	11.4
12	2600	18	21	3	5	1658.3	12.0
16	3189	18	21	3	5	3385.6	14.7
20	3189	18	21	3	5	5589.8	14.7

Table 5. Performance of LR Algorithm for eMPC Solutions with Different Horizons

6. Conclusions

This paper proposes a general lattice PWA representation theorem for continuous eMPC solutions. A constructive proof is developed to show that the global structure information of an eMPC controller can be fully represented by a set of scalar lattice PWA functions. A lattice PWA function is a global and compact representation of an eMPC solution, because the redundant parameters are automatically removed by utilizing the continuity of eMPC controllers. A lattice representation has an explicit analytical expression. This facilitates the implementation of lattice eMPC solutions using low-cost hardware.

A complexity reduction algorithm is proposed to develop a computationally efficient lattice representation of eMPC controllers obtained from multi-parametric program. The description and evaluation complexities of a lattice eMPC controller depend on the number of super-regions instead of the number of polyhedral regions. Therefore, the lattice representation of eMPC solutions can reduce computation and memory requirements significantly, when the original MPC solutions have many polyhedral regions with equal affine control laws.

The class of lattice PWA functions provides a compact and explicit model structure for continuous eMPC solutions. The global structure information in an eMPC control is utilized to reduce the complexity of its lattice representation. The lattice eMPC controllers present the first step to use a global PWA representation model to describe generic eMPC solutions. Further investigation is needed to generalize the lattice representation into the discontinuous eMPC solutions.

7. References

[1] Alessio A. & Bemporad A. (2009), "A survey on explicit model predictive control," in Proc. Int. Workshop on Assessment and Future Directions of Nonlinear Model Predictive Control (Pavia, Italy), D.M. Raimondo L. Magni, F. Allgower, Ed., *Lecture Notes in Control and Information Sciences*, 2009.

[2] Beccuti, A.G., Mariethoz, S., Cliquennois, S., Wang, S., & Morari, M. (2009) "Explicit Model Predictive Control of DC-DC Switched Mode Power Supplies with Extended Kalman Filtering", *IEEE Trans. on Industrial Electronics*, vol. 56, no. 5, To appear.

[3] Bemporad A., Borelli F. & Morari, M. (2002) "Model predictive control based on linear programming-The explicit solution." *IEEE Transactions on Automatic Control*, vol. 47 (12), 1974-1984.

[4] Bemporad, A. , Morari, M., Dua, V. & Pistikopoulos, E. N. (2002), "The explicit linear quadratic regulator for constrained systems," *Automatica*, vol. 38 (1), 3-20.

[5] Bemporad A. & Filippi, C. (2003) "Suboptimal explicit RHC via approximate quadratic programming," *Journal of Optimization Theory and Application*, vol. 117(1), 5-38.

[6] Bertsekas, D. P. & Tsitsiklis, J. N. (1998) "Neuro-dynamic prgramming." Belmont: Athena Scientific.

[7] Boom, T. J. J. & Schutter, B. D. (2002) "Model predictive control for perturbed max-plus-linear systems," *Systems & Control Letters*, vol. 45, no. 1, pp. 21-33.

[8] Borrelli, F., Baotic, M., Bemporad, A., & Morari, M. (2001) "Efficient on-Line computation of constrained optimal control." *IEEE Conference on Decision and Control*, Orlando, Florida, pp. 1187-1192.

[9] Chua, L. O. & Kang, S. M. (1977) “Section-wise piecewise-linear functions: canonical representation, properties, and applications." *IEEE Trans, Circuits Syst. I*, 24(1), 125-140.

[10] Christophersen, F. J., Kvasnica, M., Jones, C. N., & Morari, M. (2007) “Efficient evaluation of piecewise control laws defined over a large number of polyhedra", *European Control Conference*, Kos, Greece.

[11] Chikkula, Y., Lee, J. H. & Ogunnaike, B. A. (1998) “Dynamically scheduled MPC of nonlinear processes using hinging hyperplane models.", *AIChE Journal*, vol. 44, pp. 2658-2674.

[12] Dua, P., Sakizlis, V., Dua, V., Doyle III, F.J., Pistikopoulos, E. N. (2004) “Model based control for insulin delivery for Type 1 diabetics via parametric programming", *Proceedings of European Symposium on Computer Aided Process Engineering*, Lisbon, pp. 1045-1050.

[13] Dua, P., Kouramas, K., Dua, V., & Pistikopoulos, E. N. (2008) “MPC on a chip-Recent advances on the application of multi-parametric model-based control." *Computers & Chemical Engineering*, Vol. 32, No. 4-5, pp. 754-765.

[14] Geyer, T., Torrisi, F. D., Morari, M. (2008) “Optimal complexity reduction of polyhedral piecewise affine systems." *Automatica*, vol. 44, no. 7: pp. 1728-1740.

[15] Grancharova, A., Johansen, T. A. & Kocijan, J. (2004)“Explicit model predictive control of gas-liquid separation plant", *Computers & Chemical Engineering*, Vol. 28, No. 12,pp. 2481-2491.

[16] Grancharova, A. & Johansen, T. A. (2005) “Survey of explicit approaches to constrained optimal control", *Switching and Learning in Feedback Systems*, R. Shorten and R. Murray-Smith, Ed., LNCS 3355, Springer Verlag

[17] Hegrenas, O., Gravdahl, J. T. & Tondel, P. (2007) “ Spacecraft attitude control using explicit model predictive control." *Automatica*, Vol. 43, No. 12, pp. 2107-2114 .

[18] Johansen, T. A., Jackson, W., Schreiber, R. & Tondel, P. (2007) “Hardware Synthesis of Explicit Model Predictive Controllers." *IEEE Transactions Control Systems Technology*, vol. 15, pp. 191-197.

[19] Johansen, T. A., Petersen I. & Slupphaug, O. (2002) “Explicit suboptimal linear quadratic regulation with input and state constraints," *Automatica*, vol. 38(7), 1099-1112.

[20] Johansen, T. A., & Grancharova, A. (2003) “Approximate explicit constrained linear model predictive control via orthogonal search tree." *IEEE Trans. Automatic Control*, vol.48, No.5, pp.810-815.

[21] Jones,C. N., Grieder, P. & Rakovic, S. (2006) “A Logarithmic-Time Solution to the Point Location Problem for Parametric Linear Programming." *Automatica*, vol. 42, no. 12, pp. 2215-2218.

[22] Julian, P., Desages, A. & Agamennoni, O. (1999). “High-level canonical piecewise linear representation using a simplicial partition." *IEEE Transactions on Circuits Systems I*, 46(4) 463-480.

[23] Kvasnica, M., Grieder P. & Baotić, M. (2004) “Multi-parametric toolbox (MPT)," http://control.ee.ethz.ch/ mpt/.

[24] Li X., Wang, S. & Yin, W. (2001) “A canonical representation of high-dimensional continuous piecewise-linear functions." *IEEE Transactions on Circuits Systems I*, 48(11), 1347-1351.

[25] Lin J. & Unbehauen, R. (1995)“Canonical Piecewise-Linear networks. *IEEE Trans. Neural Networks. I*, 6(1), 43-50.

[26] Lincoln B. & Rantzer, A. (2006) "Relaxed dynamic programming." *IEEE Transactions on Automatic Control*, vol. 51 (8), 1249-1260.

[27] Lincoln B. & Rantzer, A. (2002) "Suboptimal dynamic programming with error bounds." *Proceedings of the IEEE conference on decision and control*, Las Vegas, NV

[28] Mayne, D. Q., Rawlings, J. B., Rao, C. V., & Scokaert, P. O. M. (2000) "Constrained model predictive control: Stability and optimality." *Automatica*, Vol. 36, No. 6, pp. 789-814.

[29] Morari, M. & Lee, J (1999) "Model predictive control: past, present and future." *Computers & Chemical Engineering*, Vol. 23, No. 4-5, pp. 667-682.

[30] Necoara, I., Schutter, B. D., Boom, T. J. J. & Hellendoorn, H. (2008) "Model predictive control for uncertain max-min-plus-scaling systems," *International Journal of Control*, vol. 81, no. 5, pp. 701-713.

[31] Ovchinnikov, S. (2002) "Max-Min representation of piecewise linear functions," *Contributions to Algebra and Geometry*, vol. 43, no. 1, pp. 297Ű302.

[32] Pannocchia, G., Rawlings, J. B. & Wright, S. J. (2007) "Fast, large-scale model predictive control by partial enumeration." *Automatica*, Vol. 43, No. 5, pp. 852-860.

[33] Pistikopoulos, E. N., Dua, V., Bozinis, N. A., Bemporad A. & Morari, M. (2001) "On-line optimization via off-line parametric optimization tools." *Computers and Chemical Engineering*, vol. 26: 175-185.

[34] Ryu, J. & Pistikopoulos, E. N. (2007) " A novel approach to scheduling of zero-wait batch processes under processing time variations." *Computers & Chemical Engineering*, Vol. 31, No. 3, pp. 101-106.

[35] Saffer, D. R. II & Doyle, F. J. III, (2004) "Analysis of linear programming in model predictive control," *Computers and Chemical Engineering*, vol. 28: 2749-2763.

[36] Spjotvold, J., Rakovic, S. V., Tondel, P., Johansen, T. A. (2006) "Utilizing Reachability Analysis in Point Location Problems." *IEEE Conf. Decision and Control*, San Diego.

[37] Spjotvold, J., Kerrigan, E. C., Jones, C. N., Tondel, P. & Johansen, T. A. (2006) "On the Facet-to-Facet Property of Solutions to Convex Parametric Quadratic Programs", *Automatica*, Vol. 42, pp. 2209-2204.

[38] Spjotvold, J., Tondel, P. & Johansen, T. A. (2007) "A continuous selection and unique polyhedral representation of solutions to convex multiparametric quadratic programs", *Journal of Optimization Theory and Applications*, Vol. 134, pp. 177-189.

[39] Tarela, J. M. & Martínez, M. V. (1999) "Region configurations for realizability of lattice piecewise-linear models," *Mathematical and Computer Modelling*, Vol.30 (11-12), pp.17-27.

[40] Tondel, P., Johansen T. A. & Bemporad, A. (2003) "Evaluation of piecewise affine control via binary search tree." *Automatica*, Vol. 39, pp. 743-749.

[41] Wen, C., Wang, S., Zhang, H. & Khan, M. J. (2005a) "A novel compact piecewise-linear representation." *International Journal of Circuit Theory and Applications* Vol.33(1), 87-97.

[42] Wen, C., Wang, S., Li, F.& Khan, M. J. (2005b) "A compact $f - f$ model of high-dimensional piecewise-linear function over a degenerate intersection." *IEEE Transactions on Circuits Systems I*, Vol. 52(4), 815-821.

[43] Wen, C., Wang, S., Sun, X., Zhang, H., & Li, F. (2005c) "A global compact representation for three-dimensional continuous piecewise linear functions", *Journal of Tsinghua University*, 45 (4): 533-536.

[44] Wen, C., Wang, S. (2005d) "A complete 3-D canonical piecewise-linear representation", *International Conference on Technology and Automation- ICTA'05*, Thessaloniki, Greece.

[45] Wen, C. (2006) *A canonical piecewise-linear representation theorem and its applications in function approximation,* Ph.D Theses, Automation Department, Tsinghua University.

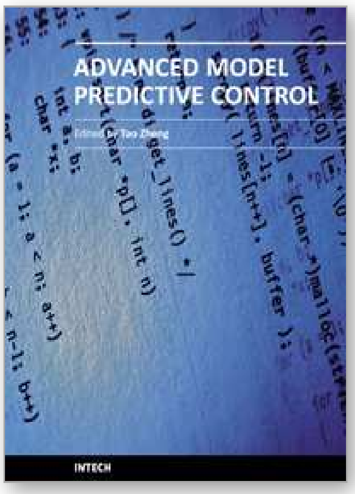
[46] Wen, C, Wang, S., Jin X. & Ma, X. (2007) “Identification of dynamic systems using piecewise-affine basis function models." *Automatica*, vol. 43(10): 1824-1831.

[47] Wen, C & Ma, X. (2008a) “A max-piecewise-linear neural network for function approximation." *Neurocomputing*, Vol. 71(4-6), 843-852.

[48] Wen, C.,& Ma, X. (2008b) “A Basis-Function Canonical Piecewise-Linear Approximation." *IEEE Transactions on Circuits and Systems I: Regular Papers*, Vol. 55, no. 5, pp. 1328-1334.

[49] Wen, C., Ma, X. & Ydstie, B. E. (2009a)“Analytical expression of explicit MPC solution via lattice piecewise-affine function",*Automatica*, Vol. 45, No. 4, pp. 910-917.

[50] Wen, C., & Ydstie, B. E. (2009b) “Lattice piecewise-affine representation for explicit model predictive control", *AIChE Annual Meeting*, Nashville, TN, November.



Advanced Model Predictive Control
Edited by Dr. Tao ZHENG

ISBN 978-953-307-298-2
Hard cover, 418 pages
Publisher InTech
Published online 24, June, 2011
Published in print edition June, 2011

Model Predictive Control (MPC) refers to a class of control algorithms in which a dynamic process model is used to predict and optimize process performance. From lower request of modeling accuracy and robustness to complicated process plants, MPC has been widely accepted in many practical fields. As the guide for researchers and engineers all over the world concerned with the latest developments of MPC, the purpose of "Advanced Model Predictive Control" is to show the readers the recent achievements in this area. The first part of this exciting book will help you comprehend the frontiers in theoretical research of MPC, such as Fast MPC, Nonlinear MPC, Distributed MPC, Multi-Dimensional MPC and Fuzzy-Neural MPC. In the second part, several excellent applications of MPC in modern industry are proposed and efficient commercial software for MPC is introduced. Because of its special industrial origin, we believe that MPC will remain energetic in the future.

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821