**Delft University of Technology**

**Delft Center for Systems and Control**

Technical report 23-027

# Parametric piecewise-affine approximation of nonlinear systems: A cut-based approach*

L. Gharavi, B. De Schutter, and S. Baldi

*If you want to cite this report, please use the following reference instead:*
L. Gharavi, B. De Schutter, and S. Baldi, "Parametric piecewise-affine approximation of nonlinear systems: A cut-based approach," *Proceedings of the 22nd IFAC World Congress*, Yokohama, Japan, pp. 6666–6671, July 2023. doi:10.1016/j.ifacol.2023.10.369

Delft Center for Systems and Control
Delft University of Technology
Mekelweg 2, 2628 CD Delft
The Netherlands
phone: +31-15-278.24.73 (secretary)
URL: https://www.dcsc.tudelft.nl

* This report can also be downloaded via https://pub.bartdeschutter.org/abs/23_027.html

# Parametric Piecewise-Affine Approximation of Nonlinear Systems: A Cut-Based Approach ⋆

**Leila Gharavi** * **Bart De Schutter** * **Simone Baldly** **

* *Delft Center for Systems and Control, Delft University of Technology, Delft, Netherlands, (email: {L.Gharavi, B.DeSchutter}@tudelft.nl).*
** *School of Mathematics, Southeast University, Nanjing, China, (e-mail: S.Baldi@tudelft.nl)*

**Abstract:** Piecewise-affine (PWA) approximations are widely used among hybrid modeling frameworks as a way to increase computational efficiency in nonlinear control and optimization problems. A variety of approaches to construct PWA approximations have been proposed, most of which are tailored to specific application areas by using some prior knowledge of the system in their assumptions and/or steps. In this paper, a parametric method is proposed to identify PWA approximations of nonlinear systems, without any prior knowledge of their dynamics or application requirements. The algorithm defines the regions parametrically using hyperplanes to cut the domain, and increases the number of regions iteratively until a user-defined error tolerance criterion is met. General remarks are given on the algorithm's implementation and a case study is provided to illustrate its application to vehicle dynamics.

*Keywords:* Hybrid and switched systems modeling, Modeling and control of hybrid systems, Piecewise-affine systems, Piecewise-affine approximation, Parametric system approximation.

## 1. INTRODUCTION

The literature on hybrid systems provides analysis and control synthesis methodology for systems featuring interacting continuous and discrete dynamics. To do so, a variety of modeling frameworks have been proposed for hybrid systems, as well as proof of their equivalence (Heemels et al., 2001), such as PWA, Mixed-Logical-Dynamical (MLD), Max-Min-Plus-Scaling (MMPS), and linear complementary systems (Lunze and Lamnabhi-Lagarrigue, 2009). Several of these frameworks have been extensively studied, including for control (De Schutter et al., 2020) and reachability analysis (Cândido et al., 2018) of MMPS systems, model predictive control design (Bemporad and Morari, 1999) and its explicit solution in some cases (Oberdieck and Pistikopoulos, 2015) for MLD, as well as for continuous PWA systems (De Schutter and van den Boom, 2004).

Among all the hybrid modeling frameworks, PWA systems have received extensive attention due to their simple, yet clear, formulation of the hybrid nature of the system behavior (i.e. explicit representation of different dynamics and their activation criteria). For example, the performance of discrete-time PWA systems (Ferrari-Trecate et al., 2002), their stability criteria in presence of uncertainty (Hovd and Olaru, 2018), their periodic solutions (Sessa et al., 2016), and bifurcation phenomena (Ito et al., 2016) were analyzed.

The PWA formalism is not only applied in domains where the hybrid nature of the system is important, but it has also been extensively utilized in a wide range of problems to increase computational efficiency, such as modeling prostate-specific antigen levels (Suzuki and Aihara, 2013), water motion in sewer networks (Joseph-Duran et al., 2014), or cornering behavior in vehicles (Sun et al., 2020). In some cases, PWA approximation of a nonlinear model facilitates reduction of the nonlinear control optimization problem into a mixed-integer programming one, while still capturing the complexity of the nonlinear behavior.

There are two main aspects to the problem of finding a PWA approximation: optimal partitioning of the state space into regions, and finding the optimal affine approximation in each one. The shape and the number of the regions influence computational complexity, the accuracy, and potential numerical issues of the final form. A higher number of regions improves accuracy and reduces the error bound, but leads to computationally more complex control problems. In addition, the shape and edge of the regions are of importance as the optimization problem is most likely to encounter numerical problems, if e.g. regions have redundant edges or gaps exist between them.

In some applications, a proper partitioning strategy is known based on heuristics or physics-based knowledge of the system (Zheng and Shyrokau, 2019; Jagga et al., 2018). In such cases, finding local affine approximations is more straightforward and can be achieved using least-squares or other regression methods. However, a generic PWA-approximation optimization problem is combined, i.e. both regions and local approximations are decision variables. Some techniques have been proposed to tackle challenges due to the combined nature of the problem, like partitioning the domain based on the variations of the nonlinear function (Azuma et al., 2010), learning-based PWA system identification using recursive adaptive control laws (Kersting and Buss, 2019) and online observer-based identifiers (Du et al., 2021), or clustering approaches, either based on convex relaxation of sparse optimization prob-

⋆ This research is funded by the Dutch Science Foundation NWO-TTW within the EVOLVE project (no. 18484). The third author was also partly supported by the National Key R&D Program of China no.2022YFE0198700, and by the Natural Science Foundation of China nos. 62150610499, 62073074.

lems (Bako, 2014) or incorporating fuzzy-based outlier rejection and k-means method (Khanmirza et al., 2016).

To date, many of the developed techniques, either explicitly or implicitly, limit the application to low dimensions or a bound on the number of local dynamics/modes (Thuan et al., 2016), and many require some prior knowledge of the PWA approximation to be found. e.g. by employing some heuristic clustering steps (Hartmann et al., 2015). Evidently, the effectiveness of the method depends upon the application area and its requirements; the cited papers have successfully found computationally efficient PWA models for their respective systems. However, to the best of our knowledge, no method has been proposed that addresses generic PWA approximation of a system, without taking specific dimensions, applications, or assumptions into account.

In this paper, we propose an iterative algorithm to find PWA approximations of nonlinear systems satisfying a user-defined error tolerance. Our proposed approach solves combined optimization problems in each iteration where parametric hyperplanes are used to cut the domain into different regions. This results in parametric definition of regions, which are then directly optimized as a subset of the decision variables. As the algorithm assumes no prior knowledge of the system, it can be implemented for discrete-time and continuous-time dynamics, as well as event-driven and time-driven dynamics, in a wide range of application areas. In any case, the algorithm can still be simplified, curtailed, or easily modified if any information on the system is available. Details of the algorithm and parametric region definition are described in Section 2, accompanied by general remarks on various steps and considerations. The algorithm is then tested using a nonlinear vehicle model as a case study in Section 3. Finally, concluding remarks and suggestions for future work are given in Section 4.

## 2. PWA APPROXIMATION

### 2.1 Problem Formulation

Consider a given nonlinear system with its dynamics expressed in the generic form

$$\dot{s} = F(s,u),$$

where $s \in \mathbb{R}^n$ and $u \in \mathbb{R}^m$ respectively represent the state and input vectors and $F : \mathbb{R}^{m+n} \to \mathbb{R}^n$ is the nonlinear function to be approximated. Without loss of generality, the augmented state vector $x = [s^T \; u^T]^T$ is used to define $F(x) := F(s,u)$ since the approximated function will be selected to be affine in both the state and the input. The augmented domain is assumed to be bounded and will be defined as $\mathrm{dom}(F) = \mathscr{D} \subset \mathbb{R}^{m+n}$.

The proposed approach approximates the nonlinear function $F$ by a PWA function $f$ defined as

$$x \in \mathscr{C}_p \implies f(x) = f_p(x), \qquad f_p(x) = A_p x + B_p, \tag{1}$$

with $p \in \{1,2,\dots,P\}$, where $P$ is the number of regions, each defined by polytope $\mathscr{C}_p \subseteq \mathbb{R}^{m+n}$ with

$$\mathscr{C}_p \neq \emptyset, \tag{2}$$

$$\mathrm{int}(\mathscr{C}_p) \cap \mathrm{int}(\mathscr{C}_q) = \emptyset, \tag{3}$$

$$\bigcup_{p=1}^{P} \mathscr{C}_p = \mathscr{D}, \tag{4}$$

for

$$\forall p,q \in \{1,2,\dots,P\}, \qquad p \neq q,$$

to form a partition of $\mathscr{D}$, with $\mathrm{int}(\mathscr{C}_p)$ denoting the interior of region $\mathscr{C}_p$. By defining the border hyperplanes $L_{p,q} \subset \mathbb{R}^{m+n-1}$ as

$$L_{p,q} = \mathscr{C}_p \cap \mathscr{C}_q, \qquad \forall p,q \in \{1,2,\dots,P\},\ p \neq q, \tag{5}$$

the set of border hyperplanes forming boundaries of the region $\mathscr{C}_p$ are represented by the set

$$\mathscr{L}_p = \{L_{p,q} \mid q \in \{1,2,\dots,P\} \wedge q \neq p\}.$$

For a fixed $P$, both the regions $\mathscr{C}_p$ and the corresponding local affine approximations $f_p$ are obtained by finding the optimal values of the matrices $A_p$ and $B_p$, as well as the set $\mathscr{L}_p$ so as to minimize the squared approximation error. This is implemented by solving the optimization problem

$$\min_{A_p \in \mathscr{A},\ B_p \in \mathscr{B},\ \mathscr{L}_p \in \mathscr{L}} \int_{\mathscr{D}} \frac{\|F(x) - f(x)\|_2^2}{\|F(x)\|_2^2 + 1}\, dx, \qquad \text{s.t. } (1)-(4), \tag{6}$$

where $\mathscr{A}$, $\mathscr{B}$, and $\mathscr{L}$ represent the sets containing $A_p$, $B_p$, and $\mathscr{L}_p$, respectively. The term $\|F(x)\|_2^2$ in the denominator is introduced such that the cost values represent the relative error and the added 1 prevents division be very small values where $\|F(x)\|_2 \approx 0$.

### 2.2 Parametric Definition of Regions

Without loss of generality, $m+n$ is assumed to an even [1] number as $m+n = 2d$, and the states are paired in couples as $(x_i, x_j)$ to form 2-dimensional subspaces. The corresponding pairs $(i,j)$ are collected in the set $\Omega$ and the local domains $\mathscr{D}_{i,j} \subset \mathbb{R}^2$, are defined as

$$x \in \mathscr{D} \implies [x_i \; x_j]^T \in \mathscr{D}_{i,j}, \qquad \forall (i,j) \in \Omega.$$

After pairing the states, the regions are defined by cutting $\mathscr{D}$ perpendicular to the $(x_i, x_j)$ planes as shown in Fig. 1. Since the region boundaries are to be optimized, the place of the cuts needs to be defined parametrically. To do so, two carrier lines are introduced on opposite sides of $\mathscr{D}_{i,j}$, on which points $\alpha_{i,j}$ and $\beta_{i,j}$ can slide. As an example, Fig. 1 shows three points (in yellow and orange) sliding on the carriers, where the lines connecting the pairs [2] of $(\alpha_{i,j}, \beta_{i,j})$ are used to cut the domain $\mathscr{D}$ perpendicular to $\mathscr{D}_{i,j}$.

*Remark 1.* Given $l_\alpha$ and $l_\beta$, the location of the points $\alpha_{i,j}$ and $\beta_{i,j}$ on the carriers can be obtained as

$$\begin{aligned} x_{i\alpha} &= x_{i_{\min}} + X_i \sin^2\phi + l_\alpha \cos\phi, \\ x_{j\alpha} &= x_{j_{\min}} + X_i \sin\phi\cos\phi + l_\alpha \sin\phi, \\ x_{i\beta} &= x_{i_{\min}} + X_i \sin^2\phi + l_\beta \cos\phi, \\ x_{j\beta} &= x_{j_{\min}} + X_i \sin\phi\cos\phi + l_\beta \sin\phi, \end{aligned} \tag{7}$$

where the domain parameters $X$, $\phi$, and $x_{\min}$ associated with the $i$ and $j$ axes are shown in Fig. 2.

To cover all possible cutting angles, the two cases [3] in Fig. 1 should be investigated separately with different carriers. For a rectangular $\mathscr{D}_{i,j}$ (e.g. due to bound constraints), or a parallelogram, it is convenient to define the carriers for $\alpha_{i,j}$ and $\beta_{i,j}$

[1] It should be noted that this assumption will not pose any restrictions on the method since for an odd $(m+n)$ value, the cutting procedure can be easily implemented on the unpaired single dimension as an axis.

[2] The $\alpha_{i,j}$ values should be increasing and the same holds for the $\beta_{i,j}$ values, since otherwise the corresponding cuts collide.

[3] Requiring two cases stems from the $(x_i, x_j)$ plane being 2-dimensional.

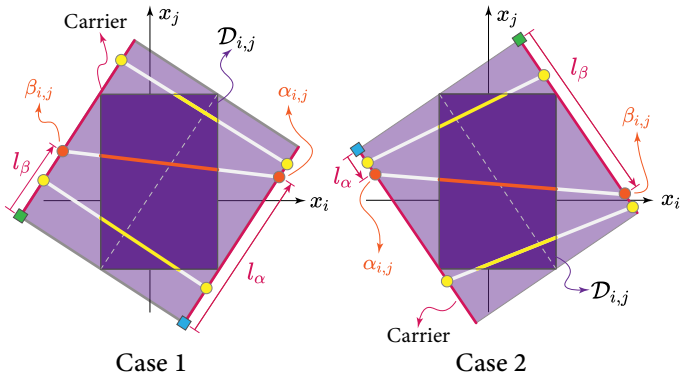


Fig. 1. Parametric definition of cutting the domain; two cases are proposed to cover all the cutting angles within the local domain.

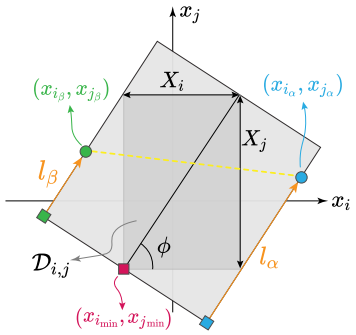


Fig. 2. A schematic view of the connection among various domain parameters and their relation to the decision variables, i.e. domain cuts.

points parallel to one of the diagonals in each case. Nevertheless, this concept can be easily extended for applications with other $\mathscr{D}_{i,j}$ forms by circumscribing a parallelogram to $\mathscr{D}_{i,j}$ and defining the carriers parallel to the diagonals for each case. The numbers of cuts perpendicular to each $(x_i, x_j)$ plane, is denoted by $n_{c_{i,j}}$ and it is equal to the number of $\alpha_{i,j}$ and $\beta_{i,j}$ points sliding on the carriers.

### 2.3 Approximation Algorithm

As the optimal number of regions is not known a priori, our proposed algorithm tackles PWA approximation through an iterative loop given in Algorithm 1. The vector $n_c$ contains the number of cuts $n_{c_{i,j}}$ with indices $i$ and $j$ such that $(x_i, x_j) \in \Omega$. By getting $n_c$ as input and solving (6) for a fixed $P$ as

$$P = \prod_{(i,j)\in\Omega} \left(n_{c_{i,j}} + 1\right),$$

the function `reg_optimization` finds the optimal affine approximations and their corresponding regions simultaneously. During each iteration, `reg_optimization` returns both the minimum objective $J^*$ and its corresponding optimal decision variables

$$\nu^* = (\mathscr{A}^*,\ \mathscr{B}^*,\ \mathscr{L}^*),$$

as output. The asterisk indicates the optimal value of the variable, and the border hyperplanes are defined using the position of the $\alpha_{i,j}$ and $\beta_{i,j}$ points as

$$\mathscr{L} = \left\{(l_{\alpha_k}, l_{\beta_k}) \mid \kappa \in \{1,2,\ldots,d\}\ ,\ k \in \{1,2,\ldots,(n_c)_\kappa\}\right\}.$$

It should be noted that (6) is a nonlinear optimization problem. Therefore, `reg_optimization` can either use a global search solver such as genetic algorithm or particle swarm, or gradient-based approaches with multiple starting points. In both cases, the best objective value would be the lowest value among the minima obtained in each trial.

**Algorithm 1** Iterative cut-based PWA approximation

```
cond ← true
n_c ← 0_{1×d}
iter ← 0
while cond do
    iter ← iter +1
    n_cuts ← 1_{d×1} × n_c + I_{d×d}
    for i ∈ {1,2,...,d} do
        n_in ← d-th row of n_cuts
        [err(i), sol(i)] ← reg_optimization(n_in)
    end for
    d_best ← arg min_i (err (i))
    n_c ← d_best-th row of n_cuts
    if min (err) ⩽ tol_err then
        cond ← false
        return n_c, sol (d_best)
    else if ∏_{q=1}^{d} n_c(q) ⩾ N_reg_max then
        cond ← false
        return print('Exceeded N_reg')
    else if iter ⩾ iter_max then
        cond ← false
        return print('Exceeded iter_max')
    end if
end while
```

However, the number of regions may not be sufficient to approximate the nonlinear function within a particular error bound. In that case, more cuts should be introduced to partition $\mathscr{D}$. To do so, the designed loops runs as follows to investigate different scenarios: in each iteration, `reg_optimization` is solved for $d$ cases, in which only one element in $n_c$ is increased by 1, and the $n_c$ with the lowest objective is selected as the best cutting strategy for the next iteration. The algorithm stops when reaching objective values below the error tolerance $\text{tol}_{\text{err}}$. To avoid an infinite loop, the procedure can also be stopped by passing maximum bounds on the number of iterations or the number of regions.

*Example.* For $\mathscr{D} \subset \mathbb{R}^4$, $d = 2$, and $\Omega = \{(1,2),(3,4)\}$, the algorithm starts by setting $n_c = [0\ 0]$, which means no cutting. In the first iteration, `reg_optimization` is called twice, finding the best approximations for $n_c = [1\ 0]$ and $n_c = [0\ 1]$ which correspond respectively to making only one cut perpendicular to $\mathscr{D}_{1,2}$, and only one cut perpendicular to $\mathscr{D}_{3,4}$. If $n_c = [1\ 0]$ gives a lower objective, but fails to satisfy the error tolerance, the next iteration starts with $n_c = [1\ 0]$, and two cases $n_c = [2\ 0]$ and $n_c = [1\ 1]$ are investigated. In other words, if one cut on $\mathscr{D}_{1,2}$ is considered a successful cutting strategy, the next step is to improve the result by adding more cuts to it as a baseline.

### 2.4 General Remarks

The power of Algorithm 1 stems from neither posing limits on system dimensions nor assuming a required number of regions. Moreover, as the approximation problem (6) can be solved by gridding the domain, our proposed method can also be applied in cases where the analytical form of the nonlinear model is not available. For instance, training measurement data sets can also

be used to find a fitted PWA approximation using Algorithm 1. In addition, some general notes should be made:

- PWA approximation of $F : \mathbb{R}^{m+n} \to \mathbb{R}^n$ is done by running Algorithm 1 independently for each of the $n$ states. This leads to the cuts and subsequently regions that are independently defined and evaluated for each component of $\dot{s}$. If this is not convenient for certain applications and it is desired to have the same regions for all the elements of $\dot{s}$, Algorithm 1 can be used in the same fashion or modified by changing the objective in (6) as

$$\min_{A_p \in \mathscr{A},\, B_p \in \mathscr{B},\, \mathscr{L}_p \in \mathscr{L}} \int_{\mathscr{D}} \frac{\|W(F(x) - f(x))\|_2^2 + 1}{\|F(x)\|_2^2}\, dx, \quad \text{s.t. } (1)-(4),$$

  where $W$ is a weight matrix.
- Implementing the proposed approach is completed by running Algorithm 1 for cases 1 and 2 in parallel and choosing the best result. However, one of the two cases may always be showing better results from the first iteration. To avoid unnecessary computation in such instances, the cases can be tested and compared by running the first iteration of Algorithm 1, identifying the better case (i.e. with a lower objective), and then implementing Algorithm 1 only for that case.
- Pairing the states as $\Omega$ can be done arbitrarily. Prior knowledge of the system and/or its application may suggest that specific states should be paired. Nevertheless, the pairing can be also done by testing different combinations of $\Omega$ through one iteration, as was proposed for evaluating cases 1 and 2.
- The proposed algorithm assumes the domains ($\mathscr{D}$ and subsequently $\mathscr{D}_{i,j}$) to be bounded. In case of an unbounded domain, a subset of the regions $\mathscr{C}_p$ need to be defined unbounded as well. This will not affect the decision variables in (6) as the cutting places are optimized, not the regions' boundaries. However, the objective in (6) approaches infinity across an unbounded domain. To avoid this, a sufficiently large bounded subset of the unbounded domain $\mathscr{D}$ can be used to find the PWA approximation using our algorithm. The result can then be directly used to approximate the behavior in the original domain.
- The matrix form of the border hyperplanes obtained from (5) can be constructed by extending the definition of the cuts. Using (7), a cut $L$ is defined by

$$L := x_j = \left(\frac{x_{j\alpha} - x_{j\beta}}{x_{i\alpha} - x_{i\beta}}\right) x_i + \left(x_{j\alpha} - x_{i\alpha}\frac{x_{j\alpha} - x_{j\beta}}{x_{i\alpha} - x_{i\beta}}\right).$$

  As each pair of cuts from different $\mathscr{D}_{i,j}$ are perpendicular, the resulting cutting hyperplanes in $\mathscr{D}$ can be directly combined in a generic matrix form

$$L_{p,q} := Hx + h = 0.$$

## 3. CASE STUDY: VEHICLE DYNAMICS

In this section, Algorithm 1 is used to find a PWA approximation of a nonlinear model of vehicle dynamics, integrating the coupled longitudinal and lateral dynamics in a single-track configuration, and considering linear tire forces. The model and implementation of the proposed approach are explained in the following sections.

### *3.1 Nonlinear Vehicle Model*

A single-track representation of the vehicle is shown in Fig. 3. With the system variables and parameters respectively defined in Tables 1 and 2, the nonlinear vehicle model is described by the following equations:

$$\dot{v}_x = \frac{1}{m}\left[F_{xf}\cos\delta - F_{yf}\sin\delta + F_{xr}\right] + v_y r, \tag{8}$$

$$\dot{v}_y = \frac{1}{m}\left[F_{xf}\sin\delta + F_{yf}\cos\delta + F_{yr}\right] - v_x r, \tag{9}$$

$$\dot{r} = \frac{1}{I_{zz}}\left[F_{xf}\sin\delta\, l_f + F_{yf}\cos\delta\, l_f - F_{yr}\, l_r\right], \tag{10}$$

and the lateral forces are given by the linear tire model

$$F_{yf} = C_{\alpha_f}\alpha_f, \qquad F_{yr} = C_{\alpha_r}\alpha_r,$$

where the slip angles are obtained by

$$\alpha_f = \delta - \tan^{-1}\left(\frac{v_y + l_f r}{v_x}\right), \quad \alpha_r = \tan^{-1}\left(\frac{v_y - l_r r}{v_x}\right).$$

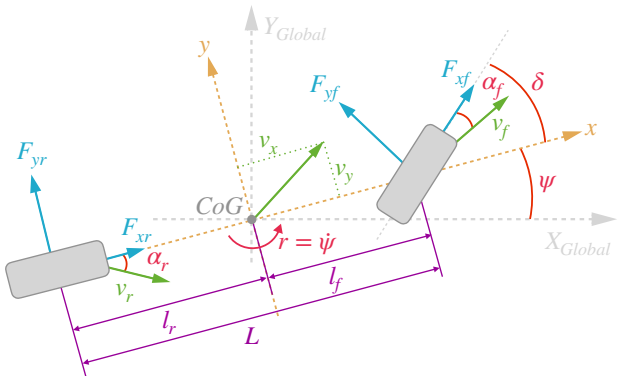


Fig. 3. Configuration of the single-track vehicle model

Table 1. System Variables

| Var. | Definition | Unit |
|---|---|---|
| $v_x$ | Longitudinal velocity | m/s |
| $v_y$ | Lateral velocity | m/s |
| $\psi$ | Yaw angle | rad |
| $r$ | Yaw rate | rad/s |
| $\delta$ | Steering angle (road) | rad |
| $F_{xf}$ | Longitudinal force on the front axis | N |
| $F_{xr}$ | Longitudinal force on the rear axis | N |
| $F_{yf}$ | Lateral force on the front axis | N |
| $F_{yr}$ | Lateral force on the rear axis | N |
| $F_{zf}$ | Normal load on the front axis | N |
| $F_{zr}$ | Normal load on the rear axis | N |
| $\alpha_f$ | Front slip angle | rad |
| $\alpha_r$ | Rear slip angle | rad |

Table 2. System Parameters

| Par. | Definition | Value | Unit |
|---|---|---|---|
| $m$ | Vehicle mass | 1970 | kg |
| $I_{zz}$ | Inertia moment about z-axis | 3498 | kg/m$^2$ |
| $l_f$ | CoG* to front axis distance | 1.4778 | m |
| $l_r$ | CoG to rear axis distance | 1.4102 | m |
| $C_{\alpha_f}$ | Front cornering stiffness | 126784 | N |
| $C_{\alpha_r}$ | Front cornering stiffness | 213983 | N |

*Center of Gravity

### 3.2 Implementation and Results

Considering system dynamics in (8) to (10), Algorithm 1 is used to find PWA approximation of $\dot{v}_x$, $\dot{v}_y$, and $\dot{r}$ independently. MATLAB's Optimization toolbox is used to implement the algorithm using `lsqnonline` for 10 starting points. The system is simulated during an evasive double lane-change maneuver and the axes corresponding to the augmented state vector $x = [v_x \; v_y \; r \; F_{xf} \; F_{xr} \; \delta]^T$ are paired as

$$\Omega = \{(v_x, r), (v_y, \delta), (F_{xf}, F_{xr})\},$$

which results from our physics-based knowledge of the system states, their dimensions, and their order of magnitude. Comparing the first iterations of cases 1 and 2 showed that case 2 gives lower objectives when cutting $\mathscr{D}$ perpendicular to $\mathscr{D}_{v_x,r}$ and $\mathscr{D}_{v_y,\delta}$, while case 1 is the better one to define cuts on $D_{F_{xf},F_{xr}}$.

The solution time depends on the number of regions due to an subsequent increase in the number of decision variables. The algorithm was run for different error tolerances using the DelftBlue supercomputer, at the Delft High Performance Computing Centre (DHPC) with every iteration for the number of regions between 2 to 10 taking on average 435 minutes.

The approximations obtained for $\text{tol}_{\text{err}}$ values in Table 3 using our proposed cut-based algorithm (CB), and the Lebesgue PWA approximation (LB) approach proposed by Azuma et al. (2010), have been compared with the nonlinear system for the open-loop system simulation in Fig. 4. In the LB approach, the domain is partitioned perpendicular to each axis and based on variation of the nonlinear function's gradient; this results in hypercubic regions. However, the CB approach cuts the domain perpendicular to 2-dimensional subspaces which leads to polytopic regions. The same tolerances were selected for both algorithms for fair comparison, and they converged to the number cuts $n_c$ defined as

$$n_c = \left[ n_{c_{(v_x,r)}}, \, n_{c_{(v_y,\delta)}}, \, n_{c_{(F_{xf},F_{xr})}} \right].$$

The total number of regions $N$ is listed as well in Table 3. Fig. 4 shows that the CB approach provides a more accurate approximation of the model, and its good performance is better seen in $\dot{v}_x$ which has a higher degree of nonlinearity where CB gives a better approximation while introducing a smaller number of regions.

Table 3. The number of cuts at convergence for case study instances with different error tolerance values

| **Instance** | | $\dot{\mathbf{v}}_\mathbf{x}$ | $\dot{\mathbf{v}}_\mathbf{y}$ | $\dot{\mathbf{r}}$ |
|---|---|---|---|---|
| | $\mathbf{tol_{err}}$ | 0.30 | 0.10 | 0.05 |
| **PWA – LB** | $\mathbf{n_c}$ | [1,3,0] | [0,0,0] | [0,0,0] |
| | **N** | 8 | 1 | 1 |
| **PWA – CB** | $\mathbf{n_c}$ | [0,3,0] | [1,0,0] | [1,0,0] |
| | **N** | 4 | 2 | 2 |

## 4. CONCLUSIONS

In this paper, an iterative algorithm for PWA approximation of nonlinear systems was proposed assuming no prior knowledge of the application area. By using a cut-based parametric definition of the regions in the optimization problem, the algorithm aims at finding an optimal partitioning of the domain into polytopic regions and the corresponding local affine approximations, simultaneously. This combined optimization problem is solved in each iteration for several cases of adding new cuts whereas the number of cuts is increased in each iteration until a user-specified error tolerance is reached. The algorithm is implemented on a nonlinear vehicle model as a case study where different error tolerances were selected for each state and the results were compared to another PWA approximation approach from the literature, where similar to our proposed algorithm, the regions are included parametrically in the decision variables of the combined optimization problem. The comparison shows that our approach gives more a accurate approximation of the nonlinear system, in some cases with fewer number of regions.

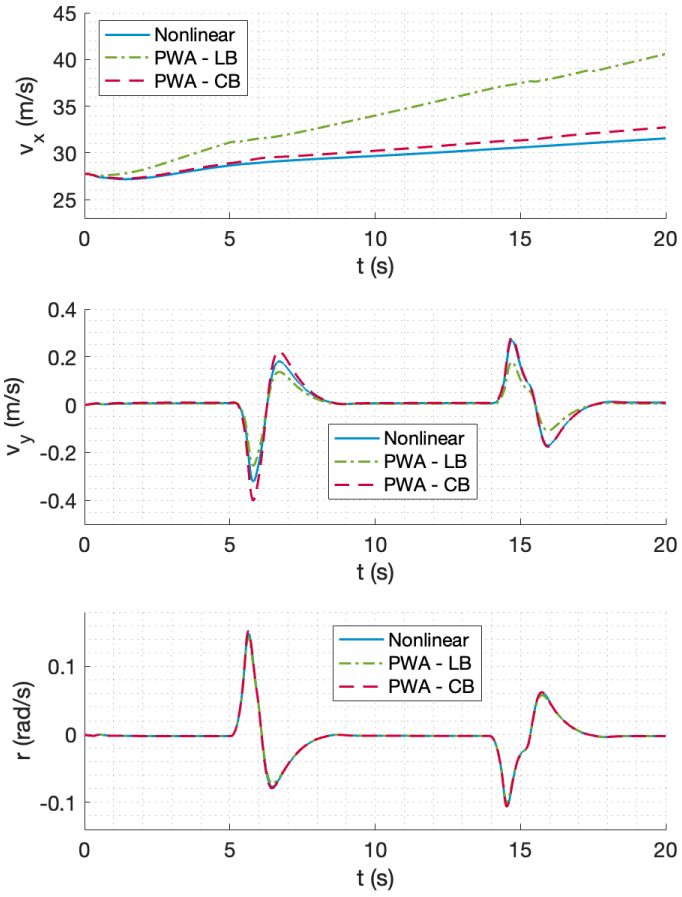


Fig. 4. Open-loop simulation of an evasive double lane-change maneuver using nonlinear vehicle model and two PWA approximations: LB and CB approaches

In future work, the current algorithm can be improved along two lines. First, the iteration law can be enhanced for faster convergence to the optimal number of regions while avoiding introduction of extra and/or redundant cuts. For instance, instead of increasing the number of cuts in each iteration by one, more cuts can be introduced based on the difference of the objective functions between the last two iterations. Second, adjustments or additions to the algorithm structure can be introduced for applications where discontinuity is problematic, to either avoid discontinuity on the region borders in the obtained PWA approximation, or to circumvent its undesired consequences (e.g. in switching analysis or control synthesis) by defining auxiliary affine dynamics or switching rules along the borders. Moreover, on the application level we aim at investigating the performance of our proposed approximation method on a wider variety of test cases, i.e. driving scenarios.

## REFERENCES


Azuma, S., Imura, J., and Sugie, T. (2010). Lebesgue piecewise affine approximation of nonlinear systems. *Nonlinear Analysis: Hybrid Systems*, 4(1), 92–102.

Bako, L. (2014). Subspace clustering through parametric representation and sparse optimization. *IEEE Signal Processing Letters*, 21, 356–360.

Bemporad, A. and Morari, M. (1999). Control of systems integrating logic, dynamics, and constraints. *Automatica*, 35(3), 407–427.

Cândido, R.M.F., Hardouin, L., Lhommeau, M., and Mendes, R.S. (2018). Conditional reachability of uncertain max plus linear systems. *Automatica*, 94, 426–435.

De Schutter, B., van den Boom, T., Xu, J., and Farahani, S.S. (2020). Analysis and control of max-plus linear discrete-event systems: An introduction. *Discrete Event Dynamic Systems: Theory and Applications*, 30, 25–54.

De Schutter, B. and van den Boom, T.J. (2004). MPC for continuous piecewise-affine systems. *Systems and Control Letters*, 52(3-4), 179–192.

Delft High Performance Computing Centre (DHPC) (2022). *DelftBlue Supercomputer (Phase 1)*. ARK: ark:/44463/DelftBluePhase1.

Du, Y., Liu, F., Qiu, J., and Buss, M. (2021). Online identification of piecewise affine systems using integral concurrent learning. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 68, 4324–4336.

Ferrari-Trecate, G., Cuzzola, F.A., Mignone, D., and Morari, M. (2002). Analysis of discrete-time piecewise affine and hybrid systems. *Automatica*, 38(12), 2139–2146.

Hartmann, A., Lemos, J.M., Costa, R.S., Xavier, J., and Vinga, S. (2015). Identification of switched ARX models via convex optimization and expectation maximization. *Journal of Process Control*, 28, 9–16.

Heemels, W.P.M.H., De Schutter, B., and Bemporad, A. (2001). Equivalence of hybrid dynamical models. *Automatica*, 37(7), 1085–1091.

Hovd, M. and Olaru, S. (2018). Parameter-dependent PWQ Lyapunov function stability criteria for uncertain piecewise linear systems. *Modeling, Identification and Control*, 39, 15–21.

Ito, D., Ueta, T., Kousaka, T., and Aihara, K. (2016). Bifurcation analysis of the Nagumo-Sato model and its coupled systems. *International Journal of Bifurcation and Chaos*, 26.

Jagga, D., Lv, M., and Baldi, S. (2018). Hybrid adaptive chassis control for vehicle lateral stability in the presence of uncertainty. In *Mediterranean Conference on Control and Automation (MED 2018)*, 529–534.

Joseph-Duran, B., Jung, M.N., Ocampo-Martinez, C., Sager, S., and Cembrano, G. (2014). Minimization of sewage network overflow. *Water Resources Management*, 28, 41–63.

Kersting, S. and Buss, M. (2019). Recursive estimation in piecewise affine systems using parameter identifiers and concurrent learning. *International Journal of Control*, 92, 1264–1281.

Khanmirza, E., Nazarahari, M., and Mousavi, A. (2016). Identification of piecewise affine systems based on fuzzy PCA-guided robust clustering technique. *Eurasip Journal on Advances in Signal Processing*, 2016, 1–15.

Lunze, J. and Lamnabhi-Lagarrigue, F. (eds.) (2009). *Handbook of Hybrid Systems Control – Theory, Tools, Applications*. Cambridge University Press, Cambridge, UK.

Oberdieck, R. and Pistikopoulos, E.N. (2015). Explicit hybrid model-predictive control: The exact solution. *Automatica*, 58, 152–159.

Sessa, V., Iannelli, L., Vasca, F., and Acary, V. (2016). A complementarity approach for the computation of periodic oscillations in piecewise linear systems. *Nonlinear Dynamics*, 85, 1255–1273.

Sun, X., Hu, W., Cai, Y., Wong, P., and Chen, L. (2020). Identification of a piecewise affine model for the tire cornering characteristics based on experimental data. *Nonlinear Dynamics*, 101, 857–874.

Suzuki, T. and Aihara, K. (2013). Nonlinear system identification for prostate cancer and optimality of intermittent androgen suppression therapy. *Mathematical Biosciences*, 245, 40–48.

Thuan, L.Q., van den Boom, T., and Baldi, S. (2016). Online identification of continuous bimodal and trimodal piecewise affine systems. In *European Control Conference (ECC 2016)*, 1075–1070.

Zheng, Y. and Shyrokau, B. (2019). A real-time nonlinear MPC for extreme lateral stabilization of passenger vehicles. In *IEEE International Conference on Mechatronics (ICM 2019)*, 519–524.