Delft University of Technology

Delft Center for Systems and Control

Technical report 14-026

Irredundant lattice piecewise affine representations and their applications in explicit model predictive control*

J. Xu, T.J.J. van den Boom, and B. De Schutter

If you want to cite this report, please use the following reference instead:
J. Xu, T.J.J. van den Boom, and B. De Schutter, "Irredundant lattice piecewise affine representations and their applications in explicit model predictive control," *Proceedings of the 53rd IEEE Conference on Decision and Control*, Los Angeles, California, pp. 4416–4421, Dec. 2014.

Delft Center for Systems and Control
Delft University of Technology
Mekelweg 2, 2628 CD Delft
The Netherlands
phone: +31-15-278.51.19 (secretary)
fax: +31-15-278.66.79
URL: http://www.dcsc.tudelft.nl

Irredundant lattice piecewise affine representations and their applications in explicit model predictive control

Jun Xu, Ton J.J. van den Boom, and Bart De Schutter

Abstract— **In this paper, we derive the irredundant lattice piecewise affine (PWA) representation, which is capable of representing any continuous PWA function. Necessary and sufficient conditions for irredundancy are proposed. Besides, we discuss how to remove redundant terms and literals and propose corresponding necessary and sufficient conditions. In a worked example, the irredundant lattice PWA representation is used to express the explicit model predictive controller of a linear system, and the result turns out to be much more compact than that given by the state-of-the-art algorithm.**

I. INTRODUCTION

In [1], the lattice piecewise affine (PWA) representation is proposed to generate PWA functions for the approximation of a nonlinear function, and the construction of the lattice PWA representation is basically a proof of the fact that any continuous PWA function can be represented by the lattice PWA representation. In [2], a formal proof is given demonstrating the representation ability. In fact, the lattice PWA representation is similar to the canonical max-min-plus-scaling (MMPS) function representation [3], [4], which can be described as

$$f = \min_{i=1,\ldots,N_1} \{\max_{j\in \bar{I}_i}\{\ell_j\}\}, \quad (1)$$

or

$$f = \max_{i=1,\ldots,N_2} \{\min_{j\in \tilde{I}_i}\{\ell_j\}\}, \quad (2)$$

in which ℓ_j is an affine function, N_1 and N_2 are integers, the sets $\bar{I}_i$ and $\tilde{I}_i$ are index sets; besides, the operations "max", "min" are performed entrywise. The first expression is referred to as the conjunctive form, while the second is the disjunctive form. The equivalence of continuous PWA functions and canonical MMPS functions has been proved in [5]. In this paper, we mainly focus on the disjunctive lattice PWA representation; however, the results can be easily extended to the conjunctive case due to duality.

Compared with other methods for representing PWA functions [6]–[10], lattice PWA representations are powerful. In fact, the methods of [6]–[8] cannot represent all continuous PWA functions, the parameters in the representation proposed in [9] are not easy to determine, and the number of parameters in the expression in [10] is large. Conversely, the integer N_1 (N_2) and the index set $\bar{I}_i$ ($\tilde{I}_i$) used in the lattice PWA representation are not hard to derive, which will be demonstrated in Section II.

Lattice PWA representations have been used to express the solution of explicit model predictive control (MPC) problems in [11]. Traditional MPC is an online control method based on optimization at each time step; it can handle a large number of system constraints by incorporating them into the optimization problem. The optimization is performed using a prediction model for predicting future outputs of the system. For linear prediction models with constraints on states and outputs, if the performance criterion is quadratic or based on a mixed $1/\infty$- norm, it is proved in [12], [13] that the optimal solution is a continuous PWA function with respect to the state vector; hence, the optimal solution can be computed offline, and the cost of online optimization can be reduced to that of online evaluation of a continuous PWA function. This is exactly what "explicit" means.

The continuous PWA optimal solution can be computed using multi-parametric quadratic programming through e.g. the MPT toolbox [14] and stored as local affine functions and subregions. It is straightforward to use a lattice PWA representation to express this, and the storage requirement will be reduced compared with the solution given by the MPT toolbox. In [11], it is pointed out that there may be redundant parameters in the lattice PWA expression and [11] gives two lemmas to remove the redundant ones. However, the lemmas have limitations and the result may be redundant. Hence, in this paper, we aim to give an irredundant lattice PWA representation.

The paper is organized as follows. The next section introduces the full lattice PWA representation, and illustrates how to construct one. The *irredundant* lattice PWA representation is derived in Section III, including necessary and sufficient conditions for irredundancy and how to remove redundant terms or literals. Section IV gives a worked example of the application of irredundant lattice PWA representations to express the solution of the explicit MPC problem. Finally, the paper ends with conclusions in Section V.

II. FULL LATTICE PWA REPRESENTATION

The full lattice PWA representation is capable of representing any single-valued continuous PWA function, which is defined through the following definition.

This work was supported by China Scholarship Council (CSC), National Natural Science Foundation of China (61104218; 21006127), the National Basic Research Program of China (973 program: 2012CB720500), the Science Foundation of China University of Petroleum (YJRC-2013-12).

Jun Xu is with the Department of Automation, China University of Petroleum, 102249, Beijing, China. yun-xu05@mails.tsinghua.edu.cn

Ton J.J. van den Boom and Bart De Schutter are with Delft Center for Systems and Control, Delft University of Technology, 2628 CD Delft, The Netherlands. a.j.j.vandenBoom@tudelft.nl, b.deschutter@tudelft.nl

Definition 1: A function $f : \mathbb{D} \to \mathbb{R}$, where $\mathbb{D} \subseteq \mathbb{R}^n$ is convex, is said to be continuous PWA if the following conditions are satisfied:

1) The domain space $\mathbb{D}$ is divided into a finite number of nonempty convex polyhedra, i.e., $\mathbb{D} = \cup_{i=1}^{N} \mathbb{D}_i,\ \mathbb{D}_i \neq \emptyset$, the polyhedra are closed and have non-overlapping interiors, $\mathrm{int}(\mathbb{D}_i) \cap \mathrm{int}(\mathbb{D}_j) = \emptyset,\ \forall i, j \in \{1, \ldots, \hat{N}\}, i \neq j$. These polyhedra are also called base regions. The boundaries of the polyhedra are nonempty sets in $(n-1)$-dimensional space.
2) In each base region $\mathbb{D}_i$, f equals an affine function $\ell_{\mathrm{act}(i)}$:
$$f(x) = \ell_{\mathrm{act}(i)}(x),\ \forall x \in \mathbb{D}_i,\ \text{with } i \in \{1, \ldots, M\}, \tag{3}$$
and we call the affine function $\ell_{\mathrm{act}(i)}$ active in the base region $\mathbb{D}_i$. In (3), the integer M is the number of distinct affine functions in f; so no two affine functions ℓ_i and ℓ_j, $i, j \in \{1, \ldots, M\}, i \neq j$, are identical.
3) In each base region $\mathbb{D}_i$, no other affine function intersects with $\ell_{\mathrm{act}(i)}$ in the interior of $\mathbb{D}_i$, i.e.,
$$\{x | \ell_j(x) = \ell_{\mathrm{act}(i)}(x), j \neq \mathrm{act}(i)\} \cap \mathrm{int}(\mathbb{D}_i) = \emptyset. \tag{4}$$
4) f is continuous on the boundaries, i.e.,
$$\ell_{\mathrm{act}(i)}(x) = \ell_{\mathrm{act}(j)}(x),\ \forall x \in \mathbb{D}_i \cap \mathbb{D}_j, \mathbb{D}_i \cap \mathbb{D}_j \neq \emptyset. \tag{5}$$

Remark. In this paper, we define the continuous PWA function with respect to base regions. In fact, the subregions mentioned in [15] can be partitioned into base regions.

The results of [1], [16] can be easily extended to the lattice PWA representation for base regions, which leads to:

Lemma 1: Let f be a continuous PWA function defined in Definition 1. Then f can be represented by
$$f(x) = \max_{i=1,\ldots,N} \{\min_{j \in I_{\geq,i}} \{\ell_j(x)\}\},\ \forall x \in \mathbb{D}, \tag{6}$$
with $I_{\geq,i} = \{j | \ell_j(x) \geq \ell_{\mathrm{act}(i)}(x), \forall x \in \mathbb{D}_i\}$.

We call (6) full lattice PWA representation.

In the base region $\mathbb{D}_i$, as (4) holds, for an affine function ℓ_j with $j \neq \mathrm{act}(i)$, either $j \in I_{\geq,i}$ or $j \in I_{\leq,i}$. If $j \in I_{\geq,i}$, then $\ell_j(x) > \ell_{\mathrm{act}(i)}(x), \forall x \in \mathrm{int}(\mathbb{D}_i)$, else if $j \in I_{\leq,i}$, we have $\ell_j(x) < \ell_{\mathrm{act}(i)}(x), \forall x \in \mathrm{int}(\mathbb{D}_i)$.

In the full lattice PWA representation (6), two binary operations "min" and "max" are present. They are similar to the Boolean AND and OR of Boolean algebra. Analog to the terminology of Boolean algebra, we call "$\min_{j \in I_{\geq,i}} \{\ell_j\}$" a term, denoted by T_i^{F}, in which the superscript "F" indicates that the term corresponds to the full representation. In each term, the affine functions $\ell_j,\ j \in I_{\geq,i}$ are called literals.

We give a simple 1-dimensional example to illustrate the above definition and lemma.

Example 1: Consider a 1-dimensional continuous PWA function f with 5 affine functions:
$$f = \begin{cases} l_1(x) = 0.5x,\ x \in [0, 2], \\ l_2(x) = x - 1,\ x \in [2, 3], \\ l_3(x) = 2,\ x \in [3, 6], \\ l_4(x) = -x + 8,\ x \in [6, 7], \\ l_5(x) = -0.5x + 4.5,\ x \in [7, 9]. \end{cases} \tag{7}$$
The plot of f is shown in Fig. 1. From Fig. 1, we can see

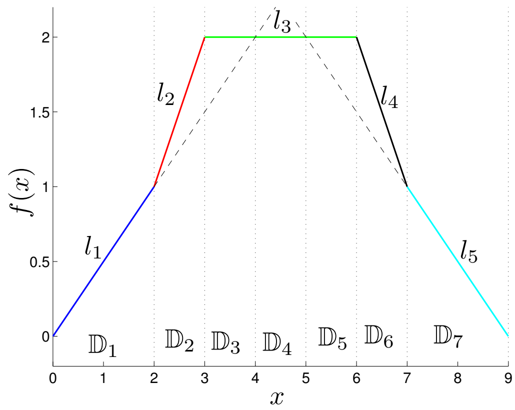

Fig. 1. An example illustrating Theorem 2.

that there are 7 base regions. According to Lemma 1, we have the following 7 index sets and corresponding terms:
$$\begin{array}{l} I_{\geq,1} = \{1,3,4,5\},\ I_{\geq,2} = \{2,3,4,5\},\ I_{\geq,3} = \{2,3,4,5\}, \\ I_{\geq,4} = \{1,2,3,4,5\},\ I_{\geq,5} = \{1,2,3,4\}, \\ I_{\geq,6} = \{1,2,3,4\},\ I_{\geq,7} = \{1,2,3,5\}, \end{array} \tag{8}$$
$$\begin{array}{l} T_1^{\mathrm{F}} = \min\{\ell_1, \ell_3, \ell_4, \ell_5\},\ T_2^{\mathrm{F}} = \min\{\ell_2, \ell_3, \ell_4, \ell_5\}, \\ T_3^{\mathrm{F}} = \min\{\ell_2, \ell_3, \ell_4, \ell_5\},\ T_4^{\mathrm{F}} = \min\{\ell_1, \ell_2, \ell_3, \ell_4, \ell_5\}, \\ T_5^{\mathrm{F}} = \min\{\ell_1, \ell_2, \ell_3, \ell_4\},\ T_6^{\mathrm{F}} = \min\{\ell_1, \ell_2, \ell_3, \ell_4\}, \\ T_7^{\mathrm{F}} = \min\{\ell_1, \ell_2, \ell_3, \ell_5\}. \end{array} \tag{9}$$
The full lattice PWA representation for this example is
$$f(x) = \max_{i=1,\ldots,7} \{\min_{j \in I_{\geq,i}} \{\ell_j(x)\}\} = \max_{i=1,\ldots,7} \{T_i^{\mathrm{F}}\},\ \forall x \in [0, 9]. \tag{10}$$
It is obvious that T_2^{F} and T_3^{F} are the same; so either of them can be deleted without affecting the value of f. Similarly, we can also delete either T_5^{F} or T_6^{F} without affecting the value of f. Besides, a more surprising fact is that removing ℓ_5 from T_1^{F} will also not affect the whole expression. Therefore, the lattice PWA expression (10) is redundant. □

In the next section, we are dedicated to find an irredundant lattice PWA representation.

III. Irredundant Lattice Piecewise Affine Representation

We define the irredundancy of a lattice PWA representation as follows:

Definition 2: A lattice PWA representation
$$f_{\mathrm{L}} = \max_{i=1,\ldots,\tilde{N}} \{T_i\} = \max_{i=1,\ldots,\tilde{N}} \{\min_{j \in I_i} \{\ell_j\}\} \tag{11}$$

with $\tilde{N} \leq N$ is irredundant, if no term $T_i = \min_{j \in I_i}\{\ell_j\}$, and no literal ℓ_j, with $i \in \{1, \ldots, \tilde{N}\}$ and $j \in I_i$, can be deleted without affecting the value of f_{L}.

To achieve irredundancy, analog to the Boolean algebra, we define implicants and prime implicants.

A. Implicants and prime implicants

Definition 3: For a continuous PWA function defined in Definition 1, we say $T_i = \min_{j \in I_i}\{\ell_j\}$ is an implicant of f, if

$$T_i(x) \leq f(x), \forall x \in \mathbb{D}, \tag{12}$$

and there is some base region $\mathbb{D}_k$ such that $T_i \equiv f$ in $\mathbb{D}_k$. The implicant $T_i = \min_{j \in I_i}\{\ell_j\}$ is prime if there exists no other implicant $T_r = \min_{j \in I_r}\{\ell_j\}$ of f such that $I_r \subsetneq I_i$.

For an implicant of f, we also define the base regions in which it is active.

Definition 4: The implicant T_i is called active in the base region $\mathbb{D}_k$, if $T_i \equiv f$ in $\mathbb{D}_k$. The indices of all base regions in which T_i is active constitute an index set $\mathcal{A}(T_i)$.

It is noted that in Definition 1 we call $\ell_{\mathrm{act}(i)}$ active in the base region $\mathbb{D}_i$. Similar to the notation in Definition 4, the indices of all base regions in which $\ell_{\mathrm{act}(i)}$ is active constitute an index set $\mathcal{A}(\ell_{\mathrm{act}(i)})$. It is obvious that $i \in \mathcal{A}(\ell_{\mathrm{act}(i)})$.

In the base region $\mathbb{D}_k$ with $k \notin \mathcal{A}(T_i)$, from the definition of implicant, we have $T_i \leq f, T_i \not\equiv f$. Besides, if $x \in \mathrm{int}(\mathbb{D}_k), k \notin \mathcal{A}(T_i)$, according to (4), we have $T_i(x) < f(x)$.

We now describe the implicants and prime implicants in the context of the lattice PWA representation. The following lemma determines the indices of base regions in which a term is active.

Lemma 2: Given an implicant $T_i = \min_{j \in I_i}\{\ell_j\}$, it is active in the base region $\mathbb{D}_k$, i.e., $k \in \mathcal{A}(T_i)$ if and only if $I_i \subseteq I_{\geq,k}$.

Proof: Necessity. If $k \in \mathcal{A}(T_i)$, we have $I_i \subseteq I_{\geq,k}$. Otherwise, if $I_i \not\subseteq I_{\geq,k}$, we can find an r such that $r \in I_i,\ r \notin I_{\geq,k}$. As $r \notin I_{\geq,k}$, there would exist an $\tilde{x} \in \mathbb{D}_k$, such that $\ell_r(\tilde{x}) < \ell_{\mathrm{act}(k)}(\tilde{x}) = f(\tilde{x})$. Since $r \in I_i$, we have $T_i(\tilde{x}) \leq \ell_r(\tilde{x}) < \ell_{\mathrm{act}(k)}(\tilde{x}) = f(\tilde{x})$, contradicting the fact that $k \in \mathcal{A}(T_i)$.

Sufficiency. If $I_i \subseteq I_{\geq,k}$, as T_i is an implicant, we have $\mathrm{act}(k) \in I_i$, then $T_i \equiv \ell_{\mathrm{act}(k)} \equiv f$ in $\mathbb{D}_k$, i.e., $k \in \mathcal{A}(T_i)$. ■

Lemma 3: Every term $T_i^{\mathrm{F}} = \min_{j \in I_{\geq,i}}\{\ell_j\}$ in the full lattice PWA expression (6) is an implicant of f. Moreover, there exists at least one prime implicant $T_i = \min_{j \in I_i}\{\ell_j\}$ of f with $I_i \subseteq I_{\geq,i}$ and $\mathcal{A}(T_i^{\mathrm{F}}) \subseteq \mathcal{A}(T_i)$.

Proof:

According to Lemma 1, $T_i^{\mathrm{F}} \equiv f$ in $\mathbb{D}_i$, thus $\mathcal{A}(T_i^{\mathrm{F}}) \neq \emptyset$, besides, $T_i^{\mathrm{F}} \leq f$ in $\mathbb{D}$, so T_i^{F} is an implicant of f.

For each T_i^{F}, we first set $I_i = I_{\geq,i}$ and $T_i = \min_{j \in I_i}\{\ell_j\}$. Then, for each $k \in I_i$, if $\min_{j \in I_i \backslash \{k\}}\{\ell_j\} \leq f$, we set $I_i = I_i \backslash \{k\}$ and $T_i = \min_{j \in I_i}\{\ell_j\}$. Thus the resulting $T_i = \min_{j \in I_i}\{\ell_j\}$ satisfies $T_i \leq f$, and for any index set $\tilde{I}_i \subsetneq I_i$, the statement $\min_{j \in \tilde{I}_i}\{\ell_j\} \leq f$ does not hold.

Considering T_i obtained above, now we explain why $\mathcal{A}(T_i^{\mathrm{F}}) \subseteq \mathcal{A}(T_i)$. Assume this does not hold. Then there would exist an index k satisfying $k \in \mathcal{A}(T_i^{\mathrm{F}})$ and $k \notin \mathcal{A}(T_i)$, so in $\mathrm{int}(\mathbb{D}_k)$, we have $T_i^{\mathrm{F}} \equiv f$, and $T_i < f$,, which contradicts the fact that $T_i^{\mathrm{F}} \leq T_i$.

From the above proof, the resulting T_i satisfying $T_i \leq f$ in $\mathbb{D}$ and no literals can be deleted from T_i without violating the inequality, moreover, $T_i \equiv f$ in $\mathcal{A}(T_i)$, thus T_i is a prime implicant of f. ■

The following lemma gives the conditions that the term in the full lattice PWA representation can be replaced.

Lemma 4: In the full lattice PWA representation (6), the term $T_i^{\mathrm{F}} = \min_{j \in I_{\geq,i}}\{\ell_j\}$ can be replaced by $T_i = \min_{j \in I_i}\{\ell_j\}$ without affecting the function value if T_i is an implicant of f and $I_i \subseteq I_{\geq,i}$.

Proof: After replacing T_i^{F} with T_i in (6), we have $\max\{T_1^{\mathrm{F}}, \ldots, T_{i-1}^{\mathrm{F}}, T_i, T_{i+1}^{\mathrm{F}}, \ldots, T_N^{\mathrm{F}}\} \equiv f$ in all base regions, hence the function value of f is not changed. ■

B. Necessary and sufficient conditions for irredundancy

The irredundancy of a lattice PWA expression can be checked through the following theorem.

Theorem 1: The lattice PWA representation (11) satisfies $f_{\mathrm{L}} = f$ in $\mathbb{D}$ is irredundant if and only if the following two conditions hold:

i) Each term $T_i = \min_{j \in I_i}\{\ell_j\}$ is a prime implicant of f.

ii) $\mathcal{A}(T_i) \not\subseteq \cup_{s=1, s \neq i}^{\tilde{N}} \mathcal{A}(T_s)$, $\forall i \in \{1, \ldots, \tilde{N}\}$.

Proof: First we prove *necessity*. If the lattice PWA expression f_{L} is irredundant, then no terms and literals can be deleted from f_{L} without affecting the function value of f_{L}.

Clearly condition i) must hold; otherwise if T_i is not an implicant of f,we can delete it without affecting the function value of f_{L}, or else if T_i is an implicant but not prime implicant, as a direct result of Lemma 4, we can delete some literal from T_i without affecting the function value.

Considering condition ii), if it is not satisfied, there is an $\hat{i}$ such that $\mathcal{A}(T_{\hat{i}}) \subseteq \cup_{s=1, s \neq \hat{i}}^{\tilde{N}} \mathcal{A}(T_s)$, then for each $k \in \mathcal{A}(T_{\hat{i}})$, there is some $i_k \in \{1, \ldots, \tilde{N}\}, i_k \neq \hat{i}$ such that $k \in \mathcal{A}(T_{i_k})$, i.e.,

$$T_{i_k}(x) \equiv f(x),\ \forall x \in \mathbb{D}_k. \tag{13}$$

Thus we have

$$\max_{s=1,\ldots,\hat{i}-1,\hat{i}+1,\ldots,\tilde{N}}\{T_s(x)\} = f(x),\ \forall x \in \mathbb{D},$$

hence $T_{\hat{i}}$ can be removed from f_{L} without affecting the function value, yielding contradiction.

Now we prove *sufficiency*. Condition i) implies that no literals can be deleted from T_i without affecting the function value.

We prove that condition ii) indicates that no prime implicant $T_i = \min_{j \in I_i}\{\ell_j\}$ can be deleted without affecting the function value of f_{L} in $\mathbb{D}$. Otherwise, if we delete $T_{\hat{i}}$ for some $\hat{i} \in \{1, \ldots, \tilde{N}\}$, according to condition ii), there is at least one index $k_{\hat{i}} \in \{1, \ldots, N\}$ satisfying $k_{\hat{i}} \in \mathcal{A}(T_{\hat{i}})$ and $k_{\hat{i}} \notin \cup_{s=1, s\neq \hat{i}}^{\tilde{N}} \mathcal{A}(T_s)$. Thus in $\mathrm{int}(\mathbb{D}_{k_{\hat{i}}})$, we have

$$\max_{s=1,\ldots,\hat{i}-1,\hat{i}+1,\ldots,\tilde{N}} \{T_s\} < f$$

the function value of f_{L} has been changed. Therefore, the two conditions ensure the irredundancy of f_{L}. ■

C. Removing redundant terms and literals

A corollary concerning removing redundant terms in a lattice PWA representation follows.

Corollary 1: In the lattice PWA representation (11), the term $T_i = \min\limits_{j\in I_i}\{\ell_j\}$ can be removed without affecting the function value f_{L} in $\mathbb{D}$ if and only if

$$\mathcal{A}(T_i) \subseteq \cup_{s=1,s\neq i}^{\tilde{N}} \mathcal{A}(T_s). \tag{14}$$

Proof: According to the proof of Theorem 1, condition ii) is a necessary and sufficient condition for the irredundancy of terms. Hence, T_i is redundant if and only if (14) is satisfied. ■

Next we explain how to remove redundant literals and derive prime implicants.

Theorem 2: Given a term $T_i^{\mathrm{F}} = \min\limits_{j\in I_{\geq,i}}\{\ell_j\}$ in the full lattice PWA representation (6). The term $T_i = \min\limits_{j\in I_i}\{\ell_j\}$ with $I_i \subsetneq I_{\geq,i}$ is an implicant of f if and only if $\forall t \in I_{\geq,i} \setminus I_i$, $\forall k \in \mathcal{A}(\ell_t)$, we have

$$T_i(x) \leq \ell_t(x), \forall x \in \mathbb{D}_k. \tag{15}$$

Proof: The proof can be divided into two parts, the first is necessity and the second is sufficiency.

(1) *Necessity.* As T_i is an implicant of f, we have $T_i \leq f$ in $\mathbb{D}$. If (15) does not hold, there would exist some index $\tilde{t} \in I_{\geq,i} \setminus I_i$, $\tilde{k} \in \mathcal{A}(\ell_t)$, and some $\tilde{x} \in \mathbb{D}_{\tilde{k}}$ such that $T_i(\tilde{x}) > \ell_t(\tilde{x}) = f(\tilde{x})$, which yields a contradiction.

(2) *Sufficiency.* Assuming that (15) holds for all $t \in I_{\geq,i} \setminus I_i$ and all $k \in \mathcal{A}(\ell_t)$, then $\mathrm{act}(i) \in I_i$. According to Definition 3, in order to prove that T_i is an implicant of f, two steps are needed, the first is to prove $T_i \leq f$ in $\mathbb{D}$ and the second is to prove that there exist some base regions in which $T_i \equiv f$.

Step 1: Now we prove that $T_i \leq f$ in $\mathbb{D}$, i.e., for all $k \in \{1, \ldots, N\}$, the following holds:

$$T_i(x) \leq \ell_{\mathrm{act}(k)}(x) = f(x), \forall x \in \mathbb{D}_k \tag{16}$$

The proof proceeds according to different *cases* of the index $\mathrm{act}(k)$: $\mathrm{act}(k) \in I_i$, $\mathrm{act}(k) \in I_{\geq,i} \setminus I_i$ and $\mathrm{act}(k) \notin I_{\geq,i}$.

Case 1: $\mathrm{act}(k) \in I_i$. In this case, (16) follows as $T_i \leq \ell_{\mathrm{act}(k)}$.

Case 2: $\mathrm{act}(k) \in I_{\geq,i} \setminus I_i$. In this case, (15) ensures the validity of (16).

Case 3: $\mathrm{act}(k) \notin I_{\geq,i}$. In this case, if (16) is invalid, we can find an index $\tilde{k}$ and an $x_{\mathrm{s}} \in \mathbb{D}_{\tilde{k}}$ satisfying $\ell_{\mathrm{act}(\tilde{k})}(x_{\mathrm{s}}) = f(x_{\mathrm{s}}) < T_i(x_{\mathrm{s}})$. As both $\ell_{\mathrm{act}(\tilde{k})}$ and T_i are continuous, such an x_{s} can be found in the interior of $\mathbb{D}_{\tilde{k}}$, i.e., $x_{\mathrm{s}} \in \mathrm{int}(\mathbb{D}_{\tilde{k}})$.

Since $\mathrm{act}(\tilde{k}) \notin I_{\geq,i}$, we can find an $x_{\mathrm{e}} \in \mathrm{int}(\mathbb{D}_i)$ satisfying $\ell_{\mathrm{act}(\tilde{k})}(x_{\mathrm{e}}) < \ell_{\mathrm{act}(i)}(x_{\mathrm{e}}) = f(x_{\mathrm{e}}) = T_i^{\mathrm{F}}(x_{\mathrm{e}}) = T_i(x_{\mathrm{e}})$.

Consider the line segment

$$[x_{\mathrm{s}},\ x_{\mathrm{e}}] = \{x \in \mathbb{D} | x = (1-\lambda)x_{\mathrm{s}} + \lambda x_{\mathrm{e}},\ 0 \leq \lambda \leq 1\}, \tag{17}$$

as $\mathbb{D}$ is convex, we conclude that f is continuous PWA when restricted to this line segment.

As $\ell_{\mathrm{act}(\tilde{k})}(x_{\mathrm{s}}) < T_i(x_{\mathrm{s}})$ and $\ell_{\mathrm{act}(\tilde{k})}(x_{\mathrm{e}}) < T_i(x_{\mathrm{e}})$, for all $j \in I_i$, $\forall x \in [x_{\mathrm{s}}, x_{\mathrm{e}}]$, we have $\ell_{\mathrm{act}(\tilde{k})}(x) < \ell_j(x)$. Hence for all $x \in [x_{\mathrm{s}}, x_{\mathrm{e}}]$, we have $\ell_{\mathrm{act}(\tilde{k})}(x) < \ell_{\mathrm{act}(i)}(x)$.

Since f is continuous when restricted to $[x_{\mathrm{s}}, x_{\mathrm{e}}]$, there must exist some point $x_1 \in [x_{\mathrm{s}}, x_{\mathrm{e}}]$ and an affine function ℓ_{i_1} with $i_1 \neq \mathrm{act}(\tilde{k})$, such that $f(x_1) = \ell_{\mathrm{act}(\tilde{k})}(x_1) = \ell_{i_1}(x_1)$, thus we have

$$\ell_{i_1}(x_1) = \ell_{\mathrm{act}(\tilde{k})}(x_1) < T_i(x_1). \tag{18}$$

According to (18), we have $x_1 \neq x_{\mathrm{e}}$. Define an index set $\mathcal{S}_1$ as

$$\mathcal{S}_1 = \{1, \ldots, M\} \setminus \left(I_{\geq,i} \cup \mathrm{act}(\tilde{k})\right). \tag{19}$$

Now we prove that $i_1 \in \mathcal{S}_1$.

As $\ell_{\mathrm{act}(\tilde{k})}(x) < T_i(x)$ for all $x \in [x_{\mathrm{s}}, x_{\mathrm{e}}]$, we have $i_1 \notin I_i$. If $i_1 \in I_{\geq,i} \setminus I_i$, then (15) does not hold for x_1 due to (18). Thus $i_1 \notin I_{\geq,i}$. Moreover, it is clear that $i_1 \neq \mathrm{act}(\tilde{k})$ and therefore, $i_1 \in \mathcal{S}_1$.

Since $i_1 \notin I_{\geq,i}$, we also have $\ell_{i_1}(x_{\mathrm{e}}) < \ell_{\mathrm{act}(i)}(x_{\mathrm{e}})$. So for all $x \in [x_1, x_{\mathrm{e}}]$, we have $\ell_{i_1}(x) < \ell_{\mathrm{act}(i)}(x)$, then there would exist a point x_2 and an affine function ℓ_{i_2} with $i_2 \neq i_1$ and $i_2 \neq \mathrm{act}(\tilde{k})$ such that $f(x_2) = \ell_{i_1}(x_2) = \ell_{i_2}(x_2)$. Clearly $x_2 \neq x_{\mathrm{e}}$.

Let the index set $\mathcal{S}_2$ be defined as

$$\mathcal{S}_2 = \mathcal{S}_1 \setminus \{i_1\},$$

then similar to the proof concerning i_1, we have $i_2 \in \mathcal{S}_2$.

Repeating the above procedure if necessary, and after l $(l < M)$ iterations we can reach an empty index set $\mathcal{S}_l$. According to the discussion for the previous iterations, we should have a point $x_l \in [x_{l-1}, x_{\mathrm{e}}]$, $x_l \neq x_{\mathrm{e}}$ and an index $i_l \in \mathcal{S}_l$ such that

$$\ell_{i_l}(x_l) = \ell_{i_{l-1}}(x_l),$$

which cannot be fulfilled as $\mathcal{S}_l$ is empty. Therefore, we have (16) for all $x \in \mathbb{D}$.

Step 2: Now we prove that there exists some base region in which $T_i \equiv f$, i.e., $\mathcal{A}(T_i) \neq \emptyset$. Considering $\mathbb{D}_k, k \in \mathcal{A}(T_i^{\mathrm{F}})$, according to (15), we have $\mathrm{act}(k) \in I_i$, thus $T_i \equiv T_i^{\mathrm{F}} \equiv f$ in $\mathbb{D}_k, \forall k \in \mathcal{A}(T_i^{\mathrm{F}})$.

Therefore T_i is an implicant of f. ■

Using Theorem 1 and 2, we can delete redundant terms or literals in a term T_i until further deletion is impossible, and the resulting expression is irredundant.

We have to point out that (14) is different from the row vector simplification lemma in [11], which states that if $I_r \subseteq I_k$, then $T_k = \min\limits_{j\in I_k}\{\ell_j\}$ can be removed without affecting the function value. Again looking into Example 1, we will explain that the row vector simplification lemma in [11] is only a sufficient condition for removing redundant terms.

Example 1 (Continued): Reconsidering Example 1, now we can use Theorem 2 to explain why ℓ_5 can be removed from T_1^{F} without affecting the function value. As $T_1(x) = \min\{\ell_1(x), \ell_3(x), \ell_4(x)\} \leq \ell_5(x), \forall x \in \mathbb{D}_7$ and $\mathcal{A}(\ell_5) = \{7\}$, T_1 is an implicant of f and can replace T_1^{F} without affecting the function value.

An interesting phenomenon is that the prime implicant $\tilde{T}_1 = \min\{\ell_1, \ell_3, \ell_5\}$ can also replace T_1^{F} as $\tilde{T}_1 \leq \ell_4$ in $\mathbb{D}_6$ and $\mathcal{A}(\ell_4) = \{6\}$. Hence, there may be more than one prime implicant resulting from a given term of the full lattice PWA representation.

Following gives a set of prime implicants for each term in (9),

$$\begin{aligned} &T_1 = \min_{j \in I_1 = \{1,3,4\}} \{\ell_j\},\ T_2 = \min_{j \in I_2 = \{2,3,5\}} \{\ell_j\}, \\ &T_3 = \min_{j \in I_3 = \{2,3,4\}} \{\ell_j\},\ T_4 = \min_{j \in I_4 = \{1,3,4\}} \{\ell_j\}, \\ &T_5 = \min_{j \in I_5 = \{1,3,4\}} \{\ell_j\},\ T_6 = \min_{j \in I_6 = \{1,3,4\}} \{\ell_j\}, \\ &T_7 = \min_{j \in I_7 = \{1,3,5\}} \{\ell_j\}. \end{aligned} \tag{20}$$

Then according to (14), if we search from 1 to N, the redundant terms $T_i, i = 1, 2, 4, 5, 6$ can be removed and we obtain an irredundant lattice PWA expression,

$$f_{\mathrm{L}} = \max\{\min\{\ell_2, \ell_3, \ell_4\}, \min\{\ell_1, \ell_3, \ell_5\}\}. \tag{21}$$

In fact, if we search from N to 1, then we can remove the redundant terms $T_i, i = 3, 4, 5, 6, 7$ and obtain the following irredundant lattice PWA expression,

$$f_{\mathrm{L}} = \max\{\min\{\ell_1, \ell_3, \ell_4\}, \min\{\ell_2, \ell_3, \ell_5\}\}. \tag{22}$$

From this example, we can conclude that the irredundant lattice PWA representation are not unique.

If we apply the row vector simplification lemma of [11], the following expression is obtained:

$$\begin{aligned} f_{[11]} = \max\{ &\min\{\ell_1, \ell_3, \ell_4\}, \min\{\ell_2, \ell_3, \ell_5\}, \\ &\min\{\ell_2, \ell_3, \ell_4\}, \min\{\ell_1, \ell_3, \ell_5\}\}, \end{aligned} \tag{23}$$

which clearly contains more terms than (22).

IV. Application in linear explicit MPC

Consider MPC of solving a constrained regulation problem for discrete-time linear time-invariant system at time step t:

$$\min_U \left\{ J(U, x_t) = x_{t+N_y}^T P x_{t+N_y} + \sum_{k=0}^{N_y-1} \left[x_{t+k}^T Q x_{t+k} \right. \right. \tag{24a}$$

$$\left. \left. + u_{t+k}^T R u_{t+k} \right] \right\}$$

$$s.t.\ y_{\min} \leq y_{t+k} \leq y_{\max}, k = 1, \ldots, N_y, \tag{24b}$$

$$u_{\min} \leq u_{t+k} \leq u_{\max}, k = 0, 1, \ldots, N_y - 1, \tag{24c}$$

$$x_{t+k+1} = A x_{t+k} + B u_{t+k}, k = 0, 1, \ldots, N_y - 1, \tag{24d}$$

$$y_{t+k} = C x_{t+k}, k = 1, \ldots, N_y, \tag{24e}$$

$$u_{t+k} = K x_{t+k}, k = N_u, \ldots, N_y - 1, \tag{24f}$$

in which the optimized variable $U = [u_t^T, \ldots, u_{t+N_y-1}^T]^T$; N_u and N_y are the control horizon and prediction horizon respectively, x_{t+k}, y_{t+k} denote the predicted state and output vector at time step $t + k$ using (24d). We assume $Q \succcurlyeq 0$, $P, R \succ 0$; K is the feedback gain of a stabilizing controller.

After solving the optimization problem (24), the optimal $U^* = [(u_t^*)^T, \ldots, (u_{t+N_y-1}^*)^T]^T$ is obtained, and only u_t^* is applied to the system. The optimization problem is subsequently reformulated and solved at the next time steps $t+1, t+2, \ldots$ by refreshing the given state vector x_t.

It is obvious that the online optimization has to be finished within the sample interval of the discrete-time linear time-invariant system, which may be hard to accomplish when the sample interval is small. Hence, explicit MPC has been proposed in [12] to calculate an explicit expression of u_t as a function of x_t. The following lemma gives the expression of the optimal solution and the MPT toolbox [14] can give the solution in the form of subregions and local affine functions defined on them.

Lemma 5: [12] The control law $u_t = f(x_t), f : \mathbb{D} \rightarrow \mathbb{R}^m$ defined by the optimization problem (24) is a continuous and PWA function of the form

$$f(x) = F^i x + g^i, \text{ if } A^i x \leq b^i,\ i = 1, \ldots, N, \tag{25}$$

where the polyhedral sets $\{A^i x \leq b^i\}$, $i = 1, \ldots, N$ form a partition of the set of states.

In [11], a lattice PWA representation is used to represent the resulting continuous PWA controller. In [11] the lattice PWA expression is also simplified to give a more compact expression. However, as pointed out in Section III above, the irredundancy of the simplification results in [11] cannot be guaranteed. Hence, we now give the irredundant lattice PWA representation to simplify the explicit MPC output.

A. Simulation Example

Consider the discrete-time double integrator example introduced in [17], in which the system dynamics can be written as

$$\begin{aligned} x_{k+1} &= \begin{bmatrix} 1 & T_{\mathrm{s}} \\ 0 & 1 \end{bmatrix} x_k + \begin{bmatrix} T_{\mathrm{s}}^2 \\ T_{\mathrm{s}} \end{bmatrix} u_k, \\ y_k &= \begin{bmatrix} 0 & 1 \end{bmatrix} x_k, \end{aligned} \tag{26}$$

where the sampling interval is $T_\mathrm{s} = 0.3$s. Consider the MPC problem (24) with $Q = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$, $R = 1$. In this example, we calculate P as the solution of the discrete-time algebraic Riccati equation and $K = (R + B^T P B)^{-1} B^T P A$, i.e., $P = \begin{bmatrix} 4.7674 & 2.6941 \\ 2.6941 & 3.8531 \end{bmatrix}$ and $K = [0.8082\ 1.1559]$. The system constraints are $-0.5 \le y_k \le 0.5$ and $-1 \le u_k \le 1$. The domain of x is $[-2.8,\ 2.8] \times [-0.8,\ 0.8]$.

Assume $N_y = N_u = 10$. First we use the MPT toolbox to compute the optimal output u_t as a function of x_t. It is a continuous PWA function with 137 subregions.

In each of the 137 subregions, there is a corresponding local affine function. Among all the affine functions, there are only 27 unique ones; hence, several subregions may share a same local affine function. After removing redundant terms and literals, only 11 terms left and the number of parameters reduces from 2073 to 133. Hence, the original solution calculated by the MPT toolbox can be represented by a much more compact irredundant lattice PWA expression.

For $N_y = 2, 6, 10, 14, 18, 20$ ($N_u = N_y$), Table I compares the performance of the MPC output, the procedure in [11] and the irredundant lattice PWA expression. In the table, N_m, $N_{[11]}$ and N_L are the number of parameters of the MPT output, of the output in [11], and of the irredundant lattice PWA representation. Also listed in Table I are the elapsed time for evaluating the optimal solution through the MPT toolbox, the representation in [11], and the irredundant lattice PWA representation, denoted by τ_m, $\tau_{[11]}$ and τ_L, respectively.

TABLE I

COMPARISON OF PERFORMANCES OF THREE REPRESENTATIONS

N_y	N_m	$N_{[11]}$	N_L	τ_m(ms)	$\tau_{[11]}$(ms)	τ_L(ms)
2	477	87	80	1.6	0.137	0.119
6	1677	176	148	2.5	0.325	0.232
10	2073	172	133	2.7	0.327	0.199
14	2385	184	133	2.8	0.366	0.201
18	2349	184	133	4.0	0.458	0.241
20	2349	184	133	2.9	0.367	0.202

From the table, we can see that the number of parameters used to describe a continuous PWA function is reduced significantly when using the irredundant lattice PWA representation, and this reduction is more evident when N_y is large. Besides, the number of parameters for the irredundant representation is less than that in [11], which means that the lattice PWA expression in [11] is redundant.

Moreover, for this example, the time needed for evaluating the optimal solution through the irredundant lattice PWA representation is less than that in MPT toolbox. It is also noticed that the evaluating time for the irredundant lattice PWA representation and the lattice representation in [11] are close, which is due to that the difference of the number of parameters of the two representations is small in this example.

V. CONCLUSIONS

In this paper, we have proposed the irredundant lattice PWA representation, in which no terms or literals can be deleted. It greatly facilitates the application of the lattice PWA representation in representing continuous PWA functions. Necessary and sufficient conditions for irredundancy have been proposed. Besides, we have also provided necessary and sufficient conditions for removing redundant terms and literals. Based on these conditions, an algorithm for obtaining an irredundant lattice PWA expression is given. A worked example is provided to express the explicit solution of a linear MPC problem, the results of which show that the number of parameters is reduced significantly.

REFERENCES

[1] R. Wilkinson, "A method of generating functions of several variables using analog diode logic," *IEEE Transactions on Electronic Computers*, vol. EC-12, no. 2, pp. 112–129, 1963.

[2] J. M. Tarela, E. Alonso, and M. V. Martinez, "A representation method for PWL functions oriented to parallel processing," *Mathematical and Computer Modelling*, vol. 13, no. 10, pp. 75–83, 1990.

[3] J. Gunawardena, "Min-max functions," *Discrete Event Dynamic Systems: Theory and Applications*, vol. 4, no. 4, pp. 377–407, 1994.

[4] B. De Schutter and T. J. J. van den Boom, "MPC for continuous piecewise-affine systems," *Systems & Control Letters*, vol. 52, no. 3, pp. 179–192, 2004.

[5] S. Ovchinnikov, "Max-min representation of piecewise linear functions," *Contributions to Algebra and Geometry*, vol. 43, no. 1, pp. 297–302, 2002.

[6] L. Breiman, "Hinging hyperplanes for regression, classification and function approximation," *IEEE Transactions on Information Theory*, vol. 39, no. 3, pp. 999–1013, 1993.

[7] P. Julián, A. Desages, and O. Agamennoni, "High-level canonical piecewise linear representation using a simplicial partition," *IEEE Transactions on Circuits and Systems-I: Fundamental Theory and Applications*, vol. 46, no. 4, pp. 463–480, 1999.

[8] J. Xu, X. Huang, and S. Wang, "Adaptive hinging hyperplanes and its applications in dynamic system identification," *Automatica*, vol. 45, no. 10, pp. 2325–2332, 2009.

[9] S. Wang and X. Sun, "Generalization of hinging hyperplanes," *IEEE Transactions on Information Theory*, vol. 12, no. 51, pp. 4425–4431, 2005.

[10] S. Wang, X. Huang, and K. M. Junaid, "Configuration of continuous piecewise-linear neural networks," *IEEE Transactions on Neural Networks*, vol. 19, no. 8, pp. 1431–1445, 2008.

[11] C. Wen, X. Ma, and B. E. Ydstie, "Analytical expression of explicit MPC solution via lattice piecewise-affine function," *Automatica*, vol. 45, no. 4, pp. 910–917, 2009.

[12] A. Bemporad, M. Morari, V. Dua, and E. N. Pistikopoulos, "The explicit linear quadratic regulator for constrained systems," *Automatica*, vol. 38, no. 1, pp. 3–20, 2002.

[13] A. Bemporad, F. Borrelli, and M. Morari, "Model predictive control based on linear programming–the explicit solution," *IEEE Transactions on Automatic Control*, vol. 47, no. 12, pp. 1974–1985, 2002.

[14] M. Herceg, M. Kvasnica, C. Jones, and M. Morari, "Multi-Parametric Toolbox 3.0," in *Proceedings of the 2013 European Control Conference*, Zürich, Switzerland, July 17–19 2013, pp. 502–510, http://control.ee.ethz.ch/~ mpt.

[15] L. O. Chua and A. C. Deng, "Canonical piecewise-linear representation," *IEEE Transactions on Circuits and Systems*, vol. 35, no. 1, pp. 101–111, 1988.

[16] J. M. Tarela and M. V. Martínez, "Region configurations for realizability of lattice piecewise-linear models," *Mathematical and Computer Modeling*, vol. 30, no. 11, pp. 17–27, 1999.

[17] T. A. Johansen and A. Grancharova, "Approximate explicit constrained linear model predictive control via orthogonal search tree," *IEEE Transactions on Automatic Control*, vol. 48, no. 5, pp. 810–815, 2003.