

Intersection-based Piecewise Affine Approximation of Nonlinear Systems

Amin Zavieh, and Luis Rodrigues

Abstract—This paper presents a new algorithm for PWA approximation of nonlinear systems. Such an approximation is very important to enable a reduction in the complexity of models of nonlinear systems while keeping the global validity of the models. The paper builds on previous work on piecewise affine (PWA) approximation methods, in particular on the work done by Casselman and Rodrigues, known as the Set of Linearization Points (SLP) PWA approximation. The proposed extension method can be used to approximate any continuous function of one variable by a PWA function. The algorithm is based on the points at which the linearization lines intersect with each other. The method assumes that a desired approximation error and one linearization point are given. The algorithm, then performs several linearizations. It is shown that the new linearization points are optimal in the sense of decreasing the error between the exact function and the approximation. The main advantages of this methodology compared to previous approaches are the reduction of the number of pieces of the PWA function, the guarantee that the approximation is continuous, and that the derivative of the approximation and the derivative of the exact function are equal at all linearization points. A detailed collection of examples from different fields of study highlight the effectiveness and the flexibility of the proposed method. It is shown that the proposed method compares favorably with other methods.

Index Terms—Intersection-based, piecewise affine, continuous functions approximation, functions of one variable

I. INTRODUCTION

Piecewise affine (PWA) systems have shown to be a powerful approach in analysis and synthesis of nonlinear systems [1], [2], [3]. The key concept behind this idea is that the nonlinearities appearing in a dynamical system can be reasonably approximated by PWA functions. This paper builds on previous work on approximation of functions of one variable using a piecewise affine method [4], emphasizing the elements that increase the effectiveness of the approximation algorithm in terms of elimination of the need to search for points of maximum error and decreasing the error between the exact function and the approximation.

Although the concept of PWA systems was initially researched in the late 1940's, the first optimal algorithm to approximate nonlinearities with PWA functions appears, to the best of our knowledge, in 1970's by Cantoni [5] and Tomek [6]. Many different attempts have been made to produce suitable PWA models in references [7], [1], [8], [9], and [4]. Reference [7] addresses PWA approximation of continuous functions using uniform simplicial partitions. Later in [1], the idea of refinement of the partitions around the origin is

introduced. The least squares technique as an optimization over simplicial partitions is addressed in [8]. In references [7] and [1], the domain of the nonlinearity was uniformly divided into a number of simplices. A point in each simplicial region was then selected for the linearization of the function.

The main disadvantage of these methods lies in the fact that for a function of one variable the number of regions uniformly grows as the domain of approximation is increased. In other words, the behavior of the nonlinearity is not considered while the domain of the function is partitioned into regions. This drawback can be avoided if the curvature or the variation of the nonlinear function is considered, as done recently in [9] and [4]. Reference [9] addresses a novel methodology in PWA approximation of functions that uses the concept of Lebesgue integration partitioning. However, the resulting PWA approximation is not guaranteed to be continuous. Uniform grid approximation techniques have been extensively used in the literature [10], [11], [12], [13], [14], [15], [16]. The authors of [4] address the approximation problem by considering the curvature of the function. Moreover the continuity problem has been solved in their work, which, nevertheless, has never been proved. Reference [4] provides the reader with an interesting and heuristic idea of their work to find the PWA model of a micro air vehicle. Although the proposed concept is shown to be efficient compared to the work done in this area, the details of the approximation and the supporting theory is missing, which is a reason for motivating the work on the current paper.

The method proposed in this paper is used to find a PWA model of four case studies, which are a unicycle vehicle following a straight line, the nonlinear aeroelastic model of an aircraft wing studied for the flutter phenomenon, a two-gene regulatory network, and a mathematical example. The resulting PWA models can later be used for the analysis and/or synthesis problems that have been addressed in reference [3]. Moreover, it is suggested in [9] that the resulting PWA model may be used for piecewise affine identification with a clustering approach [17], mixed logical dynamics (MLD) model based technique [18], system verification of conflict maneuvers [19], automated symbolic reachability analysis [20], and probabilistic controllability/observability analysis of discrete-time piecewise affine systems [21], [22].

With the intersection-based Piecewise Affine (IPWA) models, the following properties can be achieved.

- Continuity of the vector fields,
- Optimality of the linearization of the nonlinear function relative to the maximum approximation error,
- Increased reduction of the approximation error for a fixed number of regions (as compared to the Lebesgue and

The authors are with the Department of Electrical and Computer Engineering, Concordia University, Montreal, QC, H3G 2W1 Canada e-mails: {s_seyeda, luisrod}@encs.concordia.ca

uniform grid PWA models for all examples in Section V),

- Consistency of the derivative of the nonlinear function with the derivative of its approximation at the linearization points.

The continuity of the vector field may play a crucial role in controller synthesis for PWA systems [23]. By the optimality of the linearization we mean that the nonlinear function is linearized at the points of maximum approximation error. By doing so, not only the number of approximation stages is reduced, but also the number of regions is decreased. Note that the smaller the number of regions we have in the PWA model, the more the computation size of the controller synthesis problem is reduced [3]. Finally, if the user is required to have zero error at specific points as well as minimum amount of error in the neighborhood of those points, the IPWA method can serve as a good solution to such problems. The reason, as will be shown in Section IV, is that the algorithm starts by linearizing the function at a set of user-defined points. Since the function is linearized at those points, the approximation error is zero at them. Moreover, because the derivative of the exact and the IPWA function are equal, the error in the neighborhood of the user-defined points is small.

The paper is organized as follows. In Section II a list of all variables with definitions are given. In addition to the main results, Section III briefly explains PWA systems. Section IV addresses the approximation algorithms. Finally, the proposed approximation method is applied to the mentioned dynamical system in Section V, followed by the conclusions.

II. NOMENCLATURE

Scripts and Operations

Ω^o	Interior of the set Ω
$\overline{\Omega}$	Closure of the set Ω
x^+	$= x + \varepsilon$, where $\varepsilon \in \mathbb{R}^+$ is infinitesimally small
x^-	$= x - \varepsilon$
$\mathcal{S}[f(x)]$	$= \{(x, y) \mid y = f(x)\}$
$\mathcal{S}[f(x)] \cap \mathcal{S}[g(x)]$	$= \{(x, y) \mid y = g(x), y = f(x)\}$
$\partial\Omega$	$= \overline{\Omega} \setminus \Omega^o$

III. PWA SYSTEMS AND PWA APPROXIMATIONS

Consider the state space representation of a dynamical system, as

$$\dot{x} = Ax + f_{nl}(x) + Bu \quad (1)$$

where $A \in \mathbb{R}^{n \times n}$, $x \in \mathbb{R}^n$ is the state vector, $B \in \mathbb{R}^{n \times m}$, $u \in \mathbb{R}^m$ is the control input, and f_{nl} is a nonlinear continuous function defined as $f_{nl} : \Omega \rightarrow \mathbb{R}^n$, where $\Omega \subset \mathbb{R}^n$. By computing a PWA approximation for (1), Ω is partitioned into a finite number of regions, in each of which an affine function serves as the approximation of f_{nl} . Furthermore, \mathcal{R}_i denotes the i^{th} PWA region, $i \in \mathcal{S} = \{1, 2, \dots, N\}$ such that $\bigcup_{i=1}^N \overline{\mathcal{R}_i} = \Omega$. In what follows, the next concepts will be used.

Definition 1: The function $\tilde{f} : \Omega \rightarrow \mathbb{R}^n$, where $\Omega \subset \mathbb{R}^n$, is defined to be the PWA approximation of the nonlinear function f_{nl} , given by

$$\tilde{f}(x) = A_i x + b_i, \quad x \in \mathcal{R}_i \quad (2)$$

where $i \in \mathcal{S} = \{1, 2, \dots, N\}$ is the index indicating the region. Note that by replacing $f_{nl}(x)$ in (1) by $\tilde{f}(x)$ one obtains the PWA model of the nonlinear system (1), as

$$\dot{x} = (A + A_i)x + b_i + Bu, \quad x \in \mathcal{R}_i \quad (3)$$

Moreover, the name *Piecewise Affine (PWA)* covers the general concept of the systems modeled by (3) while the name *Intersection-based Piecewise Affine (IPWA)* as a subset of *PWA systems*, refers to the systems, the coefficients A_i and b_i of which is obtained by the proposed methodology.

To formalize the general notion we begin by giving the main idea of the algorithm. The first stage of the IPWA approximation is performed by linearizing the nonlinear function around specific points which are given by the user. If, for instance, the task is to solve a PWA controller synthesis problem for the obtained IPWA system, these points can be chosen to be the equilibrium points of the nonlinear system. Therefore, the selection of such points varies depending on the nature of the problem. Each linearization will be called tangent hyperplanes. The regions are created by projecting the intersection points of the hyperplanes onto the domain of the nonlinearity. In the next approximation stages, the function will be linearized at the intersection points obtained in the previous stage. This process is continued until the desired error is met.

As mentioned in equation (1), the nonlinear component of a dynamical system is described by $f_{nl}(x) = [f_1(x), f_2(x), \dots, f_n(x)]^T$. If $f_{nl}(x)$ is a function of only one variable, say x_j with j being a fixed number in $\{1, 2, \dots, n\}$, satisfying the conditions given later in Theorem 1, the proposed IPWA algorithm for continuous functions can be used to construct the IPWA model. For this purpose $\tilde{f}(x)$ is first obtained as in (2) for all $f_q(x_j)$, where $q = \{1, 2, \dots, n\}$. This approximation will later be added to the linear components, as in equation (3). By doing so, the PWA regions produced during any linearization stage will take the form of

$$\mathcal{R}_i = \{x \in \mathbb{R} \mid d_i < x < d_{i+1}\}, \quad (4)$$

where

$$d = [x_1, x_{int}^{(1)}, \dots, x_{int}^{(k)}, x_2], \quad (5)$$

and $x_{int}^{(k)}$ is the projection of the intersection of the hyperplanes onto Ω with k referring to the number of intersections in each epoch. This type of regions that can be defined with only one variable are called *slabs*. PWA systems with slab regions are thus called PWA slab systems. Accordingly, as Rodrigues and Boyd [3] have shown, for PWA slab systems the state feedback controller synthesis with a quadratic Lyapunov function can be formalized as a convex optimization problem subject to a specific set of LMIs. Although the solution to the synthesis problem is not addressed in this paper, the resulting approximation of the nonlinear systems introduced in the paper can be used to design a PWA controller with the material given in [3].

Definition 2: Consider a nonlinear function $f : \Omega \rightarrow \mathbb{R}$, where $\Omega \subset \mathbb{R}^n$. A linearization hyperplane $h(x)$ is defined to be

$$h(x) = h_i(x), \quad x \in \mathcal{R}_i, \quad (6)$$

where

$$h_i(x) = f(x_0^i) + \nabla f(x_0^i)(x - x_0^i), \quad (7)$$

and $x_0^i \in \mathcal{R}_i$ is a nominal point.

Definition 3: The distance function $\Delta : \Omega \rightarrow \mathbb{R}$ is defined as

$$\Delta(x) = \Delta_i(x), \quad x \in \mathcal{R}_i, \quad i \in \mathcal{I} \quad (8)$$

where

$$\Delta_i(x) = \begin{cases} \|h_i(x) - f(x)\| & , \quad h_i(x) < f(x), \\ h_i(x) - f(x) & , \quad h_i(x) \geq f(x). \end{cases} \quad (9)$$

Henceforth, the Greek letter Γ is used as the domain of a function if $n = 1$ while Ω is used generally for n not being specified.

Lemma 1: Consider a concave function $f : \Gamma \rightarrow \mathbb{R}$, $\Gamma \subset \mathbb{R}$. Suppose that f is class C^1 in a neighborhood of two distinct points $\{x_1, x_2 \mid x_1 < x_2\} \subset \Gamma^\circ$. Then

$$f'(x_2) \leq f'(x_1), \quad \forall x_1 \leq x_2 \quad (10)$$

Note that a lemma with a similar result to Lemma 1 is given in [24] (Chapter 5, Section 5, Lemma 15).

Proof: Let $x_b \in \Gamma^\circ$ be an arbitrary point. Note that f as a concave function is absolutely continuous in $\Upsilon = [x_b - \varepsilon, x_b + \varepsilon] \subset \Gamma$, for $\varepsilon > 0$ sufficiently small [24]. Using this property, by the same reference $f'(x)$ almost exists for $x \in \Upsilon$. Therefore, if $f'(x_b)$ does not exist, one concludes $f'(x_b - \varepsilon)$ and $f'(x_b + \varepsilon)$ exist, or simply $\exists \{f'(x_b^-), f'(x_b^+)\}$. As a result, for $x_b^- < x_b$ a tangent hyperplane $\bar{h}(x)$ can be constructed as

$$\bar{h}(x) = f(x_b^-) + f'(x_b^-)(x - x_b^-) \quad (11)$$

Remarking that

$$\begin{aligned} \lim_{\varepsilon \rightarrow 0} x_b^- &= \lim_{\varepsilon \rightarrow 0} x_b - \varepsilon = x_b, \\ \lim_{\varepsilon \rightarrow 0} f(x_b^-) &= \lim_{\varepsilon \rightarrow 0} f(x_b - \varepsilon) = f(x_b), \end{aligned}$$

we may rewrite equation (11), as

$$\bar{h}(x)|_{\varepsilon \rightarrow 0} = f(x_b) + f'(x_b^-)(x - x_b). \quad (12)$$

Function $f(x)$ can be approximated at $x_b^+ > x_b$ using Taylor series as

$$f(x)|_{x_b^+} \approx f(x_b^+) + f'(x_b^+)(x - x_b^+) \quad (13)$$

Since

$$\begin{aligned} \lim_{\varepsilon \rightarrow 0} x_b^+ &= \lim_{\varepsilon \rightarrow 0} x_b + \varepsilon = x_b, \\ \lim_{\varepsilon \rightarrow 0} f(x_b^+) &= \lim_{\varepsilon \rightarrow 0} f(x_b + \varepsilon) = f(x_b), \end{aligned}$$

using (13), one may write

$$f(x)|_{x_b^+} \approx f(x_b) + f'(x_b^+)(x - x_b) \quad (14)$$

Note that the function f is concave, and by the supporting hyperplane theorem [25]

$$f(x) \leq \bar{h}(x), \quad \forall x \in \Gamma. \quad (15)$$

Let $x = x_b + \delta$, where $\delta > 0$ is an infinitesimally small number. Accordingly, using (14) to approximate $f(x_b + \delta)$ and (12) to evaluate $\bar{h}(x_b + \delta)$, inequality (15) becomes

$$\begin{aligned} f(x_b) + f'(x_b^+)(x_b + \delta - x_b) &\leq \\ f(x_b) + f'(x_b^-)(x_b + \delta - x_b) & \end{aligned}$$

Simplifying the left and right hand side terms yields in

$$f'(x_b^+) \leq f'(x_b^-) \quad (16)$$

(16) in a sequence including f' implies that (10) is held, no matter f is differentiable at x_b or not. ■

Lemma 2: Considering the concave function f and $\{x_1, x_2\}$ as assumed in Lemma 1, let the function f be linearized around x_1 and x_2 with lines $h_1(x)$ and $h_2(x)$ respectively. Assuming $h_1'(x) \neq h_2'(x)$, the intersection of $h_1(x)$ and $h_2(x)$ is a singleton, i.e.,

$$\{x_{int}\} = \{x \mid h_1(x) = h_2(x)\} \quad (17)$$

Furthermore, $x_1 \leq x_{int} \leq x_2$.

Proof: From the assumption, since $\{x_1, x_2\} \subset \Gamma^\circ$ and the fact that f is concave and locally of class C^1 around x_1 and x_2 , we have $h_1'(x) = f'(x_1) \neq \infty$ and $h_2'(x) = f'(x_2) \neq \infty$. Moreover, $h_1'(x) \neq h_2'(x)$ implying that h_1 and h_2 are not parallel, they intersect with each other at some point equal to x_{int} . To prove that x_{int} is the unique solution as the intersection of h_1 and h_2 , one is required to show that $h_1(x) \neq h_2(x)$ for $x \neq x_{int}$. $h_1(x)$ and $h_2(x)$ are determined by the linearization of f around x_1 and x_2 , as

$$h_1(x) = f(x_1) + f'(x_1)(x - x_1), \quad (18a)$$

$$h_2(x) = f(x_2) + f'(x_2)(x - x_2), \quad (18b)$$

Using contradiction, let $h_1(x) = h_2(x)$ for $x \neq x_{int}$. This results in

$$x = \frac{[f(x_1) - f(x_2)] + [f'(x_2)x_2 - f'(x_1)x_1]}{f'(x_2) - f'(x_1)} \quad (19)$$

Since from the assumption of the lemma $h_1'(x) \neq h_2'(x)$ and thus $f'(x_1) \neq f'(x_2)$ as well as $x_1 \neq x_2$, it is concluded that equation (19) provides a unique value for x being $x = x_{int}$. Consequently $h_1(x) = h_2(x)$ only for $x = x_{int}$ or in other words $h_1(x) \neq h_2(x)$ for $x \neq x_{int}$, which implies that x_{int} is a singleton.

By the supporting hyperplane theorem [25]

$$h_1(x) \geq f(x) \quad (20a)$$

$$h_2(x) \geq f(x) \quad (20b)$$

From this point, the proof of $x_1 \leq x_{int} \leq x_2$ follows by contradiction. First, let $x_2 < x_{int}$. Since h_1 and h_2 intersect with each other, we have

$$h_1(x_{int}) = h_2(x_{int})$$

Using (18), this implies that

$$f(x_2) + f'(x_2)(x_{int} - x_2) = f(x_1) + f'(x_1)(x_{int} - x_1)$$

or alternatively, this in turn implies

$$f(x_2) - f(x_1) - f'(x_2)x_2 + f'(x_1)x_1 = -x_{int}(f'(x_2) - f'(x_1)) \quad (21)$$

On the other hand

$$\begin{aligned} h_2(x) - h_1(x) &= f(x_2) + f'(x_2)(x - x_2) - f(x_1) - f'(x_1)(x - x_1) \\ &= [f(x_2) - f(x_1) - f'(x_2)x_2 + f'(x_1)x_1] \\ &\quad + x(f'(x_2) - f'(x_1)) \end{aligned} \quad (22)$$

Incorporating (21) within (22) yields

$$\begin{aligned} h_2(x) - h_1(x) &= -x_{int}(f'(x_2) - f'(x_1)) + x(f'(x_2) - f'(x_1)) \\ &= (f'(x_2) - f'(x_1))(x - x_{int}) \end{aligned} \quad (23)$$

Finally, using (10) as the result of Lemma 1, it is concluded from (23) that

$$h_2(x) \geq h_1(x), \quad \forall x < x_{int} \quad (24)$$

It was assumed that $x_2 < x_{int}$ which, using (24), means that

$$h_2(x_2) \geq h_1(x_2), \quad x_2 < x_{int} \quad (25)$$

By substituting x_2 into (18b) we have

$$h_2(x_2) = f(x_2) \quad (26)$$

Combining (25) and (26) results in the following condition

$$f(x_2) \geq h_1(x_2), \quad \forall x < x_{int}$$

by which the supporting hyperplane theorem (20a) is violated, implying that $x_{int} \leq x_2$. Similarly, the same argument can be used to show that $x_{int} \geq x_1$, proving that $x_1 \leq x_{int} \leq x_2$. ■

Lemma 3: Let $f : \Gamma \rightarrow \mathbb{R}$, $\Gamma \subset \mathbb{R}$ be a class C^1 concave function. Consider an arbitrary point $x_0 \in \Gamma$. Assume that the linearization of f around the point x_0 is described by a hyperplane $h(x)$. Then for any direction pointing from x_0 , $\Delta(x)$ is monotonically increasing, i. e.

$$\nabla\Delta(x) \cdot \mathbf{v}_{x_0}(x) \geq 0, \quad x \in \Gamma, \quad (27)$$

where

$$\mathbf{v}_{x_0}(x) = \frac{x - x_0}{\|x - x_0\|}, \quad x \in \Gamma \quad (28)$$

is a unit vector.

Proof: f is concave, therefore $-f$ is convex. The function h is also convex, because it is affine. Since the convexity property is kept under addition of convex functions [25], $\Delta(x) = h(x) - f(x)$ is convex.

Since h is the linearization of f at x_0 and f is concave, we have $h \geq f$, according to the supporting hyperplane theorem [25]. This results in $\Delta(x) \geq 0$. In addition, because $f(x_0) = h(x_0)$ implies $\Delta(x_0) = 0$, we can conclude that

$$x_0 = \underset{x \in \Gamma}{\operatorname{argmin}} \Delta(x)$$

as the point of global minimum of $\Delta(x)$. Using the first order necessary condition for the points of minimum [25], we have

$$\nabla\Delta(x_0) = 0. \quad (29)$$

Furthermore, noting that $-\Delta(x)$ is concave, Lemma 1 is used to say that $\Delta'(x)$ is either increasing or remains stationary as

x is increased. This, in view of (29), leads to the conclusion that

$$\begin{aligned} \nabla\Delta(x) &\leq 0, \quad \forall x < x_0, \\ \nabla\Delta(x) &\geq 0, \quad \forall x > x_0, \end{aligned}$$

which, using the definition of \mathbf{v}_{x_0} in (28), can be rewritten in a more compact form, as

$$\nabla\Delta(x) \cdot \mathbf{v}_{x_0}(x) \geq 0, \quad x \in \Gamma. \quad \blacksquare$$

In Lemma 2 an intersection property of concave functions was explained. As it can be inferred from the proof of Lemma 3, $\Delta(x) \geq 0$. Therefore, with the results of Lemma 3 and Lemma 2, at this stage, Theorem 1 is given to find the points of maximum error in Γ , defined as $e_{max} = \sup_{x \in \Gamma} \Delta(x)$.

Theorem 1: Consider a concave function $f : \Gamma \rightarrow \mathbb{R}$, $\Gamma \subset \mathbb{R}$ of class C^1 . Let the function be linearized around two points $x_1, x_2 \in \Gamma$ with lines $h_1(x)$ and $h_2(x)$ respectively. Furthermore, consider the point $P_{int} = (x_{int}, y_{int}) = \{(x, y) \mid (x, y) \in S[h_1(x)] \cap S[h_2(x)]\}$. Then, the solution to the following maximization problem

$$\sup_{x \in \Gamma} \Delta(x) \quad (30)$$

lies either on x_{int} or at $\partial\Gamma = \{q_1, q_2\}$.

Proof: It can be recognized by Lemma 2 that $x_1 \leq x_{int} \leq x_2$. Furthermore $x_1, x_2 \in \Gamma$, which implies that x_{int} divides Γ into two sub-domains, each of which contains one of $\{x_1, x_2\}$. Let us denote the sub-domain containing x_1 by Γ_{x_1} and the other one by Γ_{x_2} . In order to prove this theorem, Γ will be split into 4 different sub-domains, namely $\Gamma_2, \Gamma_1, \Gamma_3$ and Γ_4 , as

$$\begin{aligned} \Gamma_2 &= \operatorname{conv}(x_1, x_{int}), & \Gamma_1 &= \Gamma_{x_1} \setminus \Gamma_2 \\ \Gamma_3 &= \operatorname{conv}(x_2, x_{int}), & \Gamma_4 &= \Gamma_{x_2} \setminus \Gamma_3 \end{aligned} \quad (31)$$

where $\operatorname{conv}(\cdot)$ denotes the convex hull of the points in the argument. Regarding the conditions of Lemma 3, the distance function $\Delta(x)$, from equation (8), can be represented as

$$\Delta(x) = \begin{cases} \Delta_1(x) = h_1(x) - f(x) & , \quad x \in \Gamma_{x_1} \\ \Delta_2(x) = h_2(x) - f(x) & , \quad x \in \Gamma_{x_2} \end{cases} \quad (32)$$

In view of the fact that \mathbf{v}_{x_1} and \mathbf{v}_{x_2} are determined using

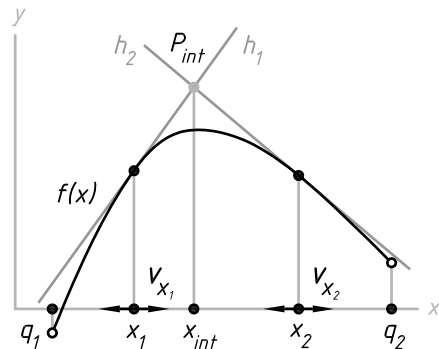


Fig. 1. The intersection of h_1 and h_2 at P_{int} .

(28), as unit direction vectors at x_1 and x_2 respectively, the following 2 cases are investigated.

- 1) $x \in \Gamma_2 \cup \Gamma_3$
- 2) $x \in \Gamma_1 \cup \Gamma_4$

CASE 1. First, we consider $x \in \Gamma_2$. Let $x_{\Gamma_2}(p) = x_1 + \varepsilon p$, where $p \in \{p \in \mathbb{Z} > 0 \mid p < p_{max^2}\}$ with p_{max^2} defined such that $x_{int} = x_1 + \varepsilon p_{max^2}$. Using \mathbf{v}_{x_1} as a function of x , we have $\nabla \Delta_1(x) \cdot \mathbf{v}_{x_1}(x_{\Gamma_2}) \geq 0$ by Lemma 3. Therefore, for ε sufficiently small, $\Delta_1(x_1) \leq \Delta_1(x_1 + \varepsilon)$, which is equivalent to $\Delta_1(x_{\Gamma_2}(0)) \leq \Delta_1(x_{\Gamma_2}(1))$. Similarly, we have $\Delta_1(x_{\Gamma_2}(1)) \leq \Delta_1(x_{\Gamma_2}(2))$, which using a sequence including $0 \leq p \leq p_{max^2}$ we reach the conclusion that $\Delta_1(x_{\Gamma_2}(p)) \leq \Delta_1(x_{\Gamma_2}(p_{max^2}))$. In another word

$$\Delta_1(x) \leq \Delta_1(x_{int}), \quad \forall x \in \Gamma_1. \quad (33)$$

Likewise, defining $x_{\Gamma_3}(p) = x_2 - \varepsilon p$ where $p \in \{p \in \mathbb{Z} > 0 \mid p < p_{max^3}\}$ with p_{max^3} defined as $x_{int} = x_2 - \varepsilon p_{max^3}$. Using the same idea, one can conclude that

$$\Delta_2(x) \leq \Delta_2(x_{int}), \quad \forall x \in \Gamma_2. \quad (34)$$

CASE 2. In the second case, let $x \in \Gamma_1$. using Lemma 3 we have $\nabla \Delta_1(x) \cdot \mathbf{v}_{x_1}(x_{\Gamma_1}) \leq 0$. Considering $x_{\Gamma_1}(p) = x_1 - \varepsilon p$ where $p \in \{p \in \mathbb{Z} > 0 \mid p < p_{max^1}\}$ with p_{max^1} defined as $q_1 = x_1 - \varepsilon p_{max^1}$, we may write $\Delta_1(x_{\Gamma_1}(0)) \leq \Delta_1(x_{\Gamma_1}(1))$. Using a sequence, similar to the argument in Case 1, we have

$$\Delta_1(x) \leq \Delta_1(q_1), \quad \forall x \in \Gamma_3. \quad (35)$$

Obviously, for $x \in \Gamma_4$, one can easily check that

$$\Delta_2(x) \leq \Delta_2(q_2), \quad \forall x \in \Gamma_4. \quad (36)$$

Equations (33), (34), (35) and (36) guarantee that an element from the set $\{x_{int}, q_1, q_2\}$ is indeed a solution to (30). ■

The objective at each stage is to linearize the function at the points of maximum error occurring during the linearization. Theorem 1 ensures that after the first stage of approximation, the solution to (30) lies either on x_{int} , or q_1 , or q_2 . If the first linearization occurs very close to q_1 and q_2 such that $\Delta(q_1) \leq e_{des}$ and $\Delta(q_2) \leq e_{des}$, the solution to (30) will be x_{int} . Therefore, x_{int} is the next point at which f should be linearized (see the upper plot in Figure 2). Denoting this linearization by h_3 , the second approximation stage is started as the point x_{int} adopts a new notation x_3 (see the lower plot in Figure 2). Updating the distance function $\Delta(x)$ using (8), the problem of approximating f is formalized with two parts, as to find

$$1) \operatorname{argmax}_{x_1 \leq x \leq x_3} \Delta(x) \quad (37)$$

$$2) \operatorname{argmax}_{x_3 \leq x \leq x_2} \Delta(x), \quad (38)$$

which are the points of maximum error. Since f is concave within $x_1 \leq x \leq x_3$ and $x_3 \leq x \leq x_2$, Theorem 1 is then used to find the solution for (37) and (38). This process is continued as long as $\max_{x \in \Gamma} \Delta(x) > e_{des}$.

With all of the theoretical background that has been provided in this section, the algorithm for the intersection-based PWA approximation can be described in the next section.

IV. ALGORITHMS

In this section, the algorithms elaborating the intersection-based PWA (IPWA) approximation for both concave/convex and continuous functions are introduced.

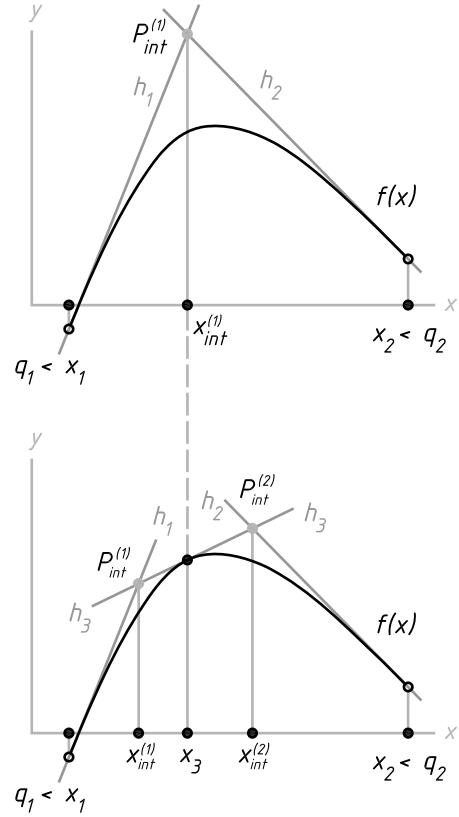


Fig. 2. The upper plot shows the primary epoch of the approximation while the lower plot provides the second epoch. Note that the superscripts reset at each epoch.

A. Concave/Convex Functions

This algorithm is proposed for a concave function. If f is convex, $-f$ is then considered.

Problem 1: Consider a concave function $f : \Gamma \rightarrow \mathbb{R}$, where $\Gamma \subset \mathbb{R}$. Find \bar{f} over the domain of f .

- 1) The primary linearization points are determined. The points are chosen at which the user needs zero error. As mentioned above, the primary linearization points are selected in accordance with the nature of the problem for which the IPWA being obtained is considered to serve. One can still add other initial points heuristically, like the points of zero curvature. This idea was originally suggested by Casselman and Rodrigues in reference [26], as the essential idea of the SLP method. Although the idea seems to improve the approximation, Example 4 shows a purely mathematical equation where setting the zero curvature point as a primary linearization point does not necessarily reduce the error.
- 2) Using Taylor series, the function f is linearized around all the points obtained in step 1. The produced linearization hyperplanes are denoted by $h_i(x)$, where $i \in \mathcal{I} = \{1, 2, \dots, N_u\}$ and N_u is the number of linearization points at the u^{th} loop.
- 3) Each two neighboring hyperplanes $h_i(x)$ will intersect with each other, resulting in a point. These points are designated by $P_{int}^{(k)}(x)$, where $k \in \mathcal{K} = \{1, 2, \dots, N_u^2\}$ and

N_u^2 is the number of the intersections in the u^{th} loop.

- 4) The projection of the all intersection points $P_{int}^{(k)}$ on Γ is denoted by $x_{int}^{(k)}$.
- 5) Using Theorem 1 the points at which the local maximum errors occur are determined. According to this theorem, these points belong to the set $M = \{x_1, x_{int}^{(1)}, \dots, x_{int}^{(k)}, x_2\}$. The global maximum is then obtained by updating $\Delta(x)$ using (8), and then evaluating $\Delta(x)$ for all the elements of M . Comparing the results, the value at which the global maximum error occurs and the error itself are found, as e_{max} is set as

$$\begin{aligned} x_{e_{max}} &= \operatorname{argmax}_{x \in M} \Delta(x) \\ e_{max} &= \max_{x \in M} \Delta(x) \end{aligned}$$

- 6) The stopping criterion is defined as

$$e_{max} \leq e_{des} \quad (39)$$

where e_{des} is called *desired error* and is defined to be the maximum allowable error determined by the user. Then e_{des} is compared with the value of e_{max} , obtained in step 5. If the stopping criterion is met, the IPWA approximation is produced. Otherwise, the point of maximum error is given to step 2 for the next linearization stage. This loop is continued until the stopping criterion is satisfied.

B. Continuous Functions

In this section, the function f is considered with only the continuity property. Despite the fact that such a function, that may not be concave nor convex, is rather difficult to tackle, many applications, including Examples 1 and 3, are neither concave nor convex. For this purpose, the IPWA algorithm for approximating continuous functions is given in this section. In this algorithm, the inflection points of $f(x)$ are found first. The curvature of f at these points attains zero and the convexity of f changes. The set of inflection points are denoted by $\mathcal{X} = \{x_k^1, x_k^2, \dots, x_k^q\}$. Therefore, function f can be split into $q+1$ functions, as $f_1(x), f_2(x), \dots, f_{q+1}(x)$, each being either convex or concave. For this aim, the initial linearization points are chosen to be the members of \mathcal{X} . The initial linearization hyperplanes $h_k^1, h_k^2, \dots, h_k^q$ are then constructed using the Taylor series. Since each $f_i(x)$ possesses either the concavity/convexity property, the IPWA algorithm for concave/convex functions is, therefore, used to compute the IPWA approximation for each $f_i(x)$.

The details of this method are summarized in the following algorithm.

Problem 2: Consider a continuous function $f : \Gamma \rightarrow \mathbb{R}$, where $\Gamma \subset \mathbb{R}$. Find \tilde{f} .

- 1) In the initial linearization stage, select the points from both categories:
 - a) members of set \mathcal{X} , and
 - b) the reference point (if the approximation purpose is to solve a control synthesis problem).
- 2) Repeat steps 2 to 6 from section IV-A for f .

The regions produced by both of the proposed algorithms are convex and have been defined in equation (4) in section III.

V. APPLICATIONS

In this section the IPWA approximation algorithm is applied to four examples. The obtained IPWA models of Examples 1 to 3 are compared with the Lebesgue and the uniform grid PWA approximation techniques while Example 4 is compared with the SLP algorithm proposed by [26]. The normalized approximation error \bar{e}_{max} that is used for the comparison purpose is computed by

$$\bar{e}_{max} = \frac{e_{max}}{\max_{x \in \Gamma} f(x) - \min_{x \in \Gamma} f(x)}, \quad (40)$$

where $f(x)$ is the exact function, and Γ is the domain of f .

Example 1: A wheeled mobile robot (WMR) with a rigid structure is considered for a path following problem. The idea is to have the rover followed the y axis with zero heading. The WMR moves with a constant velocity V_0 and is rotated by torque T exerted by an actuator, as shown in Figure 3.

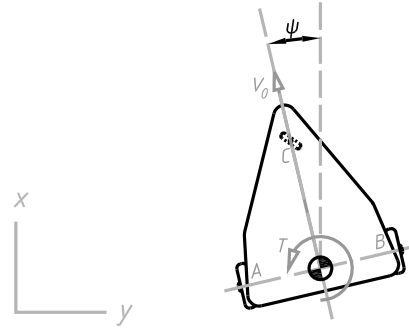


Fig. 3. Free body diagram of the WMR, where ψ is the heading, and V_0 is the velocity of CG.

For simplification, we have assumed that the center of gravity is located in the mid point of the vehicle axle. This assumption is quite reasonable since the driver motors, which make the major contribution to the vehicle weight, are usually mounted on the top of the axle position. Denoting $x_1 = y$, $x_2 = \psi$, $x_3 = \dot{\psi}$, the equations of motion for this problem can be written as

$$\dot{x}_1 = V_0 \sin x_2 \quad (41a)$$

$$\dot{x}_2 = x_3 \quad (41b)$$

$$\dot{x}_3 = I_z^{-1} T, \quad (41c)$$

where I_z is the WMR moment of inertia in the z axis. It is recognized that $f_{nl}(x) = [f_1(x_2), 0, 0]^T$, comparing (41) with (1). Therefore the IPWA algorithm can be used to obtain the IPWA approximation for $f_1(x_2) = V_0 \sin(x_2)$. The function $f_1(x_2)$ with its IPWA approximation is plotted in Figure 4. The upper plot is associated with the 2nd iteration while the lower plot indicates the 3rd iteration of the proposed algorithm. Due to the symmetry of the sine function, points $x_{int}^{(1)} = -2.136$, $x_{int}^{(2)} = -1.0$, $x_{int}^{(3)} = 1.0$ and $x_{int}^{(4)} = 2.136$ have the same value for the approximation error. Therefore, the

TABLE I
THE SUMMARY OF PIECEWISE AFFINE APPROXIMATION FOR
 $f_1(x_2) = \sin x_2$.

Normalized Approximation Error (\bar{e}_{max}) of $f_1(x_2)$			
NUMBER OF REGIONS n	IPWA APPROX.	LEBESGUE PWA APPROX.	UG PWA APPROX.
3	0.2854	0.3536	0.2182
4	NA	0.1036	0.0668
5	0.0793	NA	0.0909
7	NA	0.0849	0.0479
8	NA	0.0581	0.0244
9	0.0229	NA	0.0292
11	NA	0.0504	0.0140
12	NA	0.0398	0.0124
13	0.0191	NA	0.0144
15	NA	0.0359	0.0108
16	NA	0.0302	0.0074
17	0.0067	NA	0.0084
19	NA	0.0279	0.0065
20	NA	0.0244	0.0040
21	0.0054	NA	0.0056

function is linearized around all of the intersection points in the next loop of the algorithm. The summary of the approximation is found in Table I, where the proposed algorithm is compared to the uniform grid and the Lebesgue PWA approximation methods. As it can be seen, the approximation iteration is done until $\bar{e}_{max} = 0.0054$ with 21 regions. As mentioned earlier with the numerical values, the symmetry of the sine function for the IPWA algorithm may cause several points with the same value of error in each linearization stage, hence the number of regions may increase by more than 1 region per iteration. This is the reason we marked 'NA' in the second column of Table I. A similar phenomenon was encountered with the Lebesgue approximation applied to the sine function. Consequently, the comparison of each methodology can be accomplished using the numerical values obtained with the uniform grid technique. It can be seen that the \bar{e}_{max} of the uniform grid PWA model is less than the same parameter for the Lebesgue model in each epoch. On the other hand, smaller values for \bar{e}_{max} are obtained with the IPWA model compared to the uniform grid model.

Example 2: Consider a rigid airfoil that has two degrees of freedom: plunging along the h direction, and pitching in α . The authors of reference [27] have summarized the dynamics of the wing fluctuation using a linear and an angular springs, which are shown in Figure 5. By doing so, the nonlinear aeroelastic behavior of the wing can be modeled by a polynomial function of the pitch angle.

Denoting $[h, \alpha, \dot{h}, \dot{\alpha}]^T = [x_1, x_2, x_3, x_4]^T$ as the system states, the governing equations are written as

$$\dot{x} = A_F x + f_F(x) + B_F u, \quad (42)$$

where

$$A_F = \begin{bmatrix} 0.0 & 0.0 & 1.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 1.0 \\ -293.27 & -100.59 & -5.9027 & -0.40542 \\ 1885.9 & 743.79 & 34.728 & 2.4687 \end{bmatrix},$$

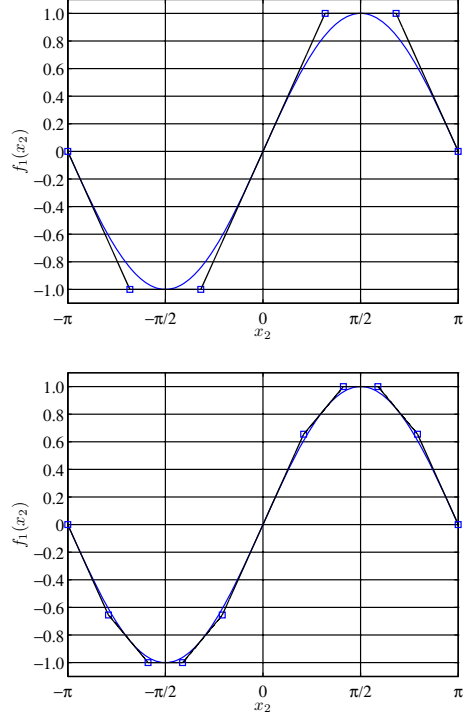


Fig. 4. IPWA approximation of the $f_1(x_2) = \sin(x_2)$. The upper plot is the approximation with 5 regions. The approximation has been refined in the lower figure by linearizing the intersection points, which are plotted with squares.

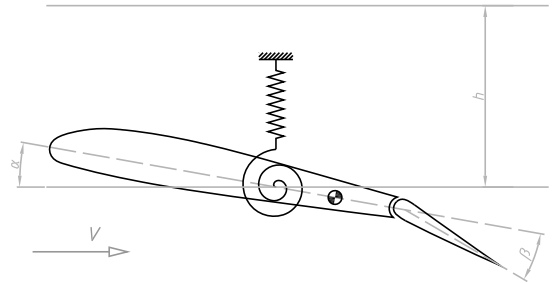


Fig. 5. Simplified aeroelastic model of the aircraft wing.

$$B_F = \begin{bmatrix} 0.0 & 0.0 \\ 0.0 & 0.0 \\ -7606.8 & -7642.6 \\ 14250 & 9021.9 \end{bmatrix},$$

and

$$f_F(x) = \begin{bmatrix} 0 \\ 0 \\ \bar{K}_\alpha(x_2) \\ \bar{K}_\alpha(x_2) \end{bmatrix},$$

with

$$\bar{K}_\alpha(x_2) = 2.82x_2 - 62.322x_2^2 + 3709.71x_2^3 - 24195.6x_2^4 + 48756.954x_2^5$$

that are taken from [27]. $\bar{K}_\alpha(x_2)$ is a function of only one variable x_2 , which allows us to use the IPWA algorithm to compute its PWA approximation. The summary of the IPWA

algorithm applied to the continuous function $\bar{K}_\alpha(x_2)$ for $x_2 \in (-\pi/6, \pi/6)$ is given in Table II. The normalized error \bar{e}_{max} is computed according to (40). Figure 6 shows two consecutive iterations of the IPWA algorithm. Point $x_{int}^{(3)} = -0.3434$ has the maximum error of approximation in the 3rd iteration (upper plot). This point is thus chosen to linearize the function $\bar{K}_\alpha(x_2)$, resulting in a finer PWA approximation (lower plot). Looking at Table II, one can conclude that the Lebesgue model has the least error for 3 regions while for more than 3 regions the IPWA is shown to produce better results.

TABLE II
THE SUMMARY OF PIECEWISE AFFINE APPROXIMATION FOR $\bar{K}_\alpha(x_2)$

Normalized Approximation Error (\bar{e}_{max}) of $\bar{K}_\alpha(x_2)$			
NUMBER OF REGIONS n	IPWA APPROX.	LEBESGUE PWA APPROX.	UG PWA APPROX.
3	0.3337	0.2492	0.3867
4	0.1113	0.1658	0.3010
5	0.0731	0.1242	0.2285
6	0.0483	0.0992	0.1684
7	0.0371	0.0825	0.1187
9	0.0195	0.0710	0.0423

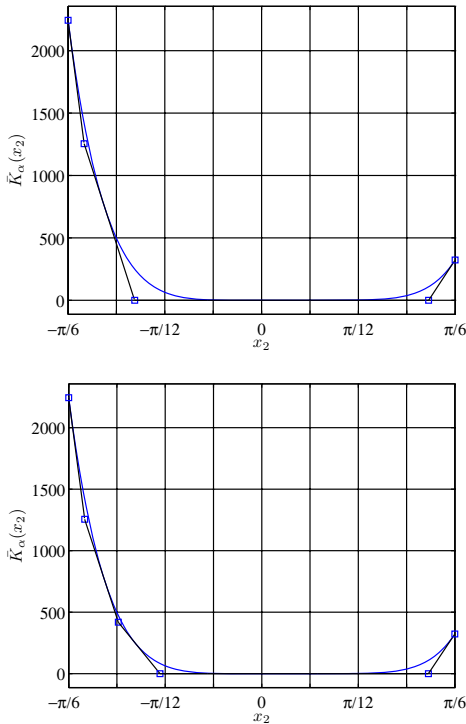


Fig. 6. IPWA approximation of the $\bar{K}_\alpha(x_2)$. The upper plot is the approximation at the 3rd iteration while the lower plot is associated with the 4th iteration of the algorithm.

Example 3: A set of cell DNAs that interact through their RNA and their protein products is called a gene regulatory network. Based on the interactive regulation of genes, a genetic network is produced, which describes the behavior of organisms, according to [28]. The two-gene regulation network is schematically shown in Figure 7.

Denoting x_1 and x_3 as the concentration of messenger RNAs (mRNAs), as well as x_2 and x_4 as the protein products

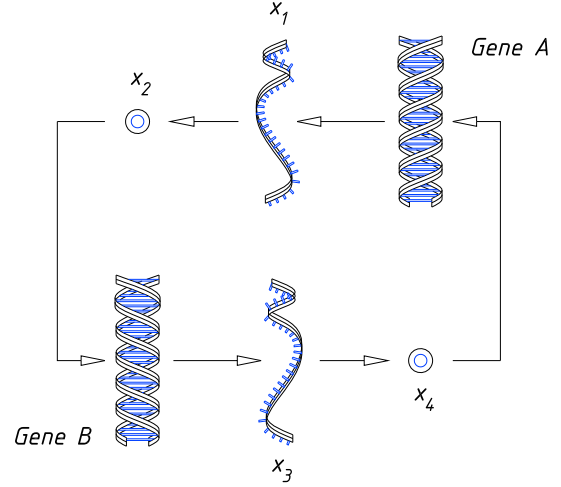


Fig. 7. Two-gene regulatory network. x_1 and x_2 represent mRNAs concentration while x_3 and x_4 stand for the protein product concentration of genes A and B, respectively. This figure is adapted from [9].

concentration of all genes A and B, respectively, the governing state space equations of the two-gene regulatory network is described by

$$\dot{x}_1 = \alpha_1 \sigma_n(x_4) - \beta_1 x_1 \quad (43a)$$

$$\dot{x}_2 = \alpha_2 x_1 - \beta_2 x_2 \quad (43b)$$

$$\dot{x}_3 = \alpha_3 \sigma_p(x_2) - \beta_3 x_3 \quad (43c)$$

$$\dot{x}_4 = \alpha_4 x_3 - \beta_4 x_4 \quad (43d)$$

where σ_n and σ_p are the transcriptional regulation functions depending on the concentrations of the protein products, according to [28] and [9]. These functions for a two-gene network problem may take the form of

$$\sigma_n(x_4) = \frac{\theta_n^\kappa}{\theta_n^\kappa + x_4^\kappa} \quad (44)$$

$$\sigma_p(x_2) = \frac{x_2^\kappa}{\theta_p^\kappa + x_2^\kappa}, \quad (45)$$

with $\theta_n = 0.20$, $\theta_p = 0.19$ and $\kappa = 4$. All the constants are taken from [9], as

$$\begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \alpha_4 \end{bmatrix} = \begin{bmatrix} 0.95 \\ 1.23 \\ 0.86 \\ 1.43 \end{bmatrix}, \quad \begin{bmatrix} \beta_1 \\ \beta_2 \\ \beta_3 \\ \beta_4 \end{bmatrix} = \begin{bmatrix} 0.90 \\ 1.31 \\ 0.83 \\ 2.60 \end{bmatrix}.$$

Using the IPWA algorithm, functions $\sigma_n(x_4)$ and $\sigma_p(x_2)$ are approximated for several desired errors. The Lebesgue approximation and the uniform grid techniques are also used to find the PWA model of these functions. The approximation error of all of the mentioned techniques are presented in Tables III and IV. For as low number of regions as three, the Lebesgue algorithm is shown to be most accurate. As the number of the regions is increased, the IPWA provides higher accuracy with respect to the other techniques.

Example 4: The last example for modeling of nonlinear systems with IPWA method consists of the mathematical

TABLE III

THE SUMMARY OF PIECEWISE AFFINE APPROXIMATION FOR $\sigma_n(x_4)$

Normalized Approximation Error (\bar{e}_{max}) of $\sigma_n(x_4)$			
NUMBER OF REGIONS n	IPWA APPROX.	LEBESGUE PWA APPROX.	UG PWA APPROX.
3	0.1738	0.1575	0.2083
4	0.0721	0.1167	0.2283
5	0.0667	0.0922	0.1912
6	0.0405	0.0760	0.1364
7	0.0259	0.0644	0.0907
8	0.0217	0.0558	0.0598
9	0.0196	0.0492	0.0439
10	0.0135	0.0439	0.0419
11	0.0109	0.0396	0.0464

TABLE IV

THE SUMMARY OF PIECEWISE AFFINE APPROXIMATION FOR $\sigma_p(x_2)$

Normalized Approximation Error (\bar{e}_{max}) of $\sigma_p(x_2)$			
NUMBER OF REGIONS n	IPWA APPROX.	LEBESGUE PWA APPROX.	UG PWA APPROX.
3	0.1745	0.1589	0.2001
4	0.0721	0.1178	0.2289
5	0.0672	0.0933	0.2043
6	0.0406	0.0771	0.1529
7	0.0262	0.0655	0.1052
8	0.0217	0.0568	0.0705
9	0.0196	0.0556	0.0479
10	0.0136	0.0447	0.0436
11	0.0109	0.0403	0.0447

function

$$f(x) = \left(x + 4^{-1/3}\right)^4 - x - 4^{-4/3}, \quad \forall x \in \Gamma, \quad (46)$$

where $\Gamma = (-2, 1)$. The specific feature of this function is that it has a zero curvature point at $x_{zc} = 4^{-1/3}$, which is not an inflection point. Therefore, the SLP algorithm picks x_{zc} as a member of the set of linearization points (SLP). The IPWA algorithm in contrast, does not linearize the function f at such point. Table V gives the normalized approximation error of PWA approximation of f by both the SLP and the IPWA algorithms. Although the error for both models with 2, 4 and 8 regions is the same, the IPWA model has smaller values of error with 3 and 5 regions.

TABLE V

THE SUMMARY OF PIECEWISE AFFINE APPROXIMATION FOR $f(x)$

Normalized Approximation Error (\bar{e}_{max}) of $f(x)$ Given in Equation (46)		
NUMBER OF REGIONS n	IPWA APPROX.	SLP PWA APPROX.
2	1.0000	1.0000
3	0.3401	0.5970
4	0.3037	0.3037
5	0.1469	0.1888
6	0.0896	0.0896

VI. CONCLUSION

In this paper, the intersection-based algorithm for approximation of functions of one variable was developed. Using the algorithms provided in section IV, PWA models can be constructed for a wide range of nonlinear functions, where the nonlinearity is a function of only one state. Finally the method was successfully applied to the path following

unicycle, the aeroelastic model of the aircraft wing, and the two-gene regulatory network. The approximation error of the intersection-based PWA model of the examples was compared with the Lebesgue [9], uniform grid [7], and the SLP [26] PWA models. Future work may involve study of the PWA approximation of functions of two or more variables.

REFERENCES

- [1] Mikael Johansson. *Piecewise Linear Control Systems*. Springer-Verlag, Berlin, first edition, 2003.
- [2] L. Rodrigues and J.P. How. Synthesis of piecewise-affine controllers for stabilization of nonlinear systems. *Proceedings of the 42nd IEEE Conference on Decision and Control*, 3:2071–2076, 2003.
- [3] L. Rodrigues and S. Boyd. Piecewise-affine state feedback for piecewise-affine slab systems using convex optimization. *Systems & Control Letters*, 54:835–853, 2005.
- [4] S. Casselman and L. Rodrigues. Piecewise affine modeling of a micro air vehicle using voronoi partitions. In *Proceedings of the European Control Conference*, pages 3857–3862, 2009.
- [5] A. Cantoni. Optimal curve fitting with piecewise linear functions. *IEEE Transactions on Computers*, C-20:59–67, January 1971.
- [6] I. Tomek. Two algorithms for piecewise-linear continuous approximation of functions of one variable. *IEEE Transactions on Computers*, C-24:445–448, April 1974.
- [7] P. Julian, 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*, 46:1057–1122, April 1999.
- [8] Richard E. Groff, Pramod P. Khargonekar, and Daniel E. Koditschek. A local convergence proof for the MINVAR algorithm for computing continuous piecewise linear approximations. *SIAM Journal of Numerical Analysis*, 41(3):983–1007, 2003.
- [9] S. Azuma, J. Imura, and T. Sugie. Lebesgue piecewise affine approximation of nonlinear systems. *Nonlinear Analysis: Hybrid Systems*, 4(1):92–102, 2010.
- [10] Marco Storace and Oscar De Feo. Piecewise-linear approximation of nonlinear dynamical systems. *IEEE Transactions on Circuits and Systems*, Vol. 51(No. 4):830–842, 2004.
- [11] Jorge M. Gonçalves, Alexandre Megretski, and Munther A. Dahleh. Global analysis of piecewise linear systems using impact maps and surface lyapunov functions. *IEEE Transactions on Automatic Control*, Vol. 48(No. 12):2089–2106, 2003.
- [12] E. Asarin, T. Dang, and A. Girard. *Hybrid Systems: Computation and Control*, pages 20–35. Springer, 2003.
- [13] Gang Feng. Controller design and analysis of uncertain piecewise-linear systems. *IEEE Transactions on Circuits and Systems - I: Fundamental Theory and Applications*, Vol. 49(No. 2):224–232, 2002.
- [14] Luis Rodrigues and Jonathan P. How. Automated Control Design for a Piecewise-Affine Approximation of a Class of Nonlinear Systems. In *Proceedings of the American Control Conference*, pages 3189–3194, 2001.
- [15] Wei Yue, Luis Rodrigues, and Brandon Gordon. Piecewise-affine control of a three DOF helicopter. In *Proceedings of the 2006 American Control Conference*, pages 3924–3929, 2006.
- [16] Daniele Corona and Bart De Schutter. Adaptive cruise control for a SMART car: A comparison benchmark for MPC-PWA control methods. *IEEE Transactions on Control Systems Technology*, Vol. 16(Number 2):365–372, 2008.
- [17] G. Ferrari-Trecate, M. Muselli, D. Liberati, and M. Morari. A clustering technique for the identification of piecewise affine systems. *Automatica*, 39(2):205–217, 2003.
- [18] A. Bemporad and M. Morari. Control of systems integrating logic, dynamics, and constraints. *AUTOMATICA-OXFORD-*, 35:407–428, 1999.
- [19] C. Tomlin, I. Mitchell, and R. Ghosh. Safety verification of conflict resolution manoeuvres. *Intelligent Transportation Systems, IEEE Transactions on*, 2(2):110–120, 2001.
- [20] R. Ghosh, A. Tiwari, and C. Tomlin. Automated symbolic reachability analysis; with application to delta-notch signaling automata. *Hybrid Systems: Computation and Control*, pages 233–248, 2003.
- [21] S.I. Azuma and J.I. Imura. Polynomial-time probabilistic controllability analysis of discrete-time piecewise affine systems. *Automatic Control, IEEE Transactions on*, 52(11):2029–2046, 2007.

- [22] S. Azuma and J. Imura. Polynomial-time probabilistic observability analysis of sampled-data piecewise affine systems. In *Decision and Control, 2005 and 2005 European Control Conference. CDC-ECC'05. 44th IEEE Conference on*, pages 6644–6649. IEEE, 2005.
- [23] Behzad Samadi and Luis Rodrigues. Extension of local linear controllers to global piecewise affine controllers for uncertain non-linear systems. *International Journal of Systems Science*, 39(9):867–879, 2008.
- [24] H.L. Royden. *Real Analysis*. MACMILLAN, second edition, 1968.
- [25] David G. Luenberger and Yinyu Ye. *Linear and Nonlinear Programming*. Springer, third edition, 2008.
- [26] S. Casselman and L. Rodrigues. A new methodology for piecewise affine models using voronoi partitions. In *Proc. 48th IEEE Conference on Decision and Control and 28th Chinese Control Conference*, pages 3920–3925, 2009.
- [27] S. Afkhami and H. Alighanbari. Nonlinear control design of an airfoil with active flutter suppression in the presence of disturbance. *IET Control Theory Appl.*, 1(6):1638–1649, November 2007.
- [28] A. Mochizuki. An analytical study of the number of steady states in gene regulatory networks. *Journal of Theoretical Biology*, 236(3):291–310, 2005.