Delft University of Technology

Delft Center for Systems and Control

Technical report 15-017

Irredundant lattice representations of continuous piecewise affine functions*

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

If you want to cite this report, please use the following reference instead:
J. Xu, T.J.J. van den Boom, B. De Schutter, and S. Wang, "Irredundant lattice representations of continuous piecewise affine functions," *Automatica*, vol. 70, pp. 109–120, Aug. 2016.

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

*This report can also be downloaded via http://pub.deschutter.info/abs/15_017.html

Irredundant lattice representations of continuous piecewise affine functions

Jun Xu [a,b], Ton J.J. van den Boom [b], Bart De Schutter [b], Shuning Wang [c,d]

[a] *Department of Automation, China University of Petroleum (Beijing), China*

[b] *Delft Center for Systems and Control, Delft University of Technology, The Netherlands*

[c] *Department of Automation, Tsinghua University, Beijing, China*

[d] *Tsinghua National Laboratory for Information Science and Technology (TNList), China*

Abstract

In this paper, we revisit the full lattice representation of continuous piecewise affine (PWA) functions and give a formal proof of its representation ability. Based on this, we derive the irredundant lattice PWA representations through removal of redundant terms and literals. Necessary and sufficient conditions for irredundancy are proposed. Besides, we explain how to remove terms and literals in order to ensure irredundancy. An algorithm is given to obtain an irredundant lattice PWA representation. In the worked examples, the irredundant lattice PWA representations are used to express the optimal solution of explicit model predictive control problems, and the results turn out to be much more compact than those given by a state-of-the-art algorithm.

Key words: piecewise affine function; irredundant representation; lattice representation.

1 Introduction

A continuous piecewise affine (PWA) function is a nonlinear function with affine components defined on polyhedral subregions. It is demonstrated in (Wilkinson, 1963) that any continuous PWA function can be expressed by a min-max or max-min composition of its affine components,

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

or

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

in which ℓ_j is an affine function, N_1 and N_2 are integers, and $\bar{I}_i$ and $\tilde{I}_i$ are index sets. In (Tarela and Martinez, 1999), formal proofs are given demonstrating that any continuous PWA function can be described by (1) and (2), which are then called lattice PWA representations. They also appeared in (Gunawardena, 1994) and (De Schutter and van den Boom, 2004). We call (1) the conjunctive form and (2) disjunctive form. In (Bartels et al., 1995; Ovchinnikov, 2002) and (Ovchinnikov, 2010), the representation ability of (1) and (2) is also proved.

Among all these papers (Bartels et al., 1995; De Schutter and van den Boom, 2004; Gunawardena, 1994; Ovchinnikov, 2002, 2010; Tarela and Martinez, 1999; Wilkinson, 1963), only (Tarela and Martinez, 1999; Wilkinson, 1963) give methods for determining the parameters N_1, $\bar{I}_i$ in (1) and N_2, $\tilde{I}_i$ in (2). However, (Wilkinson, 1963) only illustrates how to determine the parameters for a 1-dimensional example and does not provide a formal proof. Moreover, it is demonstrated in (Ovchinnikov, 2010) that an important assumption is not stated in the proofs in (Tarela and Martinez, 1999), while without that assumption the conclusions do not hold. In this paper, we mainly focus on the disjunctive lattice PWA representation (2), and give a proof concerning the representation ability as well as the determination of the

⋆ This research was partially supported by China Scholarship Council (CSC), National Natural Science Foundation of China (61104218; 61134012; 61473165), the National Basic Research Program of China (973 program: 2012CB720505), the Science Foundation of China University of Petroleum (YJRC-2013-12). This paper was not presented at any IFAC meeting.

Email addresses: xujunqgy@gmail.com (Jun Xu), a.j.j.vandenBoom@tudelft.nl (Ton J.J. van den Boom), b.deschutter@tudelft.nl (Bart De Schutter), swang@mail.tsinghua.edu.cn (Shuning Wang).

parameters. The results can be easily extended to the conjunctive case due to duality.

There are also other methods for representing PWA functions (Breiman, 1993; Julián et al., 1999; Wang et al., 2008; Wang and Sun, 2005; Xu et al., 2009). The methods of (Breiman, 1993) can only represent continuous PWA functions in 1 dimension. The representations of (Julián et al., 1999; Xu et al., 2009) can only represent continuous PWA functions of which the domain is partitioned into simplices or the union of simplices. Although the representations proposed in (Wang et al., 2008; Wang and Sun, 2005) can represent any continuous PWA function, the parameters in the expression of (Wang and Sun, 2005) are hard to derive and the number of parameters in the expression of (Wang et al., 2008) is large. Conversely, we will show in Section 2 that the integer N_2 and the index set $\tilde{I}_i$ in (2) are not hard to derive.

Lattice PWA representations have been used to express the solution of explicit model predictive control (MPC) problems in (Wen et al., 2009). In MPC, the control action is obtained by solving a finite-horizon open-loop optimal control problem at each sampling instant. At the next time step, a new optimal control problem based on new measurements of the state is solved over a shifted horizon. The optimization relies on a prediction model for predicting future outputs of the system, can take into account input and output constraints, and minimizes a performance criterion (Bemporad et al., 2002a). When the constraints are affine, a continuous PWA control law arises if the performance criterion in the optimization problem of MPC is convex quadratic or polyhedral. Then, 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 corresponding continuous PWA optimal solution can be computed using multi-parametric linear or quadratic programming through e.g. the MPT Toolbox (Herceg et al., 2013a) and stored as a collection of local affine functions and subregions. For online evaluation, many papers are dedicated to solving a point location problem, i.e., determining the subregion the present state is located in, and then finding the corresponding local affine function (Christophersen et al., 2007; Herceg et al., 2013b; Tøndel et al., 2003b). The online search complexity is logarithmic in the number of subregions (Herceg et al., 2013b; Tøndel et al., 2003b) or linear in the number of subregions (Christophersen et al., 2007). For this kind of methods, the online search can be accelerated through storing additional information apart from the polyhedral partition, such as search tree and adjacency information.

On the other hand, some papers reduce the offline storage complexity by avoiding the storage of the polyhedral information (Baotic et al., 2008; Jones et al., 2006). For the case of linear cost function, both methods store only the optimal value function; the online evaluation complexity for (Baotic et al., 2008) is linear in the number of subregions while for the method of (Jones et al., 2006) it is logarithmic. However, for the quadratic cost case, the method in (Jones et al., 2006) is not applicable and the procedure of (Baotic et al., 2008) has to store the information of the descriptor function as well as the ordering of local affine functions in neighboring polyhedra. Hence, it is of great value to find a method to reduce offline storage complexity for both the linear and the quadratic case.

For a continuous PWA controller derived in the linear or the quadratic case, through determining the parameters of (1), the lattice PWA function is used to represent the controller in (Wen et al., 2009). For online evaluation, the current state is then directly substituted into expression (1) and the optimal solution results. By removing redundant parameters in the lattice PWA representations, both the storage requirements and the online complexity can be reduced. However, the simplification lemmas in (Wen et al., 2009) have limitations and the result cannot guaranteed to be irredundant. Hence, in the current paper, we aim to give irredundant lattice PWA representations.

The paper is organized as follows. The next section introduces the full lattice PWA representation, and gives a proof of its representation ability. The *irredundant* lattice PWA representations are derived in Section 3, including necessary and sufficient conditions for irredundancy and the algorithm for obtaining an irredundant lattice PWA representation. The offline preprocessing and online evaluation complexity of the irredundant lattice PWA representations is also analyzed. In Section 4, two worked examples are given, in which the irredundant lattice PWA representations are applied to express the solutions of explicit MPC problems. Finally, the paper ends with conclusions in Section 5.

2 Full lattice PWA representation

Definition 1 *(Chua and Deng, 1988) A function $f : \mathbb{D} \to \mathbb{R}$, where $\mathbb{D} \subseteq \mathbb{R}^n$ is convex, is said to be continuous PWA if it is continuous on the domain $\mathbb{D}$ and the following conditions are satisfied:*

(1) The domain space $\mathbb{D}$ can be divided into a finite number of nonempty convex polyhedra, i.e., $\mathbb{D} = \cup_{i=1}^{\hat{N}} \Omega_i$, $\Omega_i \neq \emptyset$, the polyhedra are closed and have non-overlapping interiors, $\mathrm{int}(\Omega_i) \cap \mathrm{int}(\Omega_j) = \emptyset$, $\forall i, j \in \{1, \ldots, \hat{N}\}, i \neq j$. These polyhedra are also called subregions. The boundaries of the polyhedra are $(n-1)$-dimensional hyperplanes.

(2) In each subregion Ω_i, f equals a local affine function $\ell_{\mathrm{loc}(i)}$,

$$f(x) = \ell_{\mathrm{loc}(i)}(x),\ \forall x \in \Omega_i.$$

It is important to note that in Definition 1 some local affine function may appear in different subregions, i.e., $\ell_{\mathrm{loc}(i_1)} = \cdots = \ell_{\mathrm{loc}(i_s)}$ for different $i_1, \ldots, i_s \in \{1, \ldots, \hat{N}\}$. We collect all the local affine functions and select those distinct ones, labeling them as $\ell_1, \ldots, \ell_M$. So $\mathrm{loc}(i) \in \{1, \ldots, M\}$ and no two affine functions ℓ_i and ℓ_j with $i, j \in \{1, \ldots, M\}, i \neq j$, are identical. Therefore, there can be more subregions than distinct affine functions.

We further partition each subregion Ω_i $(i = 1, \ldots, \hat{N})$ into so called base regions $\mathbb{D}_{i,t}$ with $t = 1, \ldots, m_i$, to make sure that no other affine function intersects with $\ell_{\mathrm{loc}(i)}$ at some point in the interior of $\mathbb{D}_{i,t}$, i.e.,

$$\{x | \ell_j(x) = \ell_{\mathrm{loc}(i)}(x), j \neq \mathrm{loc}(i)\} \cap \mathrm{int}(\mathbb{D}_{i,t}) = \emptyset. \quad (3)$$

The following lemma defines the partition.

Lemma 1 *For any $i \in \{1, \ldots, \hat{N}\}$, there is a partition of the subregion Ω_i*

$$\Omega_i = \cup_{t=1}^{m_i} \mathbb{D}_{i,t} \quad (4)$$

such that the following holds,

(1) The set $\mathrm{int}(\mathbb{D}_{i,t})$ is nonempty.
(2) For each $\mathbb{D}_{i,t}$, we have

$$I_{\geq,i,t} \cup I_{\leq,i,t} = \{1, \ldots, M\}, \quad (5)$$

in which $I_{\geq,i,t} = \{j | \ell_j(x) \geq \ell_{\mathrm{loc}(i)}(x), \forall x \in \mathbb{D}_{i,t}\}$ and $I_{\leq,i,t} = \{j | \ell_j(x) \leq \ell_{\mathrm{loc}(i)}(x), \forall x \in \mathbb{D}_{i,t}\}$.
(3) For all $i, j \in \{1, \ldots, \hat{N}\}, \bar{t} \in \{1, \ldots, m_i\}, \hat{t} \in \{1, \ldots, m_j\}$, $\bar{t} \neq \hat{t}$ or $i \neq j$, the following holds,

$$\mathrm{int}(\mathbb{D}_{i,\bar{t}}) \cap \mathrm{int}(\mathbb{D}_{j,\hat{t}}) = \emptyset. \quad (6)$$

The proof of Lemma 1 as well as the time complexity of the partition process is given in Appendix A.

From (5), it follows that in the base region $\mathbb{D}_{i,t}$ (3) is satisfied.

After the partition, there are base regions $\mathbb{D}_{1,1}, \ldots, \mathbb{D}_{1,m_1}, \ldots, \mathbb{D}_{\hat{N},1}, \ldots, \mathbb{D}_{\hat{N},m_{\hat{N}}}$. We renumber them as $\mathbb{D}_1, \mathbb{D}_2, \ldots, \mathbb{D}_N$, in which $N = m_1 + \cdots + m_{\hat{N}}$.

We denote $\ell_{\mathrm{act}(i)}(x)$ as the active affine function in $\mathbb{D}_i$, which is given by

$$\ell_{\mathrm{act}(i)} = \ell_{\mathrm{loc}(j)}, \text{if } \mathbb{D}_i \subseteq \Omega_j. \quad (7)$$

Then we define the index sets

$$I_{\geq,i} = \{j | \ell_j(x) \geq \ell_{\mathrm{act}(i)}(x), \forall x \in \mathbb{D}_i\}$$
$$I_{\leq,i} = \{j | \ell_j(x) \leq \ell_{\mathrm{act}(i)}(x), \forall x \in \mathbb{D}_i\}.$$

We also introduce an index set $\mathcal{A}(\ell_i)$ such that for each index $k \in \mathcal{A}(\ell_i)$, ℓ_i is the active affine function in $\mathbb{D}_k$, i.e., $f(x) = \ell_i(x), \forall x \in \mathbb{D}_k$. Clearly, $i \in \mathcal{A}(\ell_{\mathrm{act}(i)})$.

In the base region $\mathbb{D}_i$, for an affine function ℓ_j with $j \neq \mathrm{act}(i)$, either $j \in I_{\geq,i}$ or $j \in I_{\leq,i}$. Then we have

$$\ell_j(x) > \ell_{\mathrm{act}(i)}(x), \forall x \in \mathrm{int}(\mathbb{D}_i), \forall j \in I_{\geq,i}, \quad (8)$$

and

$$\ell_j(x) < \ell_{\mathrm{act}(i)}(x), \forall x \in \mathrm{int}(\mathbb{D}_i), \forall j \in I_{\leq,i}. \quad (9)$$

For a 1-dimensional continuous PWA function f, we have the following conclusion.

Proposition 1 *Let f be a 1-dimensional continuous PWA function as defined in Definition 1, i.e., $n = 1$, then $\forall i, k \in \{1, \ldots, N\}$, we have*

$$\min_{j \in I_{\geq,k}} \{\ell_j(x)\} \leq \min_{j \in I_{\geq,i}} \{\ell_j(x)\}, \forall x \in \mathbb{D}_i. \quad (10)$$

The proof of Proposition 1 can be found in Appendix B.

Based on Proposition 1, we propose the full lattice PWA representation for an n-dimensional continuous PWA function.

Theorem 1 *Let f be a continuous PWA function as defined in Definition 1. Then f can be represented as*

$$f(x) = \max_{i=1,\ldots,N} \{\min_{j \in I_{\geq,i}} \{\ell_j(x)\}\},\ \forall x \in \mathbb{D}, \quad (11)$$

and (11) is called full lattice PWA representation.

PROOF. For all $i, k \in \{1, \ldots, N\}$, if we can prove that

$$\min_{j \in I_{\geq,k}} \{\ell_j(x)\} \leq \min_{j \in I_{\geq,i}} \{\ell_j(x)\},\ \ \forall x \in \mathbb{D}_i \quad (12)$$

then we have $\max_{k \in \{1,\ldots,N\}} \{\min_{j \in I_{\geq,k}} \{\ell_j(x)\}\} = \ell_{\mathrm{act}(i)}(x), \forall x \in \mathbb{D}_i$, and then the validity of (11) follows.

Randomly choose $\bar{i}, \bar{k} \in \{1, \ldots, N\}$ and an $x_0 \in \mathrm{int}(\mathbb{D}_{\bar{i}})$. Now, we will show (12) is valid, i.e.,

$$\min_{j \in I_{\geq,\bar{k}}} \{\ell_j(x_0)\} \leq \min_{j \in I_{\geq,\bar{i}}} \{\ell_j(x_0)\}. \quad (13)$$

In order to prove (13), we randomly choose an $x_1 \in \mathrm{int}(\mathbb{D}_{\bar{k}})$, and consider the line segment between x_0 and x_1,

$$\mathcal{L}(x_0, x_1) = \{x | \lambda x_0 + (1-\lambda)x_1, 0 \le \lambda \le 1\}. \quad (14)$$

As $\mathbb{D}$ is convex, the line segment $\mathcal{L}(x_0, x_1) \subseteq \mathbb{D}$.

Clearly, f is continuous when restricted to the line segment $\mathcal{L}(x_0, x_1)$. Define the line segments

$$\mathcal{B}_i = \mathbb{D}_i \cap \mathcal{L}(x_0, x_1), i = 1, \ldots, N. \quad (15)$$

Then if $\mathrm{int}(\mathcal{B}_i)$ is nonempty, we have

$$f(x) = \ell_{\mathrm{act}(i)}(x), \forall x \in \mathcal{B}_i.$$

Therefore, f is continuous PWA when restricted to the line segment $\mathcal{L}(x_0, x_1)$.

Denote the set of indices of affine functions appearing in $\mathcal{L}(x_0, x_1)$ as

$$\mathcal{N}_{\mathcal{B}} = \{\mathrm{act}(k) | k \in \{1, \ldots, N\} \text{ and } \mathrm{int}(\mathcal{B}_k) \neq \emptyset\}.$$

As $x_0 \in \mathrm{int}(\mathbb{D}_{\bar{i}})$ and $x_1 \in \mathrm{int}(\mathbb{D}_{\bar{k}})$, the sets $\mathrm{int}(\mathcal{B}_{\bar{i}})$ and $\mathrm{int}(\mathcal{B}_{\bar{k}})$ are nonempty. Hence $\bar{i}, \bar{k} \in \mathcal{N}_{\mathcal{B}}$.

For the line segment $\mathcal{B}_i$ with nonempty interior, define the index set

$$S_{\ge,i} = \{j \in \mathcal{N}_{\mathcal{B}} | \ell_j(x) \ge \ell_{\mathrm{act}(i)}(x), \forall x \in \mathcal{B}_i\}.$$

According to (8) and (9), for any $j \in S_{\ge,i}$, we have $j \in I_{\ge,i}$, i.e., $S_{\ge,i} \subseteq I_{\ge,i}$.

Since both $\mathrm{int}(\mathcal{B}_{\bar{i}})$ and $\mathrm{int}(\mathcal{B}_{\bar{k}})$ are nonempty, according to Proposition 1 we have

$$\min_{j \in S_{\ge,\bar{k}}} \{\ell_j(x)\} \le \min_{j \in S_{\ge,\bar{i}}} \{\ell_j(x)\}, \forall x \in \mathcal{B}_{\bar{i}}.$$

As $x_0 \in \mathcal{B}_{\bar{i}}$ and $S_{\ge,\bar{k}} \subseteq I_{\ge,\bar{k}}$, we have

$$\min_{j \in I_{\ge,\bar{k}}} \{\ell_j(x_0)\} \le \min_{j \in S_{\ge,\bar{k}}} \{\ell_j(x_0)\} \le \min_{j \in S_{\ge,\bar{i}}} \{\ell_j(x_0)\}$$
$$= \ell_{\mathrm{act}(\bar{i})}(x_0) = \min_{j \in I_{\ge,\bar{i}}} \{\ell_j(x_0)\}.$$

As $\bar{i}, \bar{k}, x_0$ are arbitrarily chosen, and both sides of (13) are continuous, we have (12).

Therefore, a continuous PWA function defined on a convex set can be expressed as the full lattice PWA representation (11).

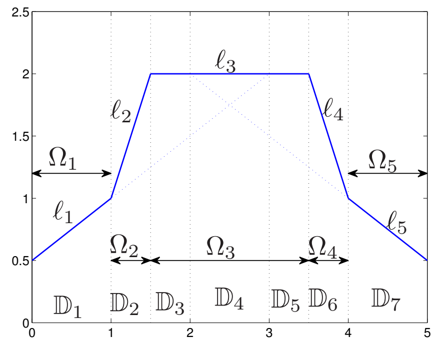


Fig. 1. A 1-dimensional example of a continuous PWA function.

In the full lattice PWA representation (11), two binary operations "min" and "max" are present. They are similar to the Boolean AND and OR of Boolean algebra. Analogously, we call "$\min_{j \in I_{\ge,i}} \{\ell_j\}$" 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 ℓ_j, $j \in I_{\ge,i}$ are called literals.

We give a simple 1-dimensional example to illustrate Lemma 1 and Theorem 1.

Example 1 *Consider the following 1-dimensional continuous PWA function defined on* $[0, 5]$*:*

$$f(x) = \begin{cases} \ell_1(x) = 0.5x + 0.5, & x \in [0,\ 1], \\ \ell_2(x) = 2x - 1, & x \in [1,\ 1.5], \\ \ell_3(x) = 2, & x \in [1.5,\ 3.5], \\ \ell_4(x) = -2x + 9, & x \in [3.5,\ 4], \\ \ell_5(x) = -0.5x + 3, & x \in [4,\ 5]. \end{cases}$$

Fig. 1 gives the plot of f*.*

It can be seen from Fig. 1 that there are 5 distinct affine functions, 5 subregions and 7 base regions, of which $\Omega_1 = \mathbb{D}_1$, $\Omega_2 = \mathbb{D}_2$, $\Omega_3 = \mathbb{D}_3 \cup \mathbb{D}_4 \cup \mathbb{D}_5$ *and* $\Omega_4 = \mathbb{D}_6, \Omega_5 = \mathbb{D}_7$*. According to Theorem 1, for all* $x \in [0, 5]$*, we have*

$$f(x) = \max_{i=1,\ldots,7} \{\min_{j \in I_{\ge,i}} \{\ell_j(x)\}\}, \quad (16)$$

with $I_{\ge,1} = \{1, 3, 4, 5\}$, $I_{\ge,2} = \{2, 3, 4, 5\}, I_{\ge,3} = \{2, 3, 4, 5\}$, $I_{\ge,4} = \{2, 3, 4\}$, $I_{\ge,5} = \{1, 2, 3, 4\}$, $I_{\ge,6} = \{1, 2, 3, 4\}$*, and* $I_{\ge,7} = \{1, 2, 3, 5\}$.

We have 7 terms, which are $T_i^{\mathrm{F}} = \min_{j \in I_{\ge,i}} \{\ell_j\}$*. It is obvious that* T_2^{F} *and* T_3^{F} *are the same, and one of them can be removed from (16) without affecting the function value of* f*. Besides, a more surprising fact is that* ℓ_3 *and* ℓ_4

in T_1^{F} can also be removed. Therefore, the lattice PWA representation (16) is redundant.

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

3 Irredundant lattice PWA representations

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

Definition 2 *A lattice PWA representation*

$$f_{\mathrm{L}} = \max_{i=1,\ldots,\tilde{N}} \{T_i\} = \max_{i=1,\ldots,\tilde{N}} \{\min_{j\in I_i}\{\ell_j\}\} \qquad (17)$$

with $\tilde{N} \leq N$ is irredundant, if no term $T_i = \min\limits_{j\in I_i}\{\ell_j\}$, and no literal ℓ_j, with $i \in \{1,\ldots,\tilde{N}\}$ and $j \in I_i$, can be removed from (17) without affecting the function value of f_{L}.

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

3.1 Implicants and prime implicants

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

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

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

We now describe the implicants and prime implicants in the context of the lattice PWA representations.

Lemma 2 *Every term $T_i^{\mathrm{F}} = \min\limits_{j\in I_{\geq,i}}\{\ell_j\}$ in the full lattice PWA representation (11) is an implicant of f. Moreover, there exists at least one prime implicant $T_i = \min\limits_{j\in I_i}\{\ell_j\}$ of f with $I_i \subseteq I_{\geq,i}$.*

The proof of Lemma 2 can be found in (Xu et al., 2014).

For an implicant, we define the base regions it covers.

Definition 4 *We say the implicant T_i* covers *the base region $\mathbb{D}_k$, if $T_i \equiv f$ in $\mathbb{D}_k$. The indices of all base regions T_i covers constitute an index set $\mathcal{C}(T_i)$.*

Following gives a lemma concerning how to identify $\mathcal{C}(T_i)$.

Lemma 3 *Given an implicant $T_i = \min\limits_{j\in I_i}\{\ell_j\}$, it covers the base region $\mathbb{D}_k$, i.e., $k \in \mathcal{C}(T_i)$ if and only if $I_i \subseteq I_{\geq,k}$.*

PROOF. Necessity. If the implicant T_i covers the base region $\mathbb{D}_k$, i.e., $T_i \equiv f$ in $\mathbb{D}_k$, we prove $I_i \subseteq I_{\geq,k}$. Otherwise, there would exist some $\upsilon \in I_i$ with $\upsilon \notin I_{\geq,k}$. According to (9), then we have

$$T_i(x) \leq \ell_\upsilon(x) < f(x),\ \forall x \in \mathrm{int}(\mathbb{D}_k),$$

which contradicts that $k \in \mathcal{C}(T_i)$.

Sufficiency. If $I_i \subseteq I_{\geq,k}$, as T_i is an implicant, we have

$$f(x) = T_k^{\mathrm{F}}(x) \leq T_i(x) \leq f(x), \forall x \in \mathbb{D}_k.$$

Thus, $T_i \equiv f$ in $\mathbb{D}_k$, i.e., $k \in \mathcal{C}(T_i)$.

For an implicant T_i, if $x \in \mathrm{int}(\mathbb{D}_k)$ with $k \notin \mathcal{C}(T_i)$, from the above proof, we have $T_i(x) < f(x)$.

We use Example 1 again to illustrate Lemmas 2 and 3.

Example 1 (Continued): According to Lemma 2, the terms T_i^{F} with $i \in \{1,\ldots,7\}$ are implicants of f defined in (16). Take T_2^{F} for example. Fig. 2(a) shows the plot of T_2^{F}. It is shown in Fig. 2(a) that $T_2^{\mathrm{F}} \leq f$ in $\mathbb{D}$. Besides, $T_2^{\mathrm{F}} = f$ in the base regions $\mathbb{D}_2$ and $\mathbb{D}_3$. Therefore, T_2^{F} is an implicant of f.

Then Fig. 2(b) shows the plot of $T_2 = \min\limits_{j\in I_2}\{\ell_j\}$ with $I_2 = \{2,3,4\} \subseteq I_{\geq,2} = \{2,3,4,5\}$. It is clear from Fig. 2(b) that $\min\{\ell_2,\ell_3,\ell_4\}$ is an implicant of f and that there are no $I_r \subsetneq I_2$ such that $\min\limits_{j\in I_r}\{\ell_j\}$ is also an implicant of f. Hence T_2 is a prime implicant of f.

Considering the prime implicant T_2, we can see from Fig. 2(b) that T_2 covers the base regions $\mathbb{D}_j, j = 2,3,4,5,6$. According to Lemma 3, the following holds:

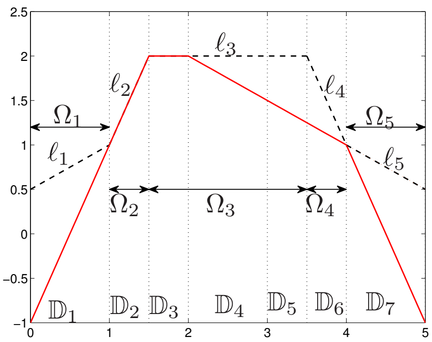
$$I_2 \subseteq I_{\geq,j}, j = 2,3,4,5,6. \qquad (18)$$

As $I_2 = \{2,3,4\}$, $I_{\geq,2} = I_{\geq,3} = \{2,3,4,5\}$, $I_{\geq,4} = \{2,3,4\}$, $I_{\geq,5} = I_{\geq,6} = \{1,2,3,4\}$, (18) is clearly true.

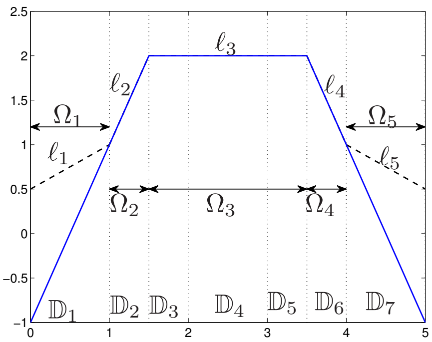
3.2 Necessary and sufficient conditions for irredundancy

The irredundancy of a lattice PWA representation can be verified through the following theorem.

Theorem 2 *The lattice PWA representation (17) satisfying $f_{\mathrm{L}} = f$ in $\mathbb{D}$ is irredundant if and only if the following two conditions hold:*



(a) Plot of $T_2^{\mathrm{F}} = \min\{\ell_2, \ell_3, \ell_4, \ell_5\}$: solid line.



(b) Plot of $T_2 = \min\{\ell_2, \ell_3, \ell_4\}$: solid line.

Fig. 2. Plot of an implicant and prime implicant of Example 1. The dashed line is the plot of f.

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

ii) For each prime implicant, $\mathcal{C}(T_i) \nsubseteq \cup_{s=1,s\neq i}^{\tilde{N}} \mathcal{C}(T_s)$, $\forall i \in \{1, \ldots, \tilde{N}\}$.

PROOF. First we prove *necessity*. Suppose the lattice PWA representation f_{L} is irredundant.

Condition i) must hold; otherwise if there is some $T_{\hat{i}}$ that is not an implicant of f. As $f_{\mathrm{L}} = f$, we have $T_{\hat{i}} < f$ in $\mathrm{int}(\mathbb{D}_i), i = 1, \ldots, N$. Hence we can delete it without affecting the function value of f_{L}; or else if $T_{\hat{i}}$ is an implicant but not prime implicant, according to Lemma 2, we can find a prime implicant $\bar{T}_{\hat{i}} = \min_{j \in \bar{I}_{\hat{i}}}\{\ell_j\}$ with $\bar{I}_{\hat{i}} \subsetneq I_{\hat{i}}$ satisfying

$$T_{\hat{i}}(x) \leq \bar{T}_{\hat{i}}(x) \leq f(x), \forall x \in \mathbb{D}.$$

Define $g_1 = \max\{T_1, \ldots, T_{\hat{i}-1}, \bar{T}_{\hat{i}}, T_{\hat{i}+1}, \ldots, T_{\tilde{N}}\}$, then in $\mathbb{D}$ we have

$$f = f_{\mathrm{L}} \leq g_1 \leq f,$$

which means $g_1 \equiv f$ in $\mathbb{D}$ and then f_{L} is redundant.

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

$$f(x) = T_{i_k}(x), \ \forall x \in \mathbb{D}_k \text{ with } k \in \mathcal{C}(T_{\hat{i}}).$$

Define $g_2 = \max\{T_1, \ldots, T_{\hat{i}-1}, T_{\hat{i}+1}, \ldots, T_{\tilde{N}}\}$, then we have

$$g_2(x) \equiv f(x), \forall x \in \mathbb{D}_k, \forall k \in \{1, \ldots, N\},$$

and then f_{L} is redundant.

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 $k_{\hat{i}} \in \mathcal{C}(T_{\hat{i}})$ and $k_{\hat{i}} \notin \cup_{s=1,s\neq \hat{i}}^{\tilde{N}} \mathcal{C}(T_s)$. Thus, in $\mathrm{int}(\mathbb{D}_{k_{\hat{i}}})$, we have

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

Then $\max_{s=1,\ldots,\hat{i}-1,\hat{i}+1,\ldots,\tilde{N}} \{T_s\} \not\equiv f$, meaning that the removal of $T_{\hat{i}}$ affects the function value of f_{L}. Therefore, the two conditions ensure the irredundancy of f_{L}.

3.3 Removing redundant terms and literals

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

Corollary 1 *In the lattice PWA representation (17), the implicant* $T_i = \min_{j \in I_i}\{\ell_j\}$ *can be removed without affecting the function value of* f_{L} *in* $\mathbb{D}$ *if and only if*

$$\mathcal{C}(T_i) \subseteq \cup_{s=1,s\neq i}^{\tilde{N}} \mathcal{C}(T_s). \tag{19}$$

The proof of Corollary 1 can be found in (Xu et al., 2014).

We have to point out that Corollary 1 is different from the row vector simplification lemma in (Wen et al., 2009), which states that if

$$I_k \subseteq I_i, \tag{20}$$

then $T_i = \min_{j\in I_i}\{\ell_j\}$ can be removed without affecting the function value. In fact, according to Lemma 3, if (20) holds, we have $\mathcal{C}(T_i) \subseteq \mathcal{C}(T_k)$, which then indicates (19). However, there exist situations in which (19) holds but (20) is not satisfied. We will show this by revisiting Example 1 at the end of Section 3.3.

Then, according to the proof of Theorem 2, for an implicant $T_i^{\rm F}$ in $f_{\rm L}$, we can replace it with a prime implicant T_i with $I_i \subseteq I_{\geq,i}$ without affecting the function value of $f_{\rm L}$.

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

Theorem 3 *Given a term $T_i^{\rm F} = \min_{j\in I_{\geq,i}}\{\ell_j\}$ in the full lattice PWA representation (11). The term $T_i = \min_{j\in I_i}\{\ell_j\}$ with $I_i \subsetneq I_{\geq,i}$ is an implicant of f if and only if $\forall t \in I_{\geq,i}\setminus I_i$, $\forall \upsilon \in \mathcal{A}(\ell_t)$, there exists at least one $k_{t,\upsilon} \in I_i$ such that*

$$\ell_{k_{t,\upsilon}(x)} \leq f(x) = \ell_t(x), \forall x \in \mathbb{D}_\upsilon. \tag{21}$$

PROOF. From (8) and (9), (21) is equivalent to the inequality below

$$T_i(x) \leq f(x) = \ell_t(x), \forall x \in \mathbb{D}_\upsilon. \tag{22}$$

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

(1) Necessity. As T_i is an implicant of f, we have $T_i \leq f$ in $\mathbb{D}$. Thus, (22) should be satisfied, and (21) holds.

(2) Sufficiency. Assuming that (21) holds (or (22) holds) for all $t \in I_{\geq,i}\setminus I_i$ and all $\upsilon \in \mathcal{A}(\ell_t)$, then $\mathrm{act}(i) \in I_i$; otherwise, if $\mathrm{act}(i) \notin I_i$, there is no $k_{t,\upsilon} \in I_i$ such that (21) holds for $t = \mathrm{act}(i)$, $\upsilon = i$, and $x \in \mathrm{int}(\mathbb{D}_i)$.

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: First we prove that $T_i \leq f$ in $\mathbb{D}$. Suppose this is not true, i.e., there is a point $x_0 \in \mathbb{D}$ such that

$$T_i(x_0) > f(x_0). \tag{23}$$

Assume $x_0 \in \mathbb{D}_\beta$, then we have

$$\ell_{\mathrm{act}(\beta)}(x_0) < \min_{j\in I_i}\{\ell_j(x_0)\}. \tag{24}$$

Since both sides of (24) are continuous, such an x_0 can be found in the interior of $\mathbb{D}_\beta$, i.e., $x_0 \in \mathrm{int}(\mathbb{D}_\beta)$. As (22)

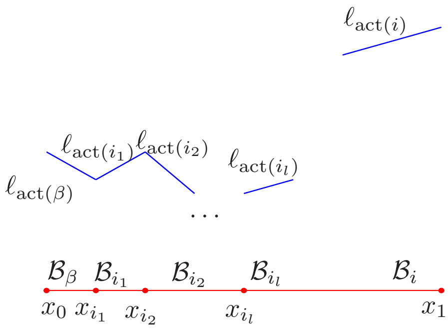


Fig. 3. The continuous PWA function f when restricted to $\mathcal{L}(x_0, x_1)$.

holds for all $t \in I_{\geq,i}\setminus I_i$ and all $\upsilon \in \mathcal{A}(\ell_t)$, we have $\mathrm{act}(\beta) \notin I_{\geq,i}\setminus I_i$. Besides, we have $\mathrm{act}(\beta) \notin I_i$ according to (24). Thus $\mathrm{act}(\beta) \notin I_{\geq,i}$.

Randomly choose a point $x_1 \in \mathrm{int}(\mathbb{D}_i)$. As $\mathrm{act}(\beta) \notin I_{\geq,i}$, we have

$$\ell_{\mathrm{act}(\beta)}(x_1) < \ell_{\mathrm{act}(i)}(x_1) = \min_{j\in I_i}\{\ell_j(x_1)\}. \tag{25}$$

Consider the line segment $\mathcal{L}(x_0, x_1)$ (14), and define $\mathcal{B}_\beta$ and $\mathcal{B}_i$ the same as (15); then we have $\mathrm{int}(\mathcal{B}_\beta) \neq \emptyset$ and $\mathrm{int}(\mathcal{B}_i) \neq \emptyset$. According to the proof of Theorem 1, f is continuous PWA when restricted to $\mathcal{L}(x_0, x_1)$.

Combining (24) and (25) we have

$$\ell_{\mathrm{act}(\beta)}(x) < \min_{j\in I_i}\{\ell_j(x)\} \leq \ell_{\mathrm{act}(i)}(x), \forall x \in \mathcal{L}(x_0, x_1). \tag{26}$$

Then according to the continuity of f, the line segments $\mathcal{B}_\beta$ and $\mathcal{B}_i$ are not adjacent. Thus there exists another line segment $\mathcal{B}_{i_1}$ with nonempty interior adjacent to $\mathcal{B}_\beta$. Define an index set $\mathcal{N}_1$ as,

$$\mathcal{N}_1 = \{1, \dots, N\} \setminus \{i, \beta\}, \tag{27}$$

then $i_1 \in \mathcal{N}_1$.

Assume $\mathcal{B}_{i_1} \cap \mathcal{B}_\beta = \{x_{i_1}\}$. Then $\mathcal{B}_{i_1} \subsetneq \mathcal{L}(x_{i_1}, x_1)$, in which $\mathcal{L}(x_{i_1}, x_1)$ is the line segment between x_{i_1} and x_1. We have

$$f(x_{i_1}) = \ell_{\mathrm{act}(\beta)}(x_{i_1}) = \ell_{\mathrm{act}(i_1)}(x_{i_1}) < \min_{j\in I_i}\{\ell_j(x_{i_1})\}. \tag{28}$$

Fig. 3 illustrates this.

Similar to the proof concerning $\mathrm{act}(\beta)$, we have $\mathrm{act}(i_1) \notin I_{\geq,i}$. Then we have

$$\ell_{\mathrm{act}(i_1)}(x) < \min_{j\in I_i}\{\ell_j(x)\} \leq \ell_{\mathrm{act}(i)}(x),\ \forall x \in \mathcal{L}(x_{i_1}, x_1).$$

Thus, according to the continuity of f, the line segments $\mathcal{B}_{i_1}$ and $\mathcal{B}_i$ are not adjacent. Then there must exist another line segment $\mathcal{B}_{i_2}$ adjacent to $\mathcal{B}_{i_1}$ and $\mathrm{int}(\mathcal{B}_{i_2}) \neq \emptyset$. Let the index set $\mathcal{N}_2$ be defined as

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

then as $\mathcal{B}_\beta$ is convex, $i_2 \neq \beta$, and further, $i_2 \in \mathcal{N}_2$.

Repeating the above procedure if necessary, as Fig. 3 shows, after l ($l < N$) iterations, we can reach an empty index set $\mathcal{N}_{l+1}$ and a point $x_{i_l} \in \mathcal{B}_{i_l} \subseteq \mathcal{L}(x_{i_l}, x_1)$ such that

$$\ell_{\mathrm{act}(i_l)}(x) < \min_{j \in I_i}\{\ell_j(x)\} \leq \ell_{\mathrm{act}(i)}(x), \forall x \in \mathcal{L}(x_{i_l}, x_1).$$

This contradicts the continuity of f. Therefore, $T_i(x) \leq f(x)$ 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., $\mathcal{C}(T_i) \neq \emptyset$. Considering $\mathbb{D}_k, k \in \mathcal{C}(T_i^{\mathrm{F}})$. As

$$f(x) = T_i^{\mathrm{F}}(x) \leq T_i(x) \leq f(x), \forall x \in \mathbb{D}_k,$$

we have $T_i \equiv T_i^{\mathrm{F}} \equiv f$ in $\mathbb{D}_k, \forall k \in \mathcal{C}(T_i^{\mathrm{F}})$. Therefore, T_i is an implicant of f.

Using Theorem 3, we can delete literals in a term T_i^{F} until further deletion is impossible, and the prime implicants are obtained. The implicants obtained can replace the original implicant in the full lattice PWA representation (11).

It should be noted that Theorem 3 is different from the column vector simplification lemma in (Wen et al., 2009), which is proposed for the conjunctive form and when using duality, it can be rephrased as follows: T_i^{F} can be replaced by T_i without affecting the function value if for all $t \in I_{\geq,i} \setminus I_i$, for all α such that $\mathrm{loc}(\alpha) = t$, there is some $k_{t,\alpha} \in I_i$ such that

$$\ell_{k_{t,\alpha}}(x) \leq f(x) = \ell_t(x), \forall x \in \Omega_\alpha. \tag{29}$$

Revisiting Example 1, we will now show that not only (20) of (Wen et al., 2009) is a sufficient condition for removing redundant terms, but also (29) of (Wen et al., 2009) is a sufficient condition for removing redundant literals.

Example 1 (Continued): Reconsidering Example 1, now we can use Theorem 3 to explain why ℓ_3, ℓ_4 can be removed from $T_1^{\mathrm{F}} = \min\{\ell_1, \ell_3, \ell_4, \ell_5\}$ without affecting the function value.

As $\mathcal{A}(\ell_3) = \{3,4,5\}$, $\ell_1(x) \leq \ell_3(x), \forall x \in \mathbb{D}_3 \cup \mathbb{D}_4$, $\ell_5(x) \leq \ell_3(x), \forall x \in \mathbb{D}_5$; besides, $\mathcal{A}(\ell_4) = \{6\}$ and $\ell_5(x) \leq \ell_4(x), \forall x \in \mathbb{D}_6$. According to Theorem 3, the term $T_1 = \min\{\ell_1, \ell_5\}$ is an implicant of f as defined in (16). Since neither ℓ_1 nor ℓ_5 can be further removed, T_1 is a prime implicant of f and it can replace T_1^{F} without affecting the function value of f.

Similarly, we obtain the prime implicants

$$\begin{aligned} T_2 &= \min\{\ell_2, \ell_3, \ell_4\}, T_3 = \min\{\ell_2, \ell_3, \ell_4\}, \\ T_4 &= \min\{\ell_2, \ell_3, \ell_4\}, T_5 = \min\{\ell_1, \ell_3, \ell_4\}, \\ T_6 &= \min\{\ell_1, \ell_3, \ell_4\}, T_7 = \min\{\ell_1, \ell_5\}. \end{aligned}$$

The indices of base regions the prime implicants cover are $\mathcal{C}(T_1) = \mathcal{C}(T_7) = \{1, 7\}, \mathcal{C}(T_2) = \mathcal{C}(T_3) = \mathcal{C}(T_4) = \{2, 3, 4, 5, 6\}, \mathcal{C}(T_5) = \mathcal{C}(T_6) = \{1, 5, 6\}$.

According to Corollary 1, as $\mathcal{C}(T_3) = \mathcal{C}(T_4) \subseteq (\mathcal{C}(T_1) \cup \mathcal{C}(T_2))$, $\mathcal{C}(T_5) = \mathcal{C}(T_6) \subseteq (\mathcal{C}(T_1) \cup \mathcal{C}(T_2))$ and $\mathcal{C}(T_7) \subseteq (\mathcal{C}(T_1) \cup \mathcal{C}(T_2))$, we can remove the terms T_3, T_4, T_5, T_6, T_7 and obtain the following irredundant lattice PWA representation:

$$f_{\mathrm{L}} = \max\{\min\{\ell_1, \ell_5\}, \min\{\ell_2, \ell_3, \ell_4\}\}. \tag{30}$$

Conversely, if we apply the procedures of (Wen et al., 2009), first delete redundant rows, we obtain

$$\begin{aligned} \max\{ &\min\{\ell_1, \ell_3, \ell_4, \ell_5\}, \min\{\ell_2, \ell_3, \ell_4\}, \\ &\min\{\ell_1, \ell_2, \ell_3, \ell_5\}\}. \end{aligned}$$

Then we delete redundant literals. For the first term, as $\ell_5 \leq \ell_4$ in Ω_4, the term $\min\{\ell_1, \ell_3, \ell_4, \ell_5\}$ can be reduced to $\min\{\ell_1, \ell_3, \ell_5\}$. As neither ℓ_1 nor ℓ_5 is less than or equal to ℓ_3 in Ω_3, according to the column vector simplification lemma of (Wen et al., 2009), no literals can be further deleted.

For the second term, no literal can be deleted as it is irredundant. For the third term, as $\ell_2 \leq \ell_1$ in Ω_1, we can remove ℓ_1 and obtain $\min\{\ell_2, \ell_3, \ell_5\}$.

Therefore, the procedure of (Wen et al., 2009) will result in the following lattice PWA representation:

$$\begin{aligned} \tilde{f} = \max\{ &\min\{\ell_1, \ell_3, \ell_5\}, \\ &\min\{\ell_2, \ell_3, \ell_4\}, \min\{\ell_2, \ell_3, \ell_5\}\}. \end{aligned} \tag{31}$$

Compared with (30), we can see that both the term $\min\{\ell_2, \ell_3, \ell_5\}$ and the literal ℓ_3 in the term $\min\{\ell_1, \ell_3, \ell_5\}$ are redundant. Therefore, the row and column vector simplification lemma of (Wen et al., 2009) are only sufficient conditions for removing redundant terms and literals.

An interesting phenomenon is that the lattice PWA representation

$$f_{\mathrm{L}}^2 = \max\{\min\{\ell_1, \ell_3, \ell_4\}, \min\{\ell_2, \ell_3, \ell_5\}, \min\{\ell_2, \ell_3, \ell_4\}\} \tag{32}$$

is also irredundant and equals f in $[0, 5]$. Although (32) is also irredundant, the number of parameters is larger than that of (30), meaning that there may exist multiple irredundant lattice PWA representations with different number of parameters.

It should be also noted that the number of parameters in the irredundant lattice PWA representation (32) and the redundant lattice PWA representation (31) are the same. Hence, we cannot say that all irredundant lattice PWA representations are more compact than redundant ones. However, for any redundant lattice PWA representation, we can use Corollary 1 and Theorem 3 to get a corresponding irredundant one.

The following section summarizes the steps for obtaining an irredundant lattice PWA representation.

3.4 Algorithm for obtaining an irredundant lattice PWA representation

---

Algorithm 1 *Obtaining an irredundant lattice PWA representation.*

Input: *Continuous PWA function f with subregions $\Omega_1, \ldots, \Omega_{\hat{N}}$ and local affine functions $\ell_{\mathrm{loc}(1)}, \ldots, \ell_{\mathrm{loc}(\hat{N})}$ with $\mathrm{loc}(i) \in \{1, \ldots, M\}, i \in \{1, \ldots, \hat{N}\}$.*

Output: *Irredundant lattice PWA representation*

$$f_{\mathrm{L}} = \max_{i \in \tilde{\mathcal{N}}} \{\min_{j \in I_i} \{\ell_j\}\}.$$

1: *Divide $\Omega_1, \ldots, \Omega_{\hat{N}}$ into base regions $\mathbb{D}_1, \ldots, \mathbb{D}_N$.*
2: *Compute $I_{\geq, i}$ and T_i^{F} for $i = 1, \ldots, N$ to obtain the full lattice PWA representation (11) ;*
3: ***for** $i = 1 : N$ **do***
4:     $I_i = I_{\geq, i}$;
5:     ***for** $j \in I_i$ **do***
6:         $\bar{I}_i = I_i \setminus \{j\}$;
7:         ***if** $\min_{k \in \bar{I}_i} \{\ell_k\}$ is an implicant of f **then***
8:             $I_i = \bar{I}_i$;
9:         ***end if***
10:     ***end for***
11: ***end for***
12: *$\tilde{\mathcal{N}} = \{1, \ldots, N\}$;*
13: ***for** $i \in \{1, \ldots, N\}$ **do***
14:     ***if** $\mathcal{C}(\min_{j \in I_i} \{\ell_j\}) \subseteq \cup_{v \in \tilde{\mathcal{N}}, v \neq i} \mathcal{C}(\min_{j \in I_v} \{\ell_j\})$ **then***
15:         $\tilde{\mathcal{N}} = \{1, \ldots, N\} \setminus \{i\}$;
16:     ***end if***
17: ***end for***

In Algorithm 1, the second and third "For" block (line 5 and 13) are for removing redundant literals and terms, respectively. It should be noted that different search sequences in line 5 may generate different implicants T_i; so in the current algorithm we just choose one particular sequence and only get one prime implicant. For the third "For" (line 13) block, different search sequences may yield different sets $\tilde{\mathcal{N}}$, and we just select one sequence. Therefore, Algorithm 1 will result in only one irredundant lattice PWA representation, although there may exist multiple ones. We refer the reader to our recent paper (Xu et al., 2016) for the derivation of the most compact representation.

3.5 Complexity Analysis

3.5.1 Storage requirements of irredundant lattice PWA representations

For an irredundant lattice PWA representation (17) with M distinct affine functions and $\tilde{N}$ terms ($\tilde{N} \leq N$), we have to store $(n+1) \cdot M$ real numbers and $\sum_{i=1}^{\tilde{N}} |I_i|$ integer numbers, in which $|I_i|$ denotes the cardinality of the index set I_i and $|I_i| \leq M$.

If the continuous PWA function is expressed via subregions and affine functions defined on them, as there are $\hat{N}$ subregions, one has to store $(n+1) \cdot \hat{N} + \sum_{i=1}^{\hat{N}} r_i \cdot (n+1)$ real numbers, in which r_i is the number of linear inequalities defining the i-th subregion. For an n-dimensional problem, if the subregion Ω_i is bounded, we have $r_i \geq n+1$ and the inequality becomes an equality when Ω_i is a simplex. Hence, for the bounded case, the required storage is greater than or equal to $(n+1) \cdot \hat{N} + \sum_{i=1}^{\hat{N}} (n+1)^2$.

In many cases we encountered, we have $\tilde{N} < \hat{N}$, and generally speaking, $M < \hat{N}$, so if the size of $|I_i|$ is close to $(n+1)^2$, the storage requirements of an lattice PWA representation is less than that of the expression with the subregions and local affine functions. In linear explicit MPC, which will be considered in the next section, usually there are many subregions sharing the same local affine function, hence $M \ll \hat{N}$. So compared with the continuous PWA solution given by the MPT Toolbox, the storage requirements will be decreased significantly using the lattice PWA representations.

3.5.2 Offline preprocessing

Assume the continuous PWA function is given by the subregions and local affine functions. And that we use Algorithm 1 for offline preprocessing, i.e., obtaining an irredundant lattice PWA representation. The following lemma gives the worst-case time complexity of Algorithm 1.

Lemma 4 *The worst-case time complexity of Algorithm 1 is* $O\left(\sum_{i=1}^{\hat{N}} 2^{M-1}(r_i+M-1)^3 L_i\right)$*, in which* r_i *is the number of linear inequalities defining* Ω_i *and* L_i *is the bit length of the input data of the linear programming (LP) problem (A.11).*

PROOF. According to Appendix A, the time complexity for evaluating Line 1 to 2 of Algorithm 1 is $O\left(\sum_{i=1}^{\hat{N}} 2^{M-1}(r_i+M-1)^3 L_i\right)$.

Then from line 3 to 11, for each $i \in \{1,\ldots,N\}$, line 6 may be evaluated at most $|I_{\geq,i}|-1$ times. For each $\bar{I}_i$, in order to check line 7, we use Theorem 3 to check whether there exists some element $k_{t,v} \in \bar{I}_i$ such that $\ell_{k_{t,v}}(x) \leq \ell_t(x), \forall x \in \mathbb{D}_v$ for any $t \in I_{\geq,i} \setminus \bar{I}_i$, and any $v \in \mathcal{A}(\ell_t)$. According to (8) and (9), this is equivalent to check whether $k_{t,v} \notin I_{\geq,v}$, which requires $|\bar{I}_i||I_{\geq,v}|$ comparisons in $|\mathcal{A}(\ell_t)|$ base regions. As $|\bar{I}_i| \leq M$, $|I_{\geq,v}| \leq M$, $|\mathcal{A}(\ell_t)| \leq N$, in the worst case, the time complexity for evaluating line 3 to 11 is $O\left(\sum_{i=1}^{N}(M-1)M^2N\right) = O(M^3N^2)$.

Thirdly, when evaluating line 13 to 17, we have to first calculate $\mathcal{C}(\min_{j\in I_i}\{\ell_j\}), \forall i \in \{1,\ldots,N\}$, i.e, we have to check whether $I_i \subseteq I_{\geq,t}, t \in \{1,\ldots,N\}$, which requires $\sum_{i=1}^{N}\sum_{t=1}^{N}|I_i||I_{\geq,t}|$ comparisons. Then checking the condition in line 14 N times requires $\sum_{i=1}^{N}\sum_{v\in\tilde{\mathcal{N}},v\neq i}|I_i||I_v|$ comparisons. Hence, the worst-case complexity for evaluating line 13 to 17 is $O(M^2N^2)$.

In general, $O(M^3N^2) < O\left(\sum_{i=1}^{\hat{N}} 2^{M-1}(r_i+M-1)^3 L_i\right)$. Thus the worst-case time complexity of Algorithm 1 is $O\left(\sum_{i=1}^{\hat{N}} 2^{M-1}(r_i+M-1)^3 L_i\right)$.

3.5.3 Online Evaluation

Assume there are $\tilde{N}$ terms in the irredundant lattice PWA representation, according to (Wen et al., 2009), the worst-case online evaluation complexity is $O(\tilde{N}^2)$.

4 Application to linear explicit MPC

MPC problem with quadratic cost function for a discrete-time linear time-invariant system can be cast as the following optimization problem at time step t:

$$\min_U \left\{ J(U,x_t) = x_{t+N_y}^T P x_{t+N_y} + \sum_{k=0}^{N_y-1}\left[x_{t+k}^T Q x_{t+k} + u_{t+k}^T R u_{t+k}\right]\right\} \tag{33a}$$

$$\text{s.t. } y_{\min} \leq y_{t+k} \leq y_{\max}, k=1,\ldots,N_y, \tag{33b}$$

$$u_{\min} \leq u_{t+k} \leq u_{\max}, k=0,1,\ldots,N_y-1, \tag{33c}$$

$$x_{t+k+1} = Ax_{t+k} + Bu_{t+k}, k=0,1,\ldots,N_y-1, \tag{33d}$$

$$y_{t+k} = Cx_{t+k}, k=1,\ldots,N_y, \tag{33e}$$

$$u_{t+k} = Kx_{t+k}, k=N_u,\ldots,N_y-1, \tag{33f}$$

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 (33d). The matrix K is the feedback gain of a stabilizing controller. We assume $Q,P \succcurlyeq 0$, $R \succ 0$. After solving the optimization problem (33), 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 updating the state vector x_t.

It is proved in (Bemporad et al., 2002b) that the solution u_t^* of (33) is a continuous PWA function of the state x_t.

In (Wen et al., 2009), a lattice PWA representation is used to represent the resulting continuous PWA solution. The lattice PWA representation is also simplified to give a more compact expression. However, as pointed out in Section 3, the irredundancy of the simplification results in (Wen et al., 2009) cannot be guaranteed. Hence, we now give the irredundant lattice PWA representations to simplify the explicit MPC output.

Now we give 2 worked examples, one is 2-dimensional and the other is 4-dimensional, and apply the irredundant lattice PWA representations to express the optimal solution in linear explicit MPC problem.

Example 2 *Consider the discrete-time double integrator example introduced in (Bemporad et al., 2002b), and for which the system dynamics can be written as*

$$\begin{aligned} x_{k+1} &= \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} x_k + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u_k, \\ y_k &= \begin{bmatrix} 1 & 0 \end{bmatrix} x_k. \end{aligned} \tag{34}$$

Consider the MPC problem (33) with $Q = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$, $R = 0.01$ *and* $P = \begin{bmatrix} 2.0191 & 1.0288 \\ 1.0288 & 1.0484 \end{bmatrix}$. *The system constraint is* $-1 \leq u_k \leq 1$.

Assume $N_y = N_u = 10$. *First we use the MPT Toolbox version 3.0.16 (Herceg et al., 2013a) to compute the optimal output* u_t *as a function of* x_t. *This yields a continuous PWA function with 303 subregions, in each of which there is a corresponding local affine function. Among all the affine functions, there are only 41 unique ones; hence, several subregions share the same local affine function. After applying Algorithm 1, only 18 terms are left. If the procedure of (Wen et al., 2009) is used, there are 24 terms, indicating that the procedure in (Wen et al., 2009) may result in redundant representations. Hence, the original solution calculated by the MPT Toolbox can be represented by a more compact irredundant lattice PWA representation.*

For $N_y = N_u = 2, 6, 10, 14, 20$, *Table 1 compares the complexity of five methods, i.e., the MPT output, the lattice PWA representation in (Wen et al., 2009), denoted as "LR", the irredundant lattice PWA representation, denoted as "ILR", the binary search tree (BST) of (Tøndel et al., 2003a), and the graph traversal (GT) method of (Herceg et al., 2013b). The complexity includes the storage requirements, the preprocessing time as well as the online evaluation time. In Table 1,* N_{aff} *and* N_{reg} *represent the number of distinct affine functions and subregions given by the MPT Toolbox.*

It is noted that here the BST is exported from the MPT Toolbox, and the graph traversal method is also realized through the MPT Toolbox. The preprocessing time for MPT output is set to be "—" as the other methods are based on the MPT output; besides, the adjacency list for the GT method is included in the MPT output; hence the preprocessing time for GT is also recorded as "—". The online evaluation is carried out for 1000 points and the recorded time is the mean. For the examples in this paper, both the offline preprocessing and the online evaluation are implemented through MATLAB 2012b on a 2.4 GHz Intel Core i5 computer.

From the table, for all the prediction horizons, we can see that the number of parameters in the irredundant lattice PWA representation is the least among the five methods. According to the analysis in Section 3.5, the storage requirements for the irredundant lattice PWA representation are much lower than those of the MPT output. Besides, in the binary search tree and graph traversal method, compared with the MPT output, more information has to be stored in order to facilitate the point location procedure. Thus the storage requirements for these two methods are even higher. Compared with the lattice PWA representation of (Wen et al., 2009), the irredundant lattice PWA representation has a smaller number of parameters and is faster for online evaluation, which is more evident when N_y *increases.*

Of course, the preprocessing time of the irredundant lattice PWA representation is longer than that of the lattice PWA representation of (Wen et al., 2009) because it takes time to delete all the redundant parameters. For this example, the number of parameters in the irredundant lattice PWA representation is the least and the online evaluation performance is also excellent. At the same time, the preprocessing time is not too long.

Table 1
Comparison of the complexity of five methods.

N_y	Method	N_{aff}	N_{reg}	Storage (Numbers)	Preprocessing Time (s)	Evaluation Time (ms)
2	MPT	7	13	195	—	1.5
	LR			30	0.01	0.06
	BST			311	0.85	0.06
	GT			247	—	1.3
	ILR			30	0.09	0.06
6	MPT	17	87	1305	—	2.0
	LR			81	0.32	0.12
	BST			2719	45.17	0.12
	GT			1653	—	2.2
	ILR			81	1.5	0.12
10	MPT	41	303	4545	—	3.9
	LR			262	1.7	0.42
	BST			10181	600	0.07
	GT			5757	—	5
	ILR			203	15	0.27
14	MPT	71	747	11205	—	8.3
	LR			518	8.99	1.0
	BST			26799	4050.2	0.07
	GT			14193	—	7.3
	ILR			390	89.36	0.62
20	MPT	113	1829	27435	—	18.4
	LR			1136	74	2.4
	BST			63728	29728	0.09
	GT			34751	—	15.4
	ILR			754	1022.8	1.3

Example 3 *Consider the following linear system taken*

from (Borrelli, 2003):

$$x_{k+1} = \begin{bmatrix} 4 & -1.5 & 0.5 & -0.25 \\ 4 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 0.5 & 0 \end{bmatrix} x_k + \begin{bmatrix} 0.5 \\ 0 \\ 0 \\ 0 \end{bmatrix} u_k$$
$$y_k = \begin{bmatrix} 0.083 & 0.22 & 0.11 & 0.02 \end{bmatrix} x_k.$$

The system is subject to input constraints $-1 \le u_k \le 1$*, state constraints* $-10 \le x_{k,i} \le 10, i = 1, ..., 4$*, and output constraints* $-10 \le y_k \le 10$*. The MPC controller is designed with* $N_y = N_u = 6$*,* $Q = \text{diag}\{1, 1, 1, 1\}$*,* $R = 0.01$*, and* $P = 0$*. The explicit solution consists of 437 regions.*

Table 2 compares the performance of four methods. Note that the binary search tree is not listed for this example as its preprocessing time is too long.

Table 2
Comparison of the complexity of four methods.

N_y	Method	N_{aff}	N_{reg}	Storage (Numbers)	Preprocessing Time (s)	Evaluation Time (ms)
10	MPT	93	437	22985	—	5.6
	LR			5580	35.64	1.2
	GT			27145	—	5.4
	ILR			4557	2648	0.93

The notations in Table 2 are the same as those in Table 1. For this example, compared with the other 3 methods, the irredundant lattice PWA representation also saves significantly in storage requirements.

Although there are only 437 subregions and 93 distinct affine functions, compared with Example 2, the preprocessing time for the irredundant lattice PWA representation is longer. This is due to the increase in dimension. For higher dimensions, it is more likely that other affine functions intersect with the local affine functions in the interior of the subregions, thus yielding more base regions. Besides, the vertices are hard to derive in higher dimension and LP problems have to be solved to determine $I_{\ge,i,t}, i = 1, \ldots, \hat{N}, t = 1, \ldots, m_i$*. Hence, the offline preprocessing time is increased significantly.*

When we set x_k *to be unbounded, there are 890 subregions and 265 distinct affine functions. In this case, the offline preprocessing for the irredundant lattice PWA representation explodes, i.e., the number of base regions exceeds 300000 and the preprocessing time exceeds 24 hours. Therefore, in this case, we may resort to some other methods.*

5 Conclusions and Future work

In this paper, we have derived the irredundant lattice PWA representations, which are realized by removing redundant terms and literals in the full lattice PWA representation. The full lattice PWA representation is defined on base regions and we show that by choosing appropriate parameters it can represent any continuous PWA function. We have proposed the necessary and sufficient conditions for irredundancy as well as for removing redundant terms and literals. Based on this, an algorithm has been put forward to obtain an irredundant lattice PWA representation of any given continuous PWA function. The storage requirements of irredundant lattice PWA representations as well as the offline and online complexity have been analyzed. The irredundant lattice PWA representations have been applied to express the optimal solution of explicit MPC problem. The simulation results show that the number of parameters needed to describe a continuous PWA functions is largely reduced by using irredundant lattice PWA representations. Besides, the online evaluation speed is also improved.

For problems with a high dimension, a large number of subregions and distinct affine functions, when the preprocessing time for an irredundant lattice PWA representation explodes, maybe we can combine the irredundant lattice PWA representation and some other point location algorithms, like (Bayat et al., 2012), in which examples are given to combine truncated binary search tree with lattice PWA representations of (Wen et al., 2009).

Acknowledgements

The authors would like to thank the anonymous reviewers for carefully reading the manuscript, and for their constructive comments and valuable suggestions.

A Partition the subregions into base regions

A.1 Proof of Lemma 1.

PROOF. Define the following index sets:

$$K_{\ge,i} = \{j | \ell_j(x) \ge \ell_{\text{loc}(i)}(x), \forall x \in \Omega_i\}, \tag{A.1}$$
$$K_{\le,i} = \{j | \ell_j(x) \le \ell_{\text{loc}(i)}(x), \forall x \in \Omega_i\}, \tag{A.2}$$
$$\mathsf{J}_i = \{1, \ldots, M\} \setminus (K_{\ge,i} \cup K_{\le,i}). \tag{A.3}$$

It is clear that if $\mathsf{J}_i = \emptyset$, we have (3), and then Ω_i does not need to be partitioned. Moreover we have $m_i = 1$, $\mathbb{D}_{i,1} = \Omega_i$, $I_{\ge,i,1} = K_{\ge,i}$ and $I_{\le,i,1} = K_{\le,i}$.

If $\mathsf{J}_i \neq \emptyset$, suppose $\mathsf{J}_i = \{i_1, \ldots, i_{N_i}\}$. Then for each $i_j \in \mathsf{J}_i$, we consider two sets:

$$\Omega_{\geq,i_j} = \{x | x \in \Omega_i, \ell_{i_j}(x) \geq \ell_{\mathrm{loc}(i)}(x)\}, \quad \text{(A.4)}$$
$$\Omega_{\leq,i_j} = \{x | x \in \Omega_i, \ell_{i_j}(x) \leq \ell_{\mathrm{loc}(i)}(x)\}. \quad \text{(A.5)}$$

Since $i_j \in \mathsf{J}_i$, we have $\mathrm{int}(\Omega_{\geq,i_j}) \neq \emptyset$ and $\mathrm{int}(\Omega_{\leq,i_j}) \neq \emptyset$. Besides, according to (A.4) and (A.5), the following holds,

$$\mathrm{int}(\Omega_{\geq,i_j}) \cap \mathrm{int}(\Omega_{\leq,i_j}) = \emptyset,\ \Omega_{\geq,i_j} \cup \Omega_{\leq,i_j} = \Omega_i. \quad \text{(A.6)}$$

From (A.6), we have

$$\cap_{i_j \in \mathsf{J}_i} \left(\Omega_{\geq,i_j} \cup \Omega_{\leq,i_j}\right) = \Omega_i. \quad \text{(A.7)}$$

If we define the set $\mathcal{W}^{N_i} = \{(w_{i_1}, \ldots, w_{i_{N_i}}) | w_{i_j} \in \{\geq, \leq\}, j = 1, \ldots, N_i\}$, we can further write (A.7) as

$$\cup_{(w_{i_1}, \ldots, w_{i_{N_i}}) \in \mathcal{W}^{N_i}} \left(\Omega_{w_{i_1}, i_1} \cap \cdots \cap \Omega_{w_{i_{N_i}}, i_{N_i}}\right) = \Omega_i. \quad \text{(A.8)}$$

According to (A.8), as Ω_i is not empty, there exist combinations $(w_{i_1}, \ldots, w_{i_{N_i}})$ such that $\mathrm{int}(\Omega_{w_{i_1}, i_1} \cap \cdots \cap \Omega_{w_{i_{N_i}}, i_{N_i}})$ is nonempty. Assume the subregion Ω_i can be described as

$$E_i x \leq e_i. \quad \text{(A.9)}$$

For each combination of $(w_{i_1}, \ldots, w_{i_{N_i}})$, we have to check if there exists some x such that the following holds

$$\begin{aligned} & E_i x \leq e_i, \\ & \ell_{i_j}(x) \geq \ell_{\mathrm{loc}(i)}(x), \forall \omega_{i_j} = \geq, i_j \in \mathsf{J}_i \\ & \ell_{i_j}(x) \leq \ell_{\mathrm{loc}(i)}(x), \forall \omega_{i_j} = \leq, i_j \in \mathsf{J}_i \end{aligned} \quad \text{(A.10)}$$

i.e., whether the combination yields an intersection with a nonempty interior. Suppose the linear inequalities (A.10) can be described as $\mathcal{E}_i x \leq \epsilon_i$. According to Farkas' Lemma (Ziegler, 1995), there is no x such that (A.10) is valid if and only if the optimal value of the following LP problem

$$\begin{aligned} & \min_z -\epsilon_i^T z, \\ & s.t.\ \mathcal{E}_i^T z = 0, \\ & \quad z \geq 0, \end{aligned} \quad \text{(A.11)}$$

is positive. Hence, we can solve (A.11) to judge whether the resulting intersection has a nonempty interior.

We collect all the intersections with nonempty interior and denote them as $\mathbb{D}_{i,1}, \ldots, \mathbb{D}_{i,t}, \ldots, \mathbb{D}_{i,m_i}$. Then

$$\Omega_i = \mathbb{D}_{i,1} \cup \cdots \mathbb{D}_{i,t} \cup \cdots \mathbb{D}_{i,m_i}.$$

For $\mathbb{D}_{i,t}$, define

$$\Gamma_{\geq,i,t} = \{i_j | w_{i_j} = \geq\},\ \Gamma_{\leq,i,t} = \{i_j | w_{i_j} = \leq\}.$$

Then we have

$$\begin{aligned} \mathbb{D}_{i,t} = \{x | x \in \Omega_i, & \ell_j(x) \geq \ell_{\mathrm{loc}(i)}(x), \forall j \in \Gamma_{\geq,i,t}, \\ & \ell_j(x) \leq \ell_{\mathrm{loc}(i)}(x), \forall j \in \Gamma_{\leq,i,t}\}, \end{aligned} \quad \text{(A.12)}$$

where $\Gamma_{\geq,i,t} \cap \Gamma_{\leq,i,t} = \emptyset$, $\Gamma_{\geq,i,t} \cup \Gamma_{\leq,i,t} = \mathsf{J}_i$.

According to the expression (A.12) for $\mathbb{D}_{i,t}$, we have $I_{\geq,i,t} = K_{\geq,i} \cup \Gamma_{\geq,i,t}$ and $I_{\leq,i,t} = K_{\leq,i} \cup \Gamma_{\leq,i,t}$.

As $\Gamma_{\geq,i,t} \cup \Gamma_{\leq,i,t} = \mathsf{J}_i$ and $K_{\geq,i} \cup K_{\leq,i} \cup \mathsf{J}_i = \{1, \ldots, M\}$, we have (5).

For two sets $\mathbb{D}_{i,\bar{t}}$ and $\mathbb{D}_{i,\hat{t}}$ with $\bar{t} \neq \hat{t}$, the combinations $(w_{i_1}, \ldots, w_{i_{N_i}})$ must be different. Then according to (A.4) and (A.5), we have

$$\mathrm{int}(\mathbb{D}_{i,\bar{t}}) \cap \mathrm{int}(\mathbb{D}_{i,\hat{t}}) = \emptyset. \quad \text{(A.13)}$$

From Definition 1, we have $\mathrm{int}(\Omega_i) \cap \mathrm{int}(\Omega_j) = \emptyset$ for any $i \neq j$, which together with (A.13) yields (6). Hence, $\mathbb{D}$ is partitioned into disjoint nonempty base regions $\mathbb{D}_{1,1}, \ldots, \mathbb{D}_{1,m_1}, \ldots, \mathbb{D}_{\hat{N},1}, \ldots, \mathbb{D}_{\hat{N},m_{\hat{N}}}$. Besides, the index sets $I_{\geq,i,j}, I_{\leq,i,j}, i = 1, \ldots, \hat{N}, j = 1, \ldots, m_i$ are obtained.

A.2 Time complexity of the partition process.

Lemma 5 *For the process of partitioning the subregions into base regions, the worst-case time complexity is* $O\left(\sum_{i=1}^{\hat{N}} 2^{M-1}(r_i + M - 1)^3 L_i\right)$, *in which* L_i *is the bit length of the input data of the LP problem (A.11).*

PROOF. First, we have to calculate the index sets (A.1)-(A.3), i.e., finding the indices of affine functions that are larger than (less than) or equal to $\ell_{\mathrm{loc}(i)}$ in Ω_i, $i = 1, \ldots, \hat{N}$. This can be done through the evaluation of the affine functions at the vertices of Ω_i (Wen et al., 2009). However, the vertices of the subregions may not be readily available. Suppose $\ell_j(x) = a_j^T x + b_j$ for all $j \in \{1, \ldots, M\}$, in which $a_j \in \mathbb{R}^n$ and $b_j \in \mathbb{R}$. According to Farkas' lemma (Ziegler, 1995) and (A.9), we have $j \in K_{\geq,i}$ if and only if the optimal value of the LP problem

$$\begin{aligned} & \min_z \ -e_i^T z + b_j - b_{\mathrm{loc}(i)}, \\ & s.t.\ E_i^T z = a_{\mathrm{loc}(i)} - a_j, \\ & \quad z \geq 0, \end{aligned} \quad \text{(A.14)}$$

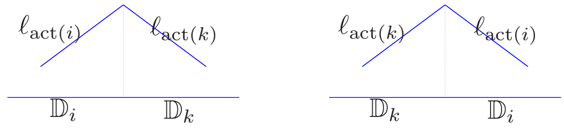
is nonnegative. Similarly, we have $j \in K_{\leq,i}$ if and only if the optimal value of the LP problem

$$\begin{aligned} \min_{z} \quad & -e_i^T z + b_{\mathrm{loc}(i)} - b_j, \\ s.t. \; & E_i^T z = a_j - a_{\mathrm{loc}(i)}, \\ & z \geq 0, \end{aligned} \tag{A.15}$$

is nonnegative. In each subregion Ω_i, we then have to solve at most $2(M-1)$ LP problems to determine the index sets $K_{\geq,i}, K_{\leq,i}, i = 1, \ldots, \hat{N}$. According to (Gonzaga, 1995), the worst-case time complexity for solving those LP problems using the interior method is $O\left(\sum_{i=1}^{\hat{N}} 2(M-1) r_i^3 \tilde{L}_i\right)$, in which $\tilde{L}_i$ is the bit length of the input data of the LP problem (A.14) and (A.15), and r_i is the number of rows in E_i. The index sets $\mathsf{J}_i, i = 1, \ldots, \hat{N}$ can be calculated through (A.3), which requires $M(|K_{\geq,i}| + |K_{\leq,i}|)$ comparisons. Generally speaking, $O(M^2 \hat{N}) < O\left(\sum_{i=1}^{\hat{N}} 2(M-1) r_i^3 \tilde{L}_i\right)$, hence the worst-case complexity for this step is $O\left(\sum_{i=1}^{\hat{N}} 2(M-1) r_i^3 \tilde{L}_i\right)$.

Then we collect all the intersections with nonempty interior and obtain the base regions $\mathbb{D}_{1,1}, \ldots, \mathbb{D}_{1,m_1}, \ldots, \mathbb{D}_{\hat{N},1}, \ldots, \mathbb{D}_{\hat{N},m_{\hat{N}}}$. As there are M distinct affine functions, in each subregion Ω_i, there are at most $M-1$ affine functions that can intersect with the local affine function $\ell_{\mathrm{loc}(i)}$ at some point in the interior of Ω_i. According to the proof of Lemma 1, for each Ω_i, the partition process may generate at most 2^{M-1} combinations $(\omega_{i_1}, \ldots, \omega_{i_{M-1}})$. For each combination, we solve the LP problem (A.11) to check whether it has a nonempty interior. Therefore, if the interior method is used for the LP problem (A.11), the worst-case time complexity for this step is $O\left(\sum_{i=1}^{\hat{N}} 2^{M-1}(r_i + M - 1)^3 L_i\right)$, where $r_i + M - 1$ is the number of rows in $\mathcal{E}_i$ and L_i is the bit length of the input data of the LP problem (A.11).

Generally speaking, we have $O\left(\sum_{i=1}^{\hat{N}} 2(M-1) r_i^3 \tilde{L}_i\right) < O\left(\sum_{i=1}^{\hat{N}} 2^{M-1}(r_i + M - 1)^3 L_i\right)$. Hence, the worst-case time complexity for the partition process is $O\left(\sum_{i=1}^{\hat{N}} 2^{M-1}(r_i + M - 1)^3 L_i\right)$.



(a) $i = k-1$ (b) $i = k+1$

Fig. B.1. Case 1

(a) $i = k-1$ (b) $i = k+1$

Fig. B.2. Case 2

B Proof of Proposition 1

PROOF. As $\mathbb{D} \subseteq \mathbb{R}^1$ is convex, the line segment between two points in $\mathbb{D}$ still lies in $\mathbb{D}$. We number the base regions from the left to the right. Then we prove (10) by mathematical induction.

Basis: The base regions $\mathbb{D}_k$ and $\mathbb{D}_i$ are adjacent, i.e., $i = k \pm 1$. There are two cases:

Case 1: $\mathrm{act}(k) \in I_{\geq,i}, \mathrm{act}(i) \in I_{\geq,k}$. Fig. B.1 illustrates this case. In Fig. B.1 (a), $k = i+1$, while in Fig. B.1(b), $i = k+1$. As $\mathrm{act}(i) \in I_{\geq,k}$, we have $\min_{j \in I_{\geq,k}} \{\ell_j\} \leq \ell_{\mathrm{act}(i)}$. Since $\min_{j \in I_{\geq,i}} \{\ell_j\} = \ell_{\mathrm{act}(i)}$ in $\mathbb{D}_i$, (10) follows.

Case 2: $\mathrm{act}(k) \in I_{\leq,i}, \mathrm{act}(i) \in I_{\leq,k}$. Fig. B.2 illustrates this case. As $\mathrm{act}(k) \in I_{\leq,i}$, we have $\ell_{\mathrm{act}(k)} \leq \ell_{\mathrm{act}(i)}$ in $\mathbb{D}_i$. Then for all $x \in \mathbb{D}_i$,

$$\min_{j \in I_{\geq,k}} \{\ell_j(x)\} \leq \ell_{\mathrm{act}(k)}(x) \leq \ell_{\mathrm{act}(i)}(x) = \min_{j \in I_{\geq,i}} \{\ell_j(x)\},$$

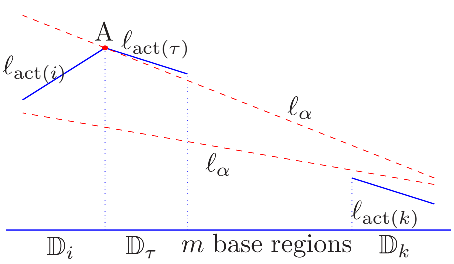
and so (10) is valid.

Induction: Assume (10) is valid when there are m base regions between $\mathbb{D}_k$ and $\mathbb{D}_\tau$, i.e., $\tau = k \pm m$. Then

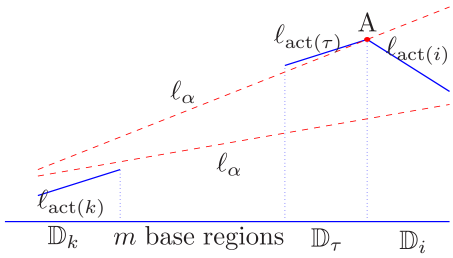
$$\min_{j \in I_{\geq,k}} \{\ell_j(x)\} \leq \min_{j \in I_{\geq,\tau}} \{\ell_j(x)\}, \forall x \in \mathbb{D}_\tau. \tag{B.1}$$

We show that (10) is true when there are $m+1$ base regions between $\mathbb{D}_k$ and $\mathbb{D}_i$, i.e., $i = k \pm (m+1)$. Fig. B.3 and B.4 sketches the relative positions of the affine functions $\ell_{\mathrm{act}(i)}$, $\ell_{\mathrm{act}(\tau)}$, and $\ell_{\mathrm{act}(k)}$. Since the assumption (B.1) holds, then there must exist an index $\alpha \in I_{\geq,k}$ such that

$$\ell_\alpha(x) \leq \ell_{\mathrm{act}(\tau)}(x), \; \forall x \in \mathbb{D}_\tau.$$

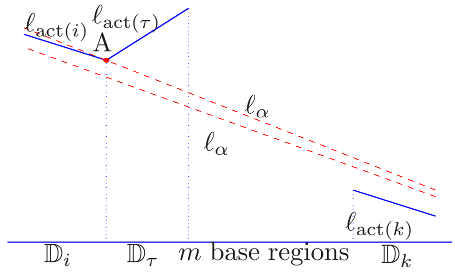


(a) $\tau = k - m, i = k - m - 1$

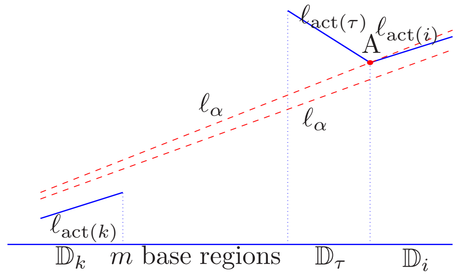


(b) $\tau = k + m, i = k + m + 1$

Fig. B.3. Sketches of 1-dimensional affine functions $\ell_{\text{act}(i)}$, $\ell_{\text{act}(\tau)}$, and $\ell_{\text{act}(k)}$ when $\text{act}(i) \in I_{\geq,\tau}$ and $\text{act}(\tau) \in I_{\geq,i}$.



(a) $\tau = k - m, i = k - m - 1$



(b) $\tau = k + m, i = k + m + 1$

Fig. B.4. Sketches of 1-dimensional affine functions $\ell_{\text{act}(i)}$, $\ell_{\text{act}(\tau)}$ and $\ell_{\text{act}(k)}$ when $\text{act}(i) \in I_{\leq,\tau}$ and $\text{act}(\tau) \in I_{\leq,i}$.

According to (3), in the base region $\mathbb{D}_i$, either $\ell_\alpha \leq \ell_{\text{act}(i)}$ or $\ell_\alpha \geq \ell_{\text{act}(i)}$. The sketch of possible position of ℓ_α is also shown in Fig. B.3 and Fig. B.4.

If $\ell_\alpha \leq \ell_{\text{act}(i)}$ in $\mathbb{D}_i$, we have (10).

Else if $\ell_\alpha \geq \ell_{\text{act}(i)}$ in $\mathbb{D}_i$, as $\ell_\alpha \leq \ell_{\text{act}(\tau)}$ in $\mathbb{D}_\tau$, the affine functions $\ell_{\text{act}(i)}$, $\ell_{\text{act}(\tau)}$, and ℓ_α must intersect at the same point, indicated by "A" in Fig. B.3 and B.4. Then we have $\ell_{\text{act}(i)} \geq \ell_\alpha$ in $\mathbb{D}_k$, as $\alpha \in I_{\geq,k}$, thus $\text{act}(i) \in I_{\geq,k}$. Therefore

$$\min_{j \in I_{\geq,k}} \{\ell_j(x)\} \leq \ell_{\text{act}(i)}(x), \forall x \in \mathbb{D}_i$$

and (10) holds.

References

Baotic, M., Borrelli, F., Bemporad, A., Morari, M., 2008. Efficient on-line computation of constrained optimal control. SIAM Journal on Control and Optimization 47 (5), 2470–2489.

Bartels, S. G., Kuntz, L., Scholtes, S., 1995. Continuous selections of linear functions and nonsmooth critical point theory. Nonlinear Analysis: Theory, Methods & Applications 24 (3), 385–407.

Bayat, F., Johansen, T. A., Jalali, A., 2012. Flexible piecewise function evaluation methods based on truncated binary search trees and lattice representation in explicit MPC. IEEE Transactions on Control Systems Technology 20 (3), 632–640.

Bemporad, A., Borrelli, F., Morari, M., 2002a. Model predictive control based on linear programming–the explicit solution. IEEE Transactions on Automatic Control 47 (12), 1974–1985.

Bemporad, A., Morari, M., Dua, V., Pistikopoulos, E. N., 2002b. The explicit linar quadratic regulator for constrained systems. Automatica 38 (1), 3–20.

Borrelli, F., 2003. Constrained Optimal Control of Linear and Hybrid Systems. Vol. 290. New York: Springer.

Breiman, L., 1993. Hinging hyperplanes for regression, classification and function approximation. IEEE Transactions on Information Theory 39 (3), 999–1013.

Christophersen, F. J., Kvasnica, M., Jones, C. N., Morari, M., 2007. Efficient evaluation of piecewise control laws defined over a large number of polyhedra. In: Proceedings of the European Control Conference. pp. 2360–2367.

Chua, L. O., Deng, A. C., 1988. Canonical piecewise-linear representation. IEEE Transactions on Circuits and Systems 35 (1), 101–111.

De Schutter, B., van den Boom, T. J. J., 2004. MPC for continuous piecewise-affine systems. Systems & Control Letters 52 (3), 179–192.

Gonzaga, C. C., 1995. On the complexity of linear programming. Resenhas IME-USP 2 (2), 197–207.

Gunawardena, J., 1994. Min-max functions. Discrete Event Dynamic Systems: Theory and Applications 4 (4), 377–407.

Herceg, M., Kvasnica, M., Jones, C. N., Morari, M., 2013a. Multi-parametric toolbox 3.0. In: Proceedings of the European Control Conference. pp. 502–510, http://control.ee.ethz.ch/~mpt.

Herceg, M., Mariéthoz, S., Morari, M., 2013b. Evaluation of piecewise affine control law via graph traversal. In: Proceedings of the European Control Conference. pp. 3083–3088.

Jones, C. N., Grieder, P., Raković, S. V., 2006. A logarithmic-time solution to the point location problem for parametric linear programming. Automatica 42 (12), 2215–2218.

Julián, P., Desages, A., Agamennoni, O., 1999. High-level canonical piecewise linear representation using a simplicial partition. IEEE Transactions on Circuits and Systems-I: Fundamental Theory and Applications 46 (4), 463–480.

Ovchinnikov, S., 2002. Max-min representation of piecewise linear functions. Contributions to Algebra and Geometry 43 (1), 297–302.

Ovchinnikov, S., 2010. Discrete piecewise linear functions. European Journal of Combinatorics 31 (5), 1283–1294.

Tarela, J. M., Martinez, M. V., 1999. Region configurations for realizability of lattice piecewise-linear models. Mathematical and Computer Modelling 30 (11), 17–27.

Tøndel, P., Johansen, T. A., Bemporad, A., 2003a. An algorithm for multi-parametric quadratic programming and explicit MPC solutions. Automatica 39 (3), 489–497.

Tøndel, P., Johansen, T. A., Bemporad, A., 2003b. Evaluation of piecewise affine control via binary search tree. Automatica 39 (5), 945–950.

Wang, S., Huang, X., Junaid, K. M., 2008. Configuration of continuous piecewise-linear neural networks. IEEE Transactions on Neural Networks 19 (8), 1431–1445.

Wang, S., Sun, X., 2005. Generalization of hinging hyperplanes. IEEE Transactions on Information Theory 12 (51), 4425–4431.

Wen, C., Ma, X., Ydstie, B. E., 2009. Analytical expression of explicit MPC solution via lattice piecewise-affine function. Automatica 45 (4), 910–917.

Wilkinson, R. H., 1963. A method of generating functions of several variables using analog diode logic. IEEE Transactions on Electronic Computers 12 (2), 112–129.

Xu, J., Huang, X., Wang, S., 2009. Adaptive hinging hyperplanes and its applications in dynamic system identification. Automatica 45 (10), 2325–2332.

Xu, J., van den Boom, T. J. J., De Schutter, B., 2014. Irredundant lattice piecewise affine representations and their applications in explicit model predictive control. In: Proceedings of the 53rd IEEE Conference on Decision and Control (CDC). pp. 4416–4421.

Xu, J., van den Boom, T. J. J., De Schutter, B., Luo, X. L., 2016. Minimal conjunctive normal expression of continuous piecewise affine functions. IEEE Transactions on Automatic Control, in print.

Ziegler, G., 1995. Lectures on Polytopes. Springer.