Automatic Derivation of Optimal Piecewise Affine Approximations of Nonlinear Systems

Michal Kvasnica ∗,1, Alexander Szűcs ∗, and Miroslav Fikar ∗

∗ Institute of Automation, Information Engineering and Mathematics, Slovak University of Technology, 812 37 Bratislava, Slovakia

Abstract: The paper proposes a method for deriving optimal piecewise affine (PWA) approximations of nonlinear systems with known analytic form. The procedure employs nonlinear optimization to derive an approximation of maximal accuracy and given complexity. We show that under mild assumptions, the task can be transformed into a series of one-dimensional approximations. An automatic procedure for generation of such approximations is discussed as well.

Keywords: hybrid systems, approximation, nonlinear optimization

1. INTRODUCTION

Mathematical models of physical plants play a vital role in many areas, such as in rigorous simulations, analysis, or control synthesis. Typically, high model accuracy is usually desired while keeping the model complexity on an acceptable level. Traditionally, nonlinear models were preferred from simulations, while most of available control techniques are based on a local approximation around a single operating point. PWA systems (Sontag, 1981) can be viewed as a compromise solution between accuracy of the model and its complexity. They are composed of several local models accompanied with logic IF-THEN conditions which enforce switching of the local dynamics. Therefore they belong to the class of hybrid systems.

The problem which we address in this paper is the following: given a nonlinear dynamical model \( \dot{x} = f(x, u) \) and a fixed complexity of its PWA approximation \( \tilde{f}(x, u) \approx f(x, u) \), how should one design \( \tilde{f} \) which minimizes the approximation error \( \| f(x, u) - \tilde{f}(x, u) \|^2 \)? The answer is non-trivial even when putting optimality of the approximation aside. Traditionally, two distinct approaches for deriving PWA approximations are used. When the mathematical formulation of the original nonlinear system is known, one can design the approximation by hand. This is usually done by employing human knowledge and experience to devise several linearization points around which the original nonlinear model should be linearized. Needless to say, placement of such points has a crucial impact on the accuracy of the approximation. The HYSDEL (Hybrid Systems Description Language) tool (Torrisi and Bemporad, 2004; Kvasnica and Herceg, 2010) can be used to accelerate this line of development. Formally, HYSDEL transforms a linguistic description of a PWA system (or of a hybrid system in general) into the corresponding mathematical form. The language allows to define IF-THEN switching rules which, based on whether some logic condition is satisfied or not, enforce certain continuous dynamics. Another option is to use hybrid identification techniques (Ferrari-Trecate et al., 2001; Roll et al., 2004; Ferrari-Trecate, 2005) to construct the PWA approximation from the input-output measurements. The crucial advantage is that the model of the original nonlinear system is not required to be fully available. The downside, however, is that the approximation is only accurate in the interval captured by the identification data. Moreover, the procedure is computationally expensive and suited mainly to low-dimensional problems.

In this work we propose to use an optimization-based approach to derive PWA approximations of nonlinear systems whose vector field is an a-priori known function of multiple variables. After formally stating the problem in Section 2, we show in Section 3 that an optimal PWA approximation of generic nonlinear functions in one variable can be formulated and solved as a nonlinear programming problem. Several non-trivial illustrative cases are discussed to show that the approach is both efficient and computationally tractable. Subsequently, the approach is extended to deriving PWA approximations of multivariable functions in Section 4. We show that, under a certain assumption, the problem boils down to solving a series of one-dimensional approximations. The algorithmic and software implementation of the approximation procedure are then discussed in Section 5. Specifically, we introduce a new software tool which is capable of exporting the obtained optimal PWA approximations into the HYSDEL language. This brings two crucial advantages. First, the HYSDEL compiler can be used to convert the PWA approximation into a mathematical form, which is then suitable e.g. for control design. Second, since the exported approximation is described in a human-readable format, it can be further fine-tuned by hand. Finally, in Section 6 we illustrate the procedure on a case study involving a highly non-linear chemical reactor.

1 Corresponding author, e-mail: michal.kvasnica@stuba.sk
2. PROBLEM STATEMENT

We consider generic dynamic systems in discrete-time
\[ x^{+} = f(x, u), \]
where the vector field \( f(\cdot, \cdot) \) is assumed to be continuous in the state variables \( x \in \mathbb{R}^{n_x} \) and in the inputs \( u \in \mathbb{R}^{n_u} \). System states and inputs are assumed to be constrained to connected and closed domains \( X \subset \mathbb{R}^{n_x} \) and \( U \subset \mathbb{R}^{n_u} \), respectively.

The objective is to approximate (1) by a different dynamic system \( x^{+} = \tilde{f}(x, u) \) whose vector field \( \tilde{f}(x, u) \) is a PWA function which consists of a pre-specified number \( N \) of local linear dynamics:
\[ \tilde{f}(x, u) = \begin{cases} A_1 x + B_1 u + c_1 & \text{if } [\begin{array}{c} x \\ u \end{array} ] \in R_1 \\ \vdots & \vdots \\ A_N x + B_N u + c_N & \text{if } [\begin{array}{c} x \\ u \end{array} ] \in R_N \end{cases} \] (2)

Here, \( A_i \in \mathbb{R}^{n_x \times n_x}, B_i \in \mathbb{R}^{n_x \times n_u}, c_i \in \mathbb{R}^{n_x} \), are the state-update matrices of the \( i \)-th local linear approximation, and \( R_i \subset \mathbb{R}^{n_x \times n_u} \) is the region of validity of the \( i \)-th local model satisfying \( R_i \neq \emptyset, R_i \cap R_j = \emptyset, \forall i \neq j, \) and \( \bigcup_1 \bigcup_i R_i = X \times U \).

Formally, the problem which we aim at solving can be stated as follows:

**Problem 2.1.** Given a nonlinear vector field \( f(x, u) \) of system (1), find the PWA approximation (2) of pre-specified complexity which minimizes the approximation error
\[ e_{\text{apprx}} = \int (f(x, u) - \tilde{f}(x, u))^2 \, dx 
\] (3)

where the integral is evaluated over the whole region of validity of (1), i.e. over \( X \times U \).

In the sequel we show how to solve Problem 2.1 provided that the vector field \( f(z) \), \( z = [x, u]^T \) satisfies the following assumption.

**Assumption 2.2.** The function \( f(z_1, \ldots, z_n) \) can be written as
\[ \sum_{i=1}^{n} a_i \left( \prod_{j=p}^{q} f_j(z_j) \right). \]

As an example, the function \( z_1 e^{z_2} \) satisfies such an assumption, while the function \( e^{z_1z_2} \) does not. Although the assumption is somewhat restrictive, the gained advantage is that approximating any multivariable function \( f(z_1, \ldots, z_n) \) boils down to solving a series of 1D problems, as evidenced in the following two sections.

**Remark 2.3.** Since the approximation procedure discussed in the sequel considers only the vector field in the right-hand-side of (1), continuous-time systems \( \dot{x} = f(x, u) \) can be treated as well.

3. FUNCTIONS IN ONE VARIABLE

First, we consider the one-dimensional case, i.e. approximating a nonlinear function \( f(z) : \mathbb{R} \rightarrow \mathbb{R} \), with domain \( Z \subset \mathbb{R} \), by a PWA function \( \tilde{f}(z) = a_i z + c_i \) if \( z \in R_i \). Since \( Z \) is assumed to be connected and closed, it is a line segment \([\underline{z}, \overline{z}]\). Regions \( R_i \) define the partition of such a line into \( N \) non-overlapping parts, i.e. \( R_1 = [\underline{z}, r_1], R_2 = [r_1, r_2], \ldots, R_{N-1} = [r_{N-2}, r_{N-1}], R_N = [r_{N-1}, \overline{z}] \) with \( \bigcup_1 \bigcup_i R_i = [\underline{z}, \overline{z}] \). Solving Problem 2.1 then becomes to find the slopes \( a_i \), offsets \( c_i \) and breakpoints \( r_i \) such that the approximation error is minimized, i.e.
\[ \min_{a_i, c_i, r_i} \int_{\underline{z}}^{\overline{z}} (f(z) - \tilde{f}(z))^2 \, dz \] (4a)
\[ \text{s.t.} \quad \tilde{f}(z) = \begin{cases} a_1 z + c_1 & \text{if } z \in [\underline{z}, r_1] \\ \vdots & \vdots \\ a_N z + c_N & \text{if } z \in [r_{N-1}, \overline{z}] \end{cases} \] (4b)
\[ a_i r_i + c_i = a_{i+1} r_i + c_{i+1}, \quad i = 1, \ldots, N - 1 \] (4d)

where (4d) enforces continuity of \( \tilde{f}(z) \) along the breakpoints \( r_i \). The IF-THEN based nonlinear constraint (4b) can be eliminated by observing that, by definition, regions \( R_i \) are non-overlapping, and the integral in (4a) can hence be written as
\[ \int_{\underline{z}}^{\overline{z}} (f(z) - \tilde{f}(z))^2 \, dz = \sum_{i=1}^{N} \left( \int_{r_{i-1}}^{r_i} (f(z) - (a_i z + c_i))^2 \, dz \right) \] (5)
\[ \text{s.t.} \quad \underline{z} \leq r_1 \leq \cdots \leq r_{N-1} \leq \overline{z}, \] (6b)
\[ a_i r_i + c_i = a_{i+1} r_i + c_{i+1}, \quad i = 1, \ldots, N - 1 \] (6c)

For simple functions \( f(z) \), the integral in (6a) can be expressed in an analytical form in unknowns \( a_i, c_i, r_i \), along with the corresponding gradients. For more complex expressions, the integrals can be evaluated numerically, e.g. by using the trapezoidal rule. In either case, problem (6) can be solved to a local optimality e.g. by using the fmincon solver of MATLAB. Alternatively, one can use global optimization methods (Adjiman et al., 1996; Papamichail and Adjiman, 2004; Chachuat et al., 2006) which guarantee that an \( \epsilon \)-neighborhood of the global optimum can be found.

**Example 3.1.** Consider the function \( f(z) = z^3 \) on domain \(-1.5 \leq z \leq 1.5\). The analytic form of the integral (6a) is
\[ \sum_{i=1}^{N} \left( \int_{r_{i-1}}^{r_i} (f(z) - (a_i z + c_i))^2 \, dz \right) \]
\[ \text{s.t.} \quad \underline{z} \leq r_1 \leq \cdots \leq r_{N-1} \leq \overline{z}, \]
\[ a_i r_i + c_i = a_{i+1} r_i + c_{i+1}, \quad i = 1, \ldots, N - 1 \]

For simple functions \( f(z) \), the integral in (6a) can be expressed in an analytical form in unknowns \( a_i, c_i, r_i \), along with the corresponding gradients. For more complex expressions, the integrals can be evaluated numerically, e.g. by using the trapezoidal rule. In either case, problem (6) can be solved to a local optimality e.g. by using the fmincon solver of MATLAB. Alternatively, one can use global optimization methods (Adjiman et al., 1996; Papamichail and Adjiman, 2004; Chachuat et al., 2006) which guarantee that an \( \epsilon \)-neighborhood of the global optimum can be found.

**Example 3.1.** Consider the function \( f(z) = z^3 \) on domain \(-1.5 \leq z \leq 1.5\). The analytic form of the integral (6a) is
\[ \sum_{i=1}^{N} \left( \int_{r_{i-1}}^{r_i} (f(z) - (a_i z + c_i))^2 \, dz \right) \]
\[ \text{s.t.} \quad \underline{z} \leq r_1 \leq \cdots \leq r_{N-1} \leq \overline{z}, \]
\[ a_i r_i + c_i = a_{i+1} r_i + c_{i+1}, \quad i = 1, \ldots, N - 1 \]
Example 3.2. Consider the function $f(z) = |z| + 0.5z^2 - \sin(z^3)$ on domain $-1 \leq z \leq 2.5$, graph of which is shown in Figure 2(a). Since no analytic expression of the integral in (6a) could be obtained, we have opted for numeric integration of the cost while solving the NLP problem (6) by fmincon. The PWA approximations for $N = 3$ and $N = 7$ are shown in Figures 2(a) and 2(b).

4. MULTIVARIABLE FUNCTIONS

The task is to approximate a given multivariable function $f(z_1, \ldots, z_n) : \mathbb{R}^n \to \mathbb{R}$ with domain $Z \subset \mathbb{R}^n$ by a PWA function $f(z_1, \ldots, z_n)$, defined over the same domain, such that the approximation error (3) is minimized.

Definition 4.1. (Williams (1993)). Function $f(z_1, \ldots, z_n)$ is called separable if it can be expressed as the sum of functions of a single variable, i.e. $f(z_1, \ldots, z_n) = f_1(z_1) + \cdots + f_n(z_n)$.

If $f(z_1, \ldots, z_n)$ is readily separable (e.g. when $f(z_1, z_2) = e^{z_1} + \sin(z_2)$), its optimal PWA approximation can be obtained by applying the 1D scenario of Section 3 to the individual components of the function, i.e. $f(z_1, \ldots, z_n) = f_1(z_1) + \cdots + f_n(z_n)$. The total number of regions over which the PWA approximation $\tilde{f}(\cdot)$ is defined is hence given by $\sum_{j=1}^n N_j$, where $N_j$ is the pre-specified complexity of the $j$-th approximation $\tilde{f}_j(z_j)$.

A surprisingly large number of non-separable functions can be converted into the separable form by applying a simple trick, elaborated in more details e.g. in Williams (1993). To introduce the procedure, consider a non-separable function $f(z_1, z_2) = z_1 z_2$ with domain $Z := \mathbb{R}^2$. Define two new variables

$$y_1 = (z_1 + z_2), \quad y_2 = (z_1 - z_2).$$

Then it is easy to verify that $1/4(y_1^2 - y_2^2) = z_1 z_2$. The coordinate transformation therefore transforms the original function into a separable form, where both terms $(y_1^2$ and $y_2^2$) are now functions of a single variable. The procedure of Section 3 can thus be applied to compute PWA approximations of $f(y_1) := y_1^2$ and $f(y_2) := y_2^2$, where the function arguments relate to $z_1$ and $z_2$ via (7). Important to notice is that $f_{y_1}(\cdot)$ and $f_{y_2}(\cdot)$ have different domains, therefore their PWA approximations $f_{y_1}(y_1) \approx y_1^2$ and $f_{y_2}(y_2) \approx y_2^2$ will, in general, be different. Specifically, the domain of $f_{y_1}(\cdot)$ is $[\underline{y}_1, \overline{y}_1]$ with $\underline{y}_1 = \min\{z_1 + z_2 | \underline{z}_1 \leq z_1 \leq \overline{z}_1, \underline{z}_2 \leq z_2 \leq \overline{z}_2\}$ and $\overline{y}_1 = \max\{z_1 + z_2 | \underline{z}_1 \leq z_1 \leq \overline{z}_1, \underline{z}_2 \leq z_2 \leq \overline{z}_2\}$. Similarly, the domain of $f_{y_2}(\cdot)$ is $[\underline{y}_2, \overline{y}_2]$, whose boundaries can be computed by respectively minimizing and maximizing $z_1 - z_2$ subject to the constraint $[z_1, z_2]^T \in Z$. The overall PWA approximation $\tilde{f}(z_1, z_2) \approx z_1 z_2$ then becomes

$$\tilde{f}(z_1, z_2) = 1/4(f_{y_1}(z_1 + z_2) - f_{y_2}(z_1 - z_2)).$$

The value of $\tilde{f}(z_1, z_2)$ for any points $z_1, z_2$ is obtained by subtracting the value of the PWA function $\tilde{f}_{y_2}(\cdot)$ evaluated at the point $z_1 - z_2$ from the function value of $\tilde{f}_{y_1}(\cdot)$ evaluated at $z_1 + z_2$, followed by a linear scaling.

The procedure naturally extends to multivariable functions represented by the product of two nonlinear functions of a single variable, i.e. $f(z_1, z_2) = f_1(z_1)f_2(z_2)$. Here, the transformation (7) becomes

$$y_1 = f_1(z_1) + f_2(z_2), \quad y_2 = f_1(z_1) - f_2(z_2).$$

Therefore, $1/4(y_1^2 - y_2^2) = f(z_1, z_2)$. Let $f_{y_1}(y_1) := y_1^2$ and $f_{y_2}(y_2) := y_2^2$. The domain of $f_{y_1}(\cdot)$ is $[\underline{y}_1, \overline{y}_1]$ and $f_{y_2}(\cdot)$ is $[\underline{y}_2, \overline{y}_2]$ with

$$\underline{y}_1 = \min\{f_1(z_1) + f_2(z_2) | [z_1, z_2]^T \in Z\}, \quad \overline{y}_1 = \max\{f_1(z_1) + f_2(z_2) | [z_1, z_2]^T \in Z\},$$

$$\underline{y}_2 = \min\{f_1(z_1) - f_2(z_2) | [z_1, z_2]^T \in Z\}, \quad \overline{y}_2 = \max\{f_1(z_1) - f_2(z_2) | [z_1, z_2]^T \in Z\},$$

which can be computed by solving four NLP problems. Finally, since all expressions are now functions of a single variable, the PWA approximations $\tilde{f}_{y_1}(\cdot) \approx f_1(z_1)$, $\tilde{f}_{y_2}(z_2) \approx f_2(z_2)$, $\tilde{f}_{y_1}(y_1) \approx f_{y_1}(y_1)$, and $\tilde{f}_{y_2}(y_2) \approx f_{y_2}(y_2)$ can be computed by solving the NLP (6). The overall optimal PWA approximation $\tilde{f}(z_1, z_2) \approx f(z_1, z_2)$ then becomes

$$\tilde{f}(z_1, z_2) = 1/4\left(\tilde{f}_{y_1}(\tilde{f}_1(z_1) + \tilde{f}_2(z_2)) - \tilde{f}_{y_2}(\tilde{f}_1(z_1) - \tilde{f}_2(z_2))\right).$$

The evaluation procedure is similar as above. I.e., given the arguments $z_1$ and $z_2$, one first evaluates $\tilde{z}_1 = \tilde{f}_1(z_1)$ and $\tilde{z}_2 = \tilde{f}_2(z_2)$. Subsequently, one evaluates $\tilde{y}_1 = \tilde{f}_{y_1}(\cdot)$ with the argument $\tilde{z}_1 + \tilde{z}_2$, then $\tilde{y}_2 = \tilde{f}_{y_2}(\cdot)$ at the point $\tilde{z}_1 - \tilde{z}_2$. Finally, $\tilde{f}(z_1, z_2) = 1/4(\tilde{y}_1 - \tilde{y}_2)$.

Example 4.1. Consider a non-separable function given as the product of the two functions discussed in Examples 3.1 and 3.2, i.e. $f(z_1, z_2) = f_1(z_1)f_2(z_2)$ with $f_1(z_1) = z_1^3$, $f_2(z_2) = |z_2| + 0.5z_2^2 - \sin(z_2)^3$ on domain $[-1.5, 1.5] \times [-1.5, 1.5]$.
Approximation−6
\begin{equation}
y_3 = f_a(\cdot) + f_3(z_3), \quad y_4 = f_a(\cdot) - f_3(z_3),
\end{equation}
and hence \(f_a(z_1, z_2)f_0(z_3) = 1/4(y_3^2 - y_4^2)\). The domains over which \(y_3^2\) and \(y_4^2\) need to be approximated are, respectively, \([y_3', y_3]\) and \([y_4', y_4]\)
with
\begin{align}
y_3 &= \min\{f_1(z_1)f_2(z_3) + f_3(z_3) \mid z \in Z\}, \\
y_4 &= \min\{f_1(z_1)f_2(z_3) - f_3(z_3) \mid z \in Z\},
\end{align}
and \(z = [z_1, z_2, z_3]^T\). Subsequently, three additional PWA approximations
\begin{equation}
\tilde{f}_{y_3}(y_3) \approx y_3^2, \quad \tilde{f}_{y_4}(y_4) \approx y_4^2, \quad \tilde{f}_{z_3}(z_3) \approx f_3(z_3)
\end{equation}
need to be computed over the corresponding domains. The aggregated optimal PWA approximation \(\tilde{f}(z_1, z_2, z_3) \approx f(z_1)f(z_2)f(z_3)\) consists of 7 individual approximations and is given by
\begin{equation}
\tilde{f}(\cdot) = 1/4\left(\tilde{f}_{y_3}(\tilde{f}_{y_3}(z_1) + \tilde{f}_3(z_3)) - \tilde{f}_{y_4}(\tilde{f}_{y_4}(z_1) - \tilde{f}_3(z_3))\right).
\end{equation}
Here, \(\tilde{f}_a\) is the function value of \(\tilde{f}_{a}(z_1, z_2) \approx f_1(z_1)f_2(z_2)\) at \(z_1\) and \(z_2\), where \(\hat{f}_a(\cdot)\) is obtained from (11), i.e.:
\begin{equation}
\tilde{f}_a = 1/4\left(\tilde{f}_{y_3}(\tilde{f}_{y_3}(z_1) + \tilde{f}_3(z_3)) - \tilde{f}_{y_4}(\tilde{f}_{y_4}(z_1) - \tilde{f}_3(z_3))\right).
\end{equation}
The overall PWA approximation \(\tilde{f}(z_1, z_2, z_3)\) can then be evaluated, for any \(z_1, z_2, z_3 \in Z\), by computing the function values of the respective approximations in the following order:
\begin{enumerate}
  \item \(\hat{y}_1 = \tilde{f}_{y_1}(\tilde{f}_{y_1}(z_1) + \tilde{f}_2(z_2)),\)
  \item \(\hat{y}_2 = \tilde{f}_{y_2}(\tilde{f}_{y_2}(z_1) - \tilde{f}_2(z_2)),\)
  \item \(\hat{y}_3 = \tilde{f}_{y_3}(1/4(\hat{y}_1 - \hat{y}_2) + f_3(z_3)),\)
  \item \(\hat{y}_4 = \tilde{f}_{y_4}(1/4(\hat{y}_1 - \hat{y}_2) - f_3(z_3)),\)
  \item \(\hat{y}_5 = \tilde{f}_1(z_1, z_2, z_3) = 1/4(\hat{y}_3 - \hat{y}_4).\)
\end{enumerate}
Such an inductive procedure can be repeated \textit{ad-infinitum} to derive PWA approximations of any multivariable function which satisfies Assumption 2.2. In general, the PWA approximation will consists of \(2p + n\) individual PWA functions, where \(n\) is the number of variables in \(f(z_1, \ldots, z_n)\) and \(p\) is the number of products between individual subfunctions \(f_j(z_j)\). As an example, for \(f(\cdot) := \alpha_1 f_1(z_1)f_2(z_2)f_4(z_4) + \alpha_2 f_3(z_3)f_3(z_3)\) we have \(p = 3\). We remark that inclusion of scalar multipliers \(\alpha_j\) into the PWA description of the form (14)–(15) is straightforward and only requires linear scaling of the corresponding terms.

5. SOFTWARE IMPLEMENTATION

An algorithmic implementation of the inductive separation procedure of Section 4 is discussed next, provided that all functions are given in their symbolic representation. The procedure relies on two basic building blocks. The first one, represented by Algorithm 1, constructs the PWA approximation of a product of two functions, i.e. computes \(f(z_1, z_2) \approx f_1(z_1)f_2(z_2)\). Strictly speaking, the algorithm differentiates between two scenarios. If either \(f_i\) or \(f_j\) are
Algorithm 2 PWA approximation of $\prod_{i=1}^n f_i(z_i)$

**INPUT:** Functions $f_i(z_i).

**OUTPUT:** Approximation $\tilde{f}(z_1, \ldots, z_n) \approx \prod_{i=1}^n f_i(z_i)$.

1. Create an empty last-in-first-out stack $S$ and an empty tree $T$.
2. Push $f_i(z_i)$, $i = n, \ldots, 1$ to the stack $S$.
3. while $S$ has more than one element do
   4. Pop two elements $f_j(z_j)$ and $f_k(z_k)$ from $S$.
   5. Obtain $\tilde{f}_j(z_j), \tilde{f}_k(z_k)$, and $\tilde{f}(z_j, z_k) \approx f_j(z_j)f_k(z_k)$ by calling Algorithm 1.
   6. Push $\tilde{f}(z_j, z_k)$ to $S$.
   7. Create nodes $\tilde{f}_j(z_j), \tilde{f}_k(z_k)$ and insert them to $T$.
   8. Create a node $\tilde{f}(z_j, z_k)$ and append it as a child of nodes $\tilde{f}_j(z_j)$ and $\tilde{f}_k(z_k)$.
9: end while
10: return Tree $T$ representing $\tilde{f}(z_1, \ldots, z_n) \approx \prod_{i=1}^n f_i(z_i)$.

Consider a continuous stirred tank reactor (CSTR) where the reaction $A \rightarrow B$ takes place. The source component is pumped into the reactor at a constant inflow with a constant concentration. The chemical reaction is exothermic and a coolant liquid is therefore pumped into the reactor's jacket to prevent overheating. The input temperature of the coolant is constant, while its flow rate $q_c$ can be manipulated and is considered an exogenous input. Concentration of the reactant $c_A$ inside of the reactor, temperature of the reactor mixture $\vartheta$, and temperature of the cooling liquid in the jacket $\vartheta_c$ are the state variables of the CSTR. The normalized material and energy balances of such a reactor are then given by

$$\dot{c}_A = \alpha_1 - \alpha_2 c_A - \alpha_3 c_A e^{-\eta_\vartheta c},$$
$$\dot{\vartheta} = \alpha_4 - \alpha_5 \alpha_2 c_A e^{-\eta_\vartheta c} + \alpha_6 \vartheta + \alpha_7 \vartheta_c,$$
$$\dot{\vartheta}_c = \alpha_8 c_A + \alpha_9 (\vartheta - \vartheta_c) - \alpha_{10} \vartheta_c q_c,$$

with constants $\alpha_i$ and $\beta$. The state and input variables are considered to belong to intervals $c_A \in [4, 4.2]$ mol·m$^{-3}$, $\vartheta \in [300, 320]$ K, $\vartheta_c \in [290, 310]$ K, and $q_c \in [0.002, 0.02]$ m$^3$·h$^{-1}$.

The model features two nonlinearities: $\vartheta_c q_c$ and $c_A e^{-\eta_\vartheta c}$, both of which satisfy Assumption 2.2. Since the first one involves a direct product of two variables, its PWA approximation $\tilde{f}_o \approx \vartheta_c q_c$ can be obtained as in (8) by first defining $y_1 = \vartheta_c + q_c$, $y_2 = \vartheta_c - q_c$, followed by approximating the functions $y_1^2$ and $y_2^2$ by $\tilde{f}_y(y_1)$ and $\tilde{f}_y(y_2)$, respectively. Hence, the approximation $\tilde{f}_1(\vartheta_c, q_c) \approx \vartheta_c q_c$ is represented by

$$\tilde{f}_1(\vartheta_c, q_c) = 1/4(\tilde{f}_y(\vartheta_c + q_c) - \tilde{f}_y(\vartheta_c - q_c)).$$
The second nonlinearity can be approximated as in (11). First, the PWA approximation \( \tilde{g}(\vartheta) \approx e^{-\beta / \vartheta} \) is computed by solving (6). Then, \( y_3 = c_A + e^{-\beta / \vartheta}, \quad y_4 = c_A - e^{-\beta / \vartheta} \) are defined, followed by computing the respective PWA approximations \( \tilde{f}_{y_3}(y_3) \approx y_3^2 \) and \( \tilde{f}_{y_4}(y_4) \approx y_4^4 \). \( \tilde{f}_2(c_A, \vartheta) \approx \) is thus given by

\[
\tilde{f}_2(c_A, \vartheta) = 1/4 \left( \tilde{f}_{y_3}(c_A + \tilde{g}(\vartheta)) - \tilde{f}_{y_4}(c_A - \tilde{g}(\vartheta)) \right) \tag{18}
\]

The overall PWA approximation of the original nonlinear system \( \dot{x} = f(x, u) \) with \( x = [c_A, \vartheta, \vartheta_c]^T \) and \( u = q_c \) is thus

\[
\dot{c}_A \approx \alpha_1 - \alpha_2 c_A - \alpha_3 \tilde{f}_2(c_A, \vartheta), \\
\dot{\vartheta} \approx \alpha_4 - \alpha_5 \alpha_2 \tilde{f}_2(c_A, \vartheta) + \alpha_6 \vartheta + \alpha_7 \vartheta_c + \vartheta, \\
\dot{\vartheta}_c \approx \alpha_8 q_c + \alpha_9 (\vartheta - \vartheta_c) - \alpha_10 \tilde{f}_1(\vartheta_c, q_c),
\]

which can be easily converted into the general PWA form (2) as described in the previous section.

To assess approximation accuracy, we have investigated the open-loop evolution of the original nonlinear model (16) and compared it to the behavior of its PWA approximation (19). To derive the PWA model, we have chosen 3 regions for \( \tilde{f}_{y_3}(\cdot), \tilde{f}_{y_4}(\cdot) \) in (17) and \( \tilde{f}_{y_3}(\cdot), \tilde{f}_{y_4}(\cdot) \) in (18), and \( N = 2 \) for \( \tilde{g}(\vartheta) \approx e^{-\beta / \vartheta} \). The simulation results are shown in Figure 6. To better illustrate advantages of the PWA approximation, the simulation scenario also shows evolution of linearized version of (16) around the nominal steady state \( c_A^* = 4.13, \vartheta^* = 304, \vartheta_c^* = 297 \), and \( q_c^* = 0.006 \). As can be seen from the results, the PWA approximation clearly outperforms the model based on a single linearization. Specifically, the model (19) provides a 15 times more accurate tracking of the nonlinear profile compared to the linear model. Important to notice is that the PWA model consists of 14 local linear models. By increasing \( N \) to 7 when approximating \( \tilde{f}_{y_3}(\cdot), \tilde{f}_{y_4}(\cdot) \) in (17) and \( \tilde{f}_{y_3}(\cdot), \tilde{f}_{y_4}(\cdot) \) in (18), the approximation accuracy is 60 times better compared to the linear model. The cost to be paid is the increased model complexity, which would then consist of 30 regions.

7. CONCLUSIONS

We have shown that a large class of dynamical systems with nonlinear vector fields can be approximated by PWA systems of fixed complexity in an optimal manner. The procedure boils down to solving a series of one-dimensional problems for which efficient solution methods exist. Derivation of the approximation can be easily automated and the HYSDEL variant of the PWA approximation can be generated, hence allowing for subsequent control synthesis based on the hybrid model.

ACKNOWLEDGMENT

The authors are pleased to acknowledge the financial support of the Scientific Grant Agency of the Slovak Republic under the grants 1/0071/09 and 1/0095/11. This work was supported by the Slovak Research and Development Agency under the contract No. LPP-0092-07.

REFERENCES


Fig. 6. Simulation results for the CSTR. Red line: nonlinear model (16), blue dashed line: PWA model (19), black dotted line: linear approximation.