Template-Based Piecewise Affine Regression

Guillaume O. Berger and **Sriram Sankaranarayanan** FIRST.LASTNAME@COLORADO.EDU
University of Colorado Boulder, USA.

Editors: R. Firoozi, N. Mehr, E. Yel, R. Antonova, J. Bohg, M. Schwager, M. Kochenderfer

Abstract

We investigate the problem of learning piecewise affine functions (PWA) from data. Our algorithm divides the input domain into finitely many regions whose shapes are specified using a user-defined template such that the data points within each region are fit by an affine function with a desired bound on the error. We first prove that this problem is NP-hard. Next, we present a top-down algorithm that considers subsets of the overall data set in a systematic manner, trying to fit an affine function for each subset using linear regression. If regression fails on a subset, we extract a minimal set of points that led to a failure in order to split the original index set into smaller subsets. Using a combination of this top-down scheme and a set covering algorithm, we derive an overall approach that is optimal in terms of the number of pieces of the resulting PWA model. We demonstrate our approach on two interesting numerical examples that include PWA approximations of a widely used nonlinear insulin–glucose regulation model and a double inverted pendulum with soft contacts.

Keywords: Piecewise Affine Regression, Hybrid System Identification.

1. Introduction

Piecewise affine (PWA) regression seeks to fit a piecewise affine function to a given set of data points consisting of input–output pairs wherein we simultaneously partition the input domain into finitely many regions while associating each region with an affine function. In this paper, we seek a PWA model that fits the given data while keeping the error within some user-provided limit ϵ and keeping the number of pieces as small as possible. This problem has numerous applications including the identification of hybrid systems with state-based switching and simplifying nonlinear models using PWA approximations.

Existing PWA regression approaches do not restrict how the input domain is to be partitioned. For instance, many approaches simply assume that the inputs are partitioned into convex sets (Lauer and Bloch, 2019). This can be detrimental for the computational complexity. Furthermore, it is often desirable to specify the possible shapes of these partitions. Therefore, we introduce the problem of PWA regression wherein the domain is partitioned into regions defined by a user-provided template. Such a template specifies each face of the region so that various regions may be defined by varying the constant offset for each face.

We show that, similarly to the problem of classical PWA regression (Lauer and Bloch, 2019), the problem of template-based PWA regression is NP-hard in the dimension of the input space and the complexity of the template, but polynomial in the size of the data set (Section 3).

Next, we provide an algorithm to compute template-based PWA regression with a minimal number of pieces (Section 4). The main idea is to examine various subsets of the input data in order to discover maximal subsets that are *compatible*: wherein compatibility of a set of data points simply means that there is an affine function that fits all the points within the desired error tolerance. Thus,

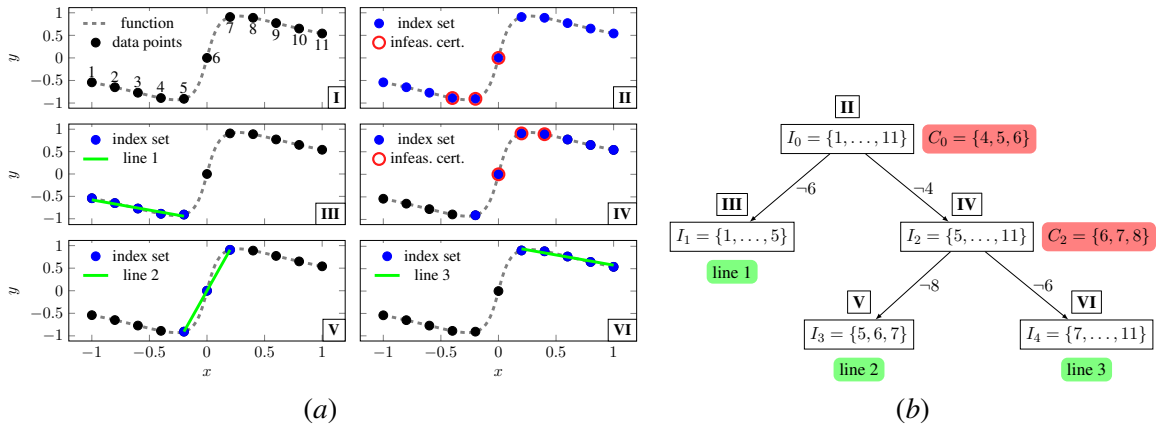
our approach starts to examine subsets of the data starting from the entire data to begin with. If a given subset is not compatible, we exploit the optimization formulation of the regression problem to extract a minimal subset of points that is itself incompatible. The key observation is that the original set can now be broken up into subsets which can themselves be examined for compatibility. We show that by integrating this process with a minimal set cover algorithm, we can extract a partition with the smallest size that in turn leads to the desired PWA model.

We apply our framework on two practical problems: the approximation of a nonlinear system, namely the insulin–glucose regulation process (Dalla Man et al., 2007), with affine functions with rectangular domains (Subsection 5.1), and the identification of a hybrid linear system consisting in an inverted double pendulum with soft contacts on the joints (Subsection 5.2). For both applications, we show that template-based PWA regression is favorable compared to classical PWA regression both in terms of computation time and our ability to formulate models from the results.

1.1. Related work

Piecewise affine systems and hybrid linear systems appear naturally in a wide range of applications (Jungers, 2009), or as approximations of more complex systems (Breiman, 1993). Therefore, the problems of switched affine (SA) and piecewise affine (PWA) regression have received a lot of attention in the literature; see, e.g., Paoletti et al. (2007); Lauer and Bloch (2019) for surveys. Both problems are known to be NP-hard (Lauer and Bloch, 2019). The problem of SA regression can be formulated as a Mixed-Integer Program and solved using MIP solvers, but the complexity is exponential in the number of data points (Paoletti et al., 2007). Vidal et al. (2003) propose an efficient algebraic approach to solve the problem, but it is restricted to noiseless data. Heuristics to solve the problem in the general case include greedy algorithms (Bemporad et al., 2005), continuous relaxations of the MIP (Münz and Krebs, 2005), block–coordinate descent (similar to k-mean regression) algorithms (Bradley and Mangasarian, 2000; Lauer, 2013) and refinement of the algebraic approach using sum-of-squares relaxations (Ozay et al., 2009); however, these methods offer no guarantees of finding an (optimal) solution to the problem. As for PWA regression, classical approaches include clustering-based methods (Ferrari-Trecate et al., 2005), data classification followed by geometric clustering (Nakada et al., 2005) and block–coordinate descent algorithms (Bemporad, 2022); however, these methods are not guaranteed to find a (minimal) piecewise affine model.

Piecewise affine systems with constraints on the domain appear naturally in several applications including biology (Porreca et al., 2009) and mechanical systems with contact forces (Aydinoglu et al., 2020), or as approximations of nonlinear systems (Smarra et al., 2020). Techniques for PWA regression with rectangular domains have been proposed in Münz and Krebs (2002); Smarra et al. (2020); however, these approaches impose further restrictions on the arrangement of the domains of the functions (e.g., forming a grid) and they are not guaranteed to find a solution with a minimal number of pieces. In the one-dimensional case (e.g., time series), an exact efficient algorithm for optimal PWA regression was proposed by Ozay et al. (2012), but the approach does not extend to higher dimension. As for the application involving mechanical systems with contact forces (presented in Subsection 5.2), a recent work by Jin et al. (2022) proposes a heuristic based on minimizing a loss function with an estimation accuracy term and a constraint-violation penalty term to learn *linear complementary systems*.



(*a*) (*b*)

Figure 1: (a) Illustration of our algorithm on a simple data set with 11 data points $(x_k, y_k) \in \mathbb{R} \times \mathbb{R}$ and (b) the index sets explored by our algorithm.

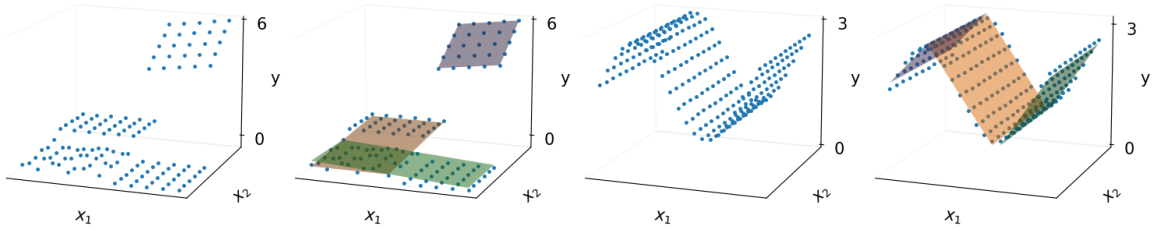
1.2. Approach at a glance

Figure 1 shows the working of our algorithm on a simple data set with $K = 11$ points $(x_k, y_k) \in \mathbb{R} \times \mathbb{R}$ (see plot "I"). The error tolerance is $\epsilon = 0.1$. We seek a piecewise affine (PWA) function that fits the data within the error tolerance ϵ with the smallest number of pieces possible. At the very first step, labelled "II" in the figure, the approach tries to fit a single straight line through all the 11 points. This corresponds to the *index set* $I_0 = \{1, \ldots, 11\}$ where the indices correspond to points as shown in the plot "I". However, no such line can fit the points for the given ϵ. Our approach generates an *infeasibility certificate* that identifies the indices $C_0 = \{4, 5, 6\}$ as a cause of this infeasibility (see plot "II"). In other words, we cannot have all three points in C_0 be part of the same piece of the PWA function we seek. Therefore, our approach now splits I_0 into two subsets $I_1 = \{1, \ldots, 5\}$ and $I_2 = \{5, \ldots, 11\}$. These two sets are maximal with respect to set inclusion and do not contain C_0. The set I_1 can be fit by a single straight line with tolerance ϵ (see plot "III"). However, considering I_2, we notice once again that a single straight line cannot be fit (see plot "IV"). We identify the set $C_2 = \{6, 7, 8\}$ as an infeasibility certificate and our algorithm splits I_2 into maximal subsets $I_3 = \{5, 6, 7\}$ and $I_4 = \{7, \ldots, 11\}$. Each of these subsets can be fit by a straight line (see plots "V" and "VI"). Thus, our approach finishes by discovering three pieces that cover all the points $\{1, \ldots, 11\}$. Note that although the data point indexed by 5 is part of two pieces, we can resolve this "tie" in an arbitrary manner by assigning 5 to the first piece and removing it from the second; the same holds for the data point indexed by 7.

2. Problem Statement

Given $K \in \mathbb{N}_{>0}$ observation data points $\{(x_k, y_k)\}_{k=1}^K \subseteq \mathbb{R}^d \times \mathbb{R}^e$ (see Figures 2*(a,c)*), we wish to find a piecewise affine (PWA) function that fits the data within some error tolerance $\epsilon \geq 0$. Formally, a PWA function over a domain $D \subseteq \mathbb{R}^d$ is defined by covering the domain with q regions $H_1, \ldots, H_q$ and associating an affine function $f_i(x) = A_i x + b_i$ with each H_i:

$$f(x) = A_1 x + b_1 \text{ if } x \in H_1, \ \ldots, \ A_i x + b_i \text{ if } x \in H_i, \ \ldots, \ A_q x + b_q \text{ if } x \in H_q.$$



(*a*) Data set $\mathcal{D}_1$ (*b*) $\mathcal{D}_1$ fit: $\epsilon = 0.375$ (*c*) Data set $\mathcal{D}_2$ (*d*) $\mathcal{D}_2$ fit: $\epsilon = 0.15$

Figure 2: Template-based piecewise affine (TPWA) regression. *(a)*, *(c)*: Data points $(x_k, y_k) \in \mathbb{R}^2 \times \mathbb{R}$. *(b)*, *(d)*: TPWA fit with rectangular domains and error tolerance ϵ.

If $H_i \cap H_j \neq \emptyset$ for $i \neq j$, then f is no longer a function. However, in such a case, we may "break the tie" by defining $f(x) = f_i(x)$ wherein $i = \min\{j \mid x \in H_j\}$.

Problem 1 (PWA regression) *Given data* $\{(x_k, y_k)\}_{k=1}^K$ *and an error bound* $\epsilon \geq 0$*, find* q *regions* $H_i \subseteq \mathbb{R}^d$ *and affine functions* $f_i(x) = A_i x + b_i$ *such that*

$$\forall\, k,\ \exists\, i : x_k \in H_i \quad \textit{and} \quad \forall\, k,\ \forall\, i,\ x_k \in H_i \Rightarrow \|y_k - f_i(x_k)\|_\infty \leq \epsilon. \tag{1}$$

Furthermore, we will restrict the domain H_i of each affine piece by specifying a template $p : \mathbb{R}^d \to \mathbb{R}^h$. Given a template p and a vector $c \in \mathbb{R}^h$, we define the set $H(c)$ as

$$H(c) = \{x \in \mathbb{R}^n : p(x) \leq c\}, \tag{2}$$

wherein $\leq$ is elementwise and $c \in \mathbb{R}^h$ parameterizes the set $H(c)$. Let $\mathcal{H} = \{H(c) : c \in \mathbb{R}^h\}$ denote the set of all regions in $\mathbb{R}^d$ described by the template p.

Fixing a template *a priori* controls the complexity of the domains, and thus of the overall PWA function. The *hyper-rectangular* template $p(x) = [x; -x]$ defines regions $H(c)$ that form boxes in $\mathbb{R}^d$. Similarly, allowing pairwise differences between individual variables as components of p yields the "octagon domain" (Miné, 2006). Figures 2*(b,c)* illustrate PWA functions with rectangular domains. Thus, we define the *template-based piecewise affine* (TPWA) regression problem:

Problem 2 (TPWA regression) *Given data* $\{(x_k, y_k)\}_{k=1}^K$*, a template* $p : \mathbb{R}^d \to \mathbb{R}^h$ *and an error bound* $\epsilon > 0$*, find* q *regions* $H_i \in \mathcal{H}$ *and affine functions* $f_i(x) = A_i x + b_i$ *such that* (1) *is satisfied.*

Problem 2 can be posed as a decision problem: given a bound $\hat{q}$, is there a TPWA function with $q \leq \hat{q}$ pieces; or as an optimization problem wherein we find a TPWA function with the minimum number of pieces. Although a solution to the decision problem can be used repeatedly to solve the optimization problem, we will focus on directly solving the optimization problem in this paper. Problem 2 is closely related to the well-known problem of *switched affine* (SA) regression, in which one aims to explain the data with a few affine functions, but there is no assumption on which function may explain a particular data point (x_k, y_k).

Problem 3 (SA regression) *Given data* $\{(x_k, y_k)\}_{k=1}^K$ *and an error bound* $\epsilon \geq 0$*, find* q *affine functions* $f_i(x) = A_i x + b_i$ *such that* $\forall\, k$*,* $\exists\, i$*:* $\|y_k - f_i(x)\|_\infty \leq \epsilon$*.*

3. Computational Complexity

The problem of SA regression (Problem 3) is known to be NP-hard, even for $q = 2$ (Lauer and Bloch, 2019, §5.2.4). In this section, we show that the same holds for the decision version of Problem 2. We study the problem in the RAM model, wherein the problem input size is $K(d+e)+\text{size}(p)$, where $\text{size}(p)$ is the size needed to describe the template p.

Theorem 4 (NP-hardness) *The decision version of problem 2 is NP-hard, even for $q = 2$ and hyper-rectangular templates.*

The proof reduces Problem 3 which is known to be NP-hard to Problem 2, and is provided in Appendix A. Despite the problem being NP-hard, one can show that for fixed dimension d, template $p : \mathbb{R}^d \to \mathbb{R}^h$ and number of pieces q, the complexity is polynomial in the size K of the data set. Note that a similar result holds for Problem 3 (Lauer and Bloch, 2019, Theorem 5.4).

Theorem 5 (Polynomial complexity in K) *For fixed dimension d, template $p : \mathbb{R}^d \to \mathbb{R}^h$ and number of pieces q, the complexity of Problem 2 is bounded by $O(K^{qh})$.*

To prove this theorem, let us first introduce some notation that will be useful in the rest of the paper. For every $c \in \mathbb{R}^h$, let $I(c) = \{k \in \mathbb{N} : 1 \leq k \leq K,\ x_k \in H(c)\}$ be the set of all indices k such that $x_k \in H(c)$. Also, let $\mathcal{I} = \{I(c) : c \in \mathbb{R}^h\}$ be the set of all such index sets.

Proof of Theorem 5: The crux of the proof is to realize that $|\mathcal{I}| \leq K^h + 1$.

For every $c \in \mathbb{R}^h$, define $P(c) = \{p(x_k) : 1 \leq k \leq K,\ p(x_k) \leq c\}$ and let $\mathcal{P} = \{P(c) : c \in \mathbb{R}^h\}$. It holds that $|\mathcal{P}| \leq K^h + 1$. Furthermore, there is a one-to-one correspondence between $\mathcal{P}$ and $\mathcal{I}$ given by: $P(c) \mapsto I(c)$. Indeed, it is clear that if $I(c_1) = I(c_2)$, then $P(c_1) = P(c_2)$. On the other hand, if $I(c_1) \nsubseteq I(c_2)$, then there is at least one k such that $p(x_k) \leq c_1$ but $p(x_k) \not\leq c_2$. This implies that $P(c_1) \nsubseteq P(c_2)$. Therefore, $|\mathcal{P}| = |\mathcal{I}| \in O(K^h)$.

Now, Problem 2 can be solved by enumerating the $L = K^h$ nonempty index sets $I_1, \ldots, I_L$ in $\mathcal{I}$, and keeping only those I_ℓ for which we can fit an affine function over the data $\{(x_k, y_k)\}_{k \in I_\ell}$ with error bound ϵ. Next, we enumerate all combinations of q such index sets that cover the indices $\{1, \ldots, K\}$. There are at most L^q such combinations. This concludes the proof of the theorem. ■

The algorithm presented in the proof of Theorem 5, although polynomial in the size of the data set, can be quite expensive in practice. For instance, in dimension $d = 2$, with rectangular regions ($h = 2d = 4$) and $K = 100$ data points, one would need to solve $K^h = 10^8$ regression problems,[1] each of which is a linear program. In the next section, we present an algorithm for TPWA regression that is generally several orders of magnitude faster by using a *top-down approach*.

4. Top-down Algorithm for TPWA Regression

We first define the concept of compatible and maximal compatible index sets.

1. Using Sauer–Shelah's lemma (Har-Peled, 2011, Lemma 6.2.2), this number can be reduced to $\sum_{i=1}^{h} \binom{K}{i} \approx 4 \times 10^6$. This is because the *VC dimension* of rectangular regions in dimension d is $2d$.

Algorithm 1: Top-down algorithm to compute maximal compatible index sets.

Data: Data set $\{(x_k, y_k)\}_{k=1}^K$, template p
Result: Collection $\mathcal{S}$ of all maximal compatible index sets
$\mathcal{S} \leftarrow \emptyset$ ("compatible"); $\mathcal{U} \leftarrow \{\{1, \dots, K\}\}$ ("to explore"); $\mathcal{V} \leftarrow \emptyset$ ("visited")
while $\mathcal{U} \setminus \mathcal{V}$ *is not empty* **do**
  Pick an index set I in $\mathcal{U} \setminus \mathcal{V}$
  if I *is compatible* **then**
    Add I to $\mathcal{S}$; Add to $\mathcal{V}$ all subsets of I; Remove from $\mathcal{S}$ all subsets of I
  else
    $(I_1, \dots, I_S) \leftarrow$ FINDSUBSETS(I) `// satisfies Definition 7`
    Add $I_1, \dots, I_S$ to $\mathcal{U}$; Add I to $\mathcal{V}$
  end
end
return $\mathcal{S}$

Definition 6 (Maximal compatible index set) *Consider an instance of Problem 2. An index set $I \subseteq \{1, \dots, K\}$ is* compatible *if (a) $I \in \mathcal{I}$ and (b) there is an affine function $f(x) = Ax + b$ such that $\forall\, k \in I$, $\|y_k - f(x_k)\|_\infty \leq \epsilon$. A compatible index set I is* maximal *if there is no compatible index set I' such that $I \subsetneq I'$.*

The key idea of the top-down approach is that one can restrict themselves to searching over *maximal* compatible index sets in order to find a solution to Problem 2. See Appendix B for a proof.

Maximal compatible index sets can be computed by using a recursive *top-down* approach (implemented in Algorithm 1): Consider the lattice $\mathcal{I}$ ordered by $\subseteq$ relationship. Our algorithm starts at the very top of this lattice and "descends" until we find maximal compatible index sets. At each step, we consider a current set $I \in \mathcal{I}$ (initially, $I = \{1, \dots, K\}$) that is a candidate for being compatible and check it for compatibility. If I is not compatible, we find subsets $I_1, \dots, I_S \subsetneq I$ using the FINDSUBSETS procedure, which is required to be *consistent*, as defined below.

Definition 7 (Consistency) *Given a non-compatible index set $I \in \mathcal{I}$, a collection of index sets $I_1, \dots, I_S \in \mathcal{I}$ is said to be* consistent *w.r.t. I if (a) for each s, $I_s \subsetneq I$ and (b) for every compatible index set $J \subseteq I$, there is s such that $J \subseteq I_s$.*

Theorem 8 (Correctness of Algorithm 1) *If* FINDSUBSETS *satisfies that for every non-compatible index set $I \in \mathcal{I}$, the output of* FINDSUBSETS(I) *is consistent w.r.t. I, then Algorithm 1 is correct, meaning that it terminates and the output $\mathcal{S}$ is the collection of all maximal compatible index sets.*

The proof is provided in Appendix C. The main idea of the proof is to show that at the start of each iteration of the main "while" loop: (a) any compatible index set J will be subsumed by some set in $\mathcal{S} \cup (\mathcal{U} \setminus \mathcal{V})$ and (b) every set in $\mathcal{S}$ is compatible. This can be shown by induction over the number of steps that the "while" loop runs.

We will now explain how to implement FINDSUBSETS, so that it is consistent. An incompatible set of points must necessarily have within it a subset that is itself incompatible. We will call such a subset an infeasibility certificate.

Algorithm 2: An implementation of FINDSUBSETS using infeasibility certificates

$C \leftarrow$ FINDCERTIFICATE(I), wherein $I = I(c)$, $c = [c^1, \ldots, c^h]$
foreach $s = 1, \ldots, h$ **do**
  $\hat{c}^s \leftarrow \max\{p^s(x_k) : k \in I,\ p^s(x_k) < \max_{\ell \in C}\ p^s(x_\ell)\}$
  Define $I_s = I([c^1, \ldots, c^{s-1}, \hat{c}^s, c^{s+1}, \ldots, c^h])$
end
return all nonempty index sets $I_1, \ldots, I_h$

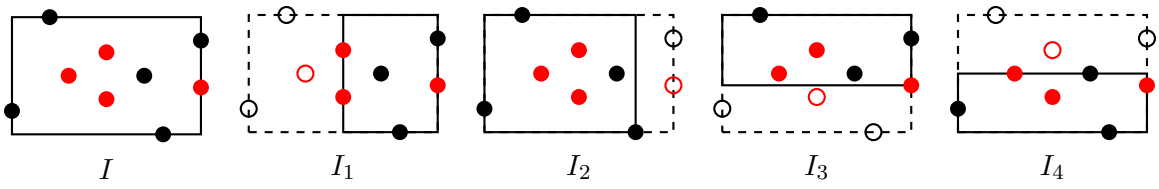


Figure 3: FINDSUBSETS implemented by Algorithm 2 with rectangular regions. The red dots represent the infeasibility certificate C. Each I_s excludes at least one point from C by moving one face of the box but keeping the others unchanged.

Definition 9 (Infeasibility certificate) *An index set $C \subseteq \{1, \ldots, K\}$ is an* infeasibility certificate *if there is no affine function $f(x) = Ax + b$ such that $\forall\, k \in C$, $\|y_k - f(x_k)\|_\infty \leq \epsilon$.*

From an infeasibility certificate, one can compute a consistent collection of index subsets by tightening each component of the template *independently*, in order to exclude a minimal nonzero number of indices from the infeasibility certificate, while keeping the other components unchanged. This implementation of FINDSUBSETS is provided in Algorithm 2. See also Figure 3 for an illustration with rectangular regions. The correctness of Algorithm 2 is proved in Appendix D.

Good infeasibility certificates Thus, the implementation of FINDSUBSETS boils down to finding infeasibility certificates. A trivial choice is to use I as infeasibility certificate (since it is not compatible). Although this is a valid choice, it will lead to an inefficient algorithm. To achieve efficiency, we seek infeasibility certificates of small cardinality. Using the *theorem of alternatives*[2] of Linear Programming, we obtain a certificate C that contains at most $d + 2$ data points. As a heuristic, we require that the points $\{x_k\}_{k \in C}$ are spatially concentrated (i.e, close to each other under some distance metric). Indeed, concentration of the points $\{x_k\}_{k \in C}$ around some center point $\bar{x}$ implies that at least one of the index sets $I_1, \ldots, I_S$ produced by Algorithm 2 is small compared to the original index set $I = I(c)$, because $\bar{x}$ cannot be tight at all components of $H(c)$; this can be seen in Figure 3 for rectangular regions. This approach is described in Appendix E.

4.1. Early stopping using set cover algorithms

Finally, Algorithm 1 can be made much more efficient by enabling early termination if $\{1, \ldots, K\}$ is optimally covered by the compatible index sets computed so far. For that, we add an extra step

2. This theorem states that if a set of linear inequalities in dimension n is not satisfiable, then there exists an *efficiently computable* subset of $n + 1$ of these inequalities that is not satisfiable (Rockafellar, 1970, Theorem 21.3).

Algorithm 3: Extra step at the beginning of each iteration of Algorithm 1

Data: $\mathcal{S}, \mathcal{U}$ and $\mathcal{V}$ at the iteration, K
Result: BREAK if we can extract from $\mathcal{S}$ an optimal cover of $\{1, \dots, K\}$ with compatible index sets; otherwise, CONTINUE
Let α be the size of an optimal cover of $\{1, \dots, K\}$ by index sets in $\mathcal{S}$
Let β be the size of an optimal cover of $\{1, \dots, K\}$ by index sets in $\mathcal{S} \cup (\mathcal{U} \setminus \mathcal{V})$
if $\alpha \leq \beta$ **then return** BREAK **else return** CONTINUE

Algorithm 4: Top-down algorithm for Problem 2.

Data: Data set $\{(x_k, y_k)\}_{k=1}^K$, template p
Result: An optimal cover of $\{1, \dots, K\}$ with compatible index sets
[...] `// same as in Algorithm 1`
while *true* **do**
  if *Algorithm 3 outputs* BREAK **then return** an optimal cover of $\{1, \dots, K\}$ using index sets from $\mathcal{S}$
  [...] `// same as in Algorithm 1`
end

at the beginning of each iteration, that consists in (i) computing a lower bound β on the size of an optimal cover of $\{1, \dots, K\}$ with compatible index sets; and (ii) checking whether we can extract from $\mathcal{S}$ a collection of β index sets that form a cover of $\{1, \dots, K\}$. The extra step returns BREAK if (ii) is successful. An implementation of the extra step is provided in Algorithm 3.

The soundness of Algorithm 3 follows from the following lemma.

Lemma 10 *Let β be as in Algorithm 3. Then, any cover of $\{1, \dots, K\}$ with compatible index sets has size at least β.*

Proof The crux of the proof relies on the observation from the proof of Theorem 8 that for any compatible index set $I \in \mathcal{I}$, there is $J \in \mathcal{S} \cup (\mathcal{U} \setminus \mathcal{V})$ such that $I \subseteq J$. It follows that for any cover of $\{1, \dots, K\}$ with compatible index sets, there is a cover of $\{1, \dots, K\}$ with index sets in $\mathcal{S} \cup (\mathcal{U} \setminus \mathcal{V})$. Since β is the smallest size of such a cover, this concludes the proof of the lemma. ■

The implementation of the extra step in Algorithm 1 is provided in Algorithm 4. The correctness of the algorithm follows from that of Algorithm 1 (Theorem 8) and Algorithm 3 (Lemma 10).

In conclusion, we provided an algorithm for optimal TPWA regression.

Theorem 11 (Optimal TPWA regression) *Algorithm 4 solves Problem 2 with minimal q.*

Proof Let $I_1, \dots, I_q$ be the output of Algorithm 4. For each i, let $H_i = H(c_i)$ where $I_i = I(c_i)$ and let $f_i(x) = A_i x + b_i$ be as in (b) of Definition 6. The fact that $H_1, \dots, H_q$ and $f_1, \dots, f_q$ is a solution to Problem 2 follows from the fact that $I_1, \dots, I_q$ is a cover of $\{1, \dots, K\}$ and the definition of $f_1, \dots, f_q$. The fact that it is a solution with minimal q follows from the optimality of $I_1, \dots, I_q$ among all covers of $\{1, \dots, K\}$ with compatible index sets. ■

5. Numerical Experiments

5.1. PWA approximation of insulin–glucose regulation model

Dalla Man et al. (2007) present a nonlinear model of insulin–glucose regulation that has been widely used to test artificial pancreas devices for treatment of type-1 diabetes. The model is nonlinear and involves 10 state variables. However, the nonlinearity arises mainly from the term U_{id} (insulin-dependent glucose utilization) involving two state variables, say x_1 and x_2 (namely, the level of insulin in the interstitial fluid, and the glucose mass in rapidly equilibrating tissue):

$$U_{\text{id}}(x_1, x_2) = \frac{(3.2667 + 0.0313x_1)x_2}{253.52 + x_2}.$$

We consider the problem of approximating U_{id} with a PWA model, thus converting the entire model into a PWA model. Therefore, we simulated trajectories of the system and collected $K = 100$ values of x_1, x_2 and $U_{\text{id}}(x_1, x_2)$; see Figure 4*(a)*. For three different values of the error tolerance, $\epsilon \in \{0.2, 0.1, 0.05\}$, we used Algorithm 4 to compute a PWA regression of the data with rectangular domains. The results of the computations are shown in Figure 4*(b,c,d)*. The computation times are respectively 1, 22 and 112 secs[3]. Finally, we evaluate the accuracy of the PWA regression for the modeling of the glucose-insulin evolution by simulating the system with U_{id} replaced by the PWA models. The results are shown in Figure 4*(e,f)*. We see that the PWA model with $\epsilon = 0.05$ induces a prediction error less than $2\,\%$ over the whole simulation interval, which is a significant improvement compared to the PWA models with only 1 affine piece ($\epsilon = 0.2$) or 2 affine pieces ($\epsilon = 0.1$).

Finally, we compare with switched affine regression and classical PWA regression. To find a switched affine model, we solved Problem 3 with $\epsilon = 0.05$ and $q = 3$ using a MILP approach. The computation is very fast (< 0.5 secs); however, the computed clusters of data points (see Figure 6 in Appendix F) do not allow to learn a PWA model, thereby hindering the derivation of a model for U_{id} that can be used for simulation and analysis.

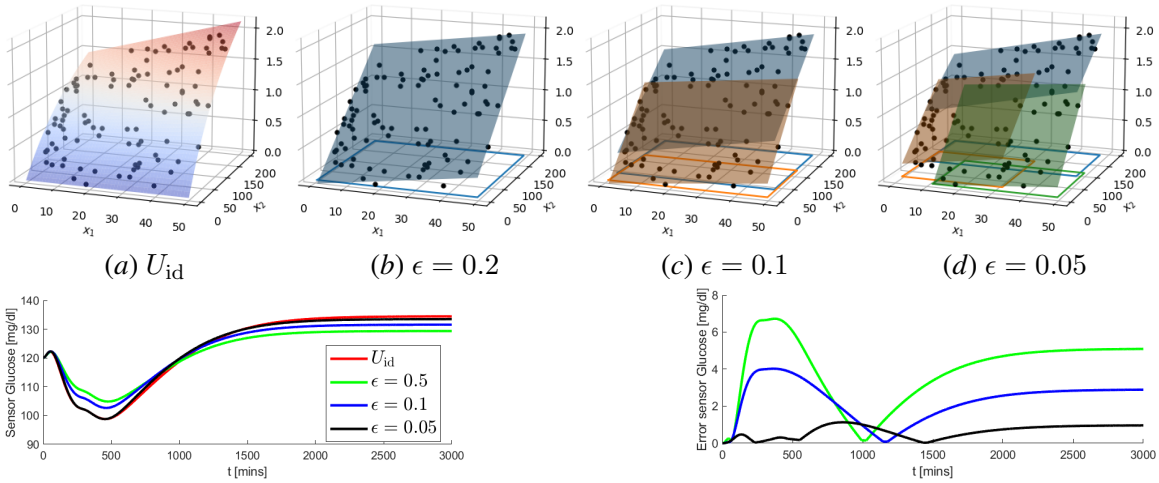
5.2. Hybrid system identification: double pendulum with soft contacts

We consider a hybrid linear system consisting in an inverted double pendulum with soft contacts at the joints, as depicted in Figure 5*(a)*. This system has nine linear modes, depending on whether the contact force of each joint is inactive, active on the left or active on the right (see Aydinoglu et al., 2020). Our goal is to learn these linear modes as well as their domain of validity, from data. For that, we simulated trajectories of the system and collected $K = 250$ sampled values of θ_1, θ_2 and the force applied on the lower joint. We used Algorithm 4 to compute a PWA regression of the data with rectangular domains and with error tolerance $\epsilon = 0.01$. The result is shown in Figure 5*(b)*. The number of iterations of the algorithm was about 23000, for a total time of 800 secs.

We see that the affine pieces roughly divide the state space into a grid of 3×3 regions. This is consistent with our ground truth model, in which the contact force at each joint has three linear modes depending only on the angle made at the joint. The PWA regression provided by Algorithm 4 allows us to learn this feature of the system from data, without assuming anything about the system except that the domains of the affine pieces are rectangular.

Finally, we compare with switched affine (SA) regression and classical PWA regression. The MILP approach to solve the SA regression (Problem 3) with $\epsilon = 0.01$ and $q = 9$ could not handle

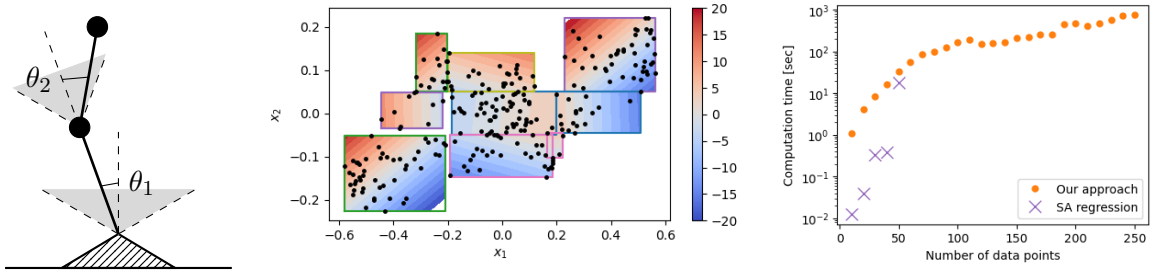
3. On a laptop with Intel Core i7-7600u and 16 GB RAM running Windows, using Gurobi™ as (MI)LP solver.



(*a*) U_{id} (*b*) $\epsilon = 0.2$ (*c*) $\epsilon = 0.1$ (*d*) $\epsilon = 0.05$

(*e*) Simulated trajectories (*f*) Average error over 50 simulations

Figure 4: Glucose–insulin system. *(a)*: 100 sampled points (black dots) on the graph of U_{id} (surface). *(b)*, *(c)*, *(d)*: Optimal TPWA regression for various error tolerances ϵ. *(e)*: Simulation of the glucose mass using the original versus the PWA approximations. *(f)*: Error in the prediction of the glucose mass from the PWA approximations, averaged over 50 simulations with different initial conditions.



(*a*) Schematic (*b*) TPWA regression with $\epsilon = 0.01$ (*c*) Computation times

Figure 5: Inverted double pendulum with soft contacts. *(a)*: Elastic contact forces apply when θ exceeds some limits (i.e., exits the gray region), *(b)*: Optimal TPWA regression of the data with rectangular domains. *(c)*: Comparison with MILP approach for SA regression in terms of computation time for different data sizes. Time limit is set to 1000 secs.

more than 51 data points within reasonable time (1000 secs); see Figure 5*(c)*. Furthermore, the computed clusters of data points (see Figure 7 in Appendix F) do not allow to learn a PWA model, thereby hindering to learn important features of the system.

References

Alp Aydinoglu, Victor M Preciado, and Michael Posa. Contact-aware controller design for complementarity systems. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1525–1531. IEEE, 2020. doi: 10.1109/ICRA40945.2020.9197568.

Alberto Bemporad. A piecewise linear regression and classification algorithm with application to learning and model predictive control of hybrid systems. *IEEE Transactions on Automatic Control*, 2022. doi: 10.1109/TAC.2022.3183036.

Alberto Bemporad, Andrea Garulli, Simone Paoletti, and Antonio Vicino. A bounded-error approach to piecewise affine system identification. *IEEE Transactions on Automatic Control*, 50 (10):1567–1580, 2005. doi: 10.1109/TAC.2005.856667.

Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge University Press, Cambridge, UK, 2004. doi: 10.1017/CBO9780511804441.

Paul S Bradley and Olvi L Mangasarian. K-plane clustering. *Journal of Global optimization*, 16(1): 23–32, 2000. doi: 10.1023/A:1008324625522.

Leo Breiman. Hinging hyperplanes for regression, classification, and function approximation. *IEEE Transactions on Information Theory*, 39(3):999–1013, 1993. doi: 10.1109/18.256506.

Chiara Dalla Man, Robert A Rizza, and Claudio Cobelli. Meal simulation model of the glucose-insulin system. *IEEE Transactions on biomedical engineering*, 54(10):1740–1749, 2007. doi: 10.1109/TBME.2007.893506.

Giancarlo Ferrari-Trecate, Marco Muselli, Diego Liberati, and Manfred Morari. A clustering technique for the identification of piecewise affine systems. *Automatica*, 39(2):205–217, 2005. doi: 10.1016/S0005-1098(02)00224-8.

Sariel Har-Peled. *Geometric approximation algorithms*, volume 173 of *Mathematical Surveys and Monographs*. American Mathematical Society, Providence, RI, 2011.

Wanxin Jin, Alp Aydinoglu, Mathew Halm, and Michael Posa. Learning linear complementarity systems. In *Proceedings of The 4th Annual Learning for Dynamics and Control Conference*, volume 168 of *Proceedings of Machine Learning Research*, pages 1137–1149. PMLR, 2022. https://proceedings.mlr.press/v168/jin22a.html.

Raphaël M Jungers. *The joint spectral radius: theory and applications*. Springer, Berlin, 2009. doi: 10.1007/978-3-540-95980-9.

Fabien Lauer. Estimating the probability of success of a simple algorithm for switched linear regression. *Nonlinear Analysis: Hybrid Systems*, 8:31–47, 2013. doi: 10.1016/j.nahs.2012.10.001.

Fabien Lauer and Gérard Bloch. *Hybrid system identification: theory and algorithms for learning switching models*. Springer, Cham, 2019. doi: 10.1007/978-3-030-00193-3.

Antoine Miné. The octagon abstract domain. *Higher-Order and Symbolic Computation*, 19(1): 31–100, 2006. doi: 10.1007/s10990-006-8609-1.

Eberhard Münz and Volker Krebs. Identification of hybrid systems using a priori knowledge. *IFAC Proceedings Volumes*, 35(1):451–456, 2002. doi: 10.3182/20020721-6-ES-1901.00563.

Eberhard Münz and Volker Krebs. Continuous optimization approaches to the identification of piecewise affine systems. *IFAC Proceedings Volumes*, 38(1):349–354, 2005. doi: 10.3182/20050703-6-CZ-1902.00342.

Hayato Nakada, Kiyotsugu Takaba, and Tohru Katayama. Identification of piecewise affine systems based on statistical clustering technique. *Automatica*, 41(5):905–913, 2005. doi: 10.1016/j.automatica.2004.12.005.

Necmiye Ozay, Constantino Lagoa, and Mario Sznaier. Robust identification of switched affine systems via moments-based convex optimization. In *Proceedings of the 48h IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference*, pages 4686–4691. IEEE, 2009. doi: 10.1109/CDC.2009.5399962.

Necmiye Ozay, Mario Sznaier, Constantino M Lagoa, and Octavia I Camps. A sparsification approach to set membership identification of switched affine systems. *IEEE Transactions on Automatic Control*, 57(3):634–648, 2012. doi: 10.1109/TAC.2011.2166295.

Simone Paoletti, Aleksandar Lj Juloski, Giancarlo Ferrari-Trecate, and René Vidal. Identification of hybrid systems — a tutorial. *European journal of control*, 13(2–3):242–260, 2007. doi: 10.3166/ejc.13.242-260.

Riccardo Porreca, Samuel Drulhe, Hidde de Jong, and Giancarlo Ferrari-Trecate. Identification of parameters and structure of piecewise affine models of genetic networks. *IFAC Proceedings Volumes*, 42(10):587–592, 2009. doi: 10.3182/20090706-3-FR-2004.00097.

R Tyrrell Rockafellar. *Convex analysis*. Princeton University Press, Princeton, NJ, 1970.

Francesco Smarra, Giovanni Domenico Di Girolamo, Vittorio De Iuliis, Achin Jain, Rahul Mangharam, and Alessandro D'Innocenzo. Data-driven switching modeling for mpc using regression trees and random forests. *Nonlinear Analysis: Hybrid Systems*, 36(100882), 2020. doi: 10.1016/j.nahs.2020.100882.

René Vidal, Stefano Soatto, Yi Ma, and Shankar Sastry. An algebraic geometric approach to the identification of a class of linear hybrid systems. In *42nd IEEE International Conference on Decision and Control (IEEE Cat. No. 03CH37475)*, pages 167–172. IEEE, 2003. doi: 10.1109/CDC.2003.1272554.

Appendix A. Proof of NP-Hardness

Given two vectors or matrices u and v, their horizontal (resp. vertical) concatenation is denoted by $[u, v]$ (resp. $[u; v]$). For positive integers d and e and a scalar α, we denote by $[\alpha]_d$ (resp. $[\alpha]_{e,d}$) the vector in $\mathbb{R}^d$ (resp. matrix in $\mathbb{R}^{e\times d}$) whose components are all equal to α.

Proof of Theorem 4: For the simplicity of notation, we will restrict here to piecewise *linear* models (i.e., with $f_i(x) = A_i x$) since PWA models can be obtained from linear ones by augmenting each data point x_k with a component equal to 1, i.e., $x_k \leftarrow [x_k; [1]_1] \in \mathbb{R}^{d+1}$.

We will reduce Problem 3 to Problem 2. Therefore, consider an instance of Problem 3 consisting in a data set $\mathcal{D} = \{(x_k, y_k)\}_{k=1}^K \subseteq \mathbb{R}^d \times \mathbb{R}^e$ and tolerance ϵ. From $\mathcal{D}$, we build another data set $\mathcal{D}' \subseteq \mathbb{R}^{d+K} \times \mathbb{R}^e$ with $|\mathcal{D}'| = 4K$ as follows. For each $1 \leq k \leq K$, we let $\chi_k \in \mathbb{R}^K$ be the indicator vector of the k^{th} component. We define

$$\mathcal{D}' = \{([x_k; \chi_k], y_k)\}_{k=1}^K \cup \{([-x_k; \chi_k], -y_k)\}_{k=1}^K \cup \{([[0]_d; \chi_k], [\epsilon]_e)\}_{k=1}^K \cup \{([[0]_d; \chi_k], [-\epsilon]_e)\}_{k=1}^K.$$

Also, we let p be the hyper-rectangular template in $\mathbb{R}^{d+K}$, which is linear with $\text{size}(p) = 2(d+K)^2$.

Main step: We show that Problem 3 with $\mathcal{D}$, ϵ and $q = 2$ has a solution iff Problem 2 with $\mathcal{D}'$, p, ϵ and $q = 2$ has a solution.

Proof of "if direction". Assume that Problem 2 has a solution given by $H_1, H_2 \subseteq \mathbb{R}^{d+K}$ and $A_1, A_2 \in \mathbb{R}^{e\times(d+K)}$, and for each i, decompose $A_i = [B_i, C_i]$, wherein $B_i \in \mathbb{R}^{e\times d}$ and $C_i \in \mathbb{R}^{e\times K}$. We will show that B_1, B_2 provide a solution to Problem 3.

Therefore, fix $1 \leq k \leq K$. Using the pigeon-hole principle, let $i \in \{1, 2\}$ be such that at least two points in $\{(x_k, \chi_k), (-x_k, \chi_k), ([0]_d, \chi_k)\}$ belong to H_i'. Then, by the convexity of H_i', it holds that $(0, \chi_k) \in H_i$. For definiteness, assume that $(x_k, \chi_k) \in H_i$. Since H_1, H_2 and A_1, A_2 provide a solution to Problem 2, it follows that

$$\|y_k - B_i x_k - C_i \chi_k\|_\infty \leq \epsilon, \quad \|[\epsilon]_e - C_i \chi_k\|_\infty \leq \epsilon, \quad \|[-\epsilon]_e - C_i \chi_k\|_\infty \leq \epsilon.$$

The last two conditions imply that $C_i \chi_k = 0$, so that $\|y_k - B_i x_k\|_\infty \leq \epsilon$. Since k was arbitrary, this shows that B_1, B_2 provide a solution to Problem 3; thereby proving the "if direction".

Proof of "only if direction". Assume that Problem 3 has a solution given by $A_1, A_2 \in \mathbb{R}^{e\times d}$. For each $1 \leq k \leq K$, define the intervals $I_{1,k}, I_{2,k} \subseteq \mathbb{R}$ as follows: $I_{i,k} = [0, 1]$ if $\|y_k - A_i x_k\|_\infty \leq \epsilon$, and $I_{i,k} = \{0\}$ otherwise. Now, define the rectangular regions $H_1, H_2 \subseteq \mathbb{R}^{d+K}$ as follows: $H_i = \mathbb{R}^d \times I_{i,1} \times \ldots \times I_{i,K}$. Also define the matrices $B_1, B_2 \in \mathbb{R}^{e\times(d+K)}$ as follows: $B_i = [A_i, [0]_{e,K}]$. We will show that H_1, H_2 and B_1, B_2 provide a solution to Problem 2.

Therefore, fix $1 \leq k \leq K$ and $i \in \{1, 2\}$. First, assume $\|y_k - A_i x_k\|_\infty \leq \epsilon$. We show that (a) $[x_k; \chi_k]$, $[-x_k; \chi_k]$ and $[[0]_d; \chi_k]$ belong to H_i, and (b)

$$\|y_k - B_i[x_k; \chi_k]\|_\infty \leq \epsilon, \quad \|-y_k - B_i[-x_k; \chi_k]\|_\infty \leq \epsilon, \quad \|[\pm\epsilon]_e - B_i[[0]_d; \chi_k]\|_\infty \leq \epsilon.$$

This is direct (a) by the definition of $I_{i,k}$, and (b) by the definition of B_i. Now, assume that $\|y_k - A_i x_k\|_\infty \leq \epsilon$ does not hold. We show that $[x_k; \chi_k]$, $[-x_k; \chi_k]$ do not belong to H_i. This is direct since $1 \notin I_{i,k}$. Thus, we have shown that H_1, H_2 and B_1, B_2 provide a solution to Problem 2; thereby proving the "only if direction".

Hence, we have built a polynomial reduction from Problem 3 to Problem 2. Since Problem 3 is NP-hard (Lauer and Bloch, 2019, §5.2.4), this shows that Problem 2 is NP-hard as well. ■

Appendix B. Maximal Compatible Index Sets

Lemma 12 *Let q be given. Problem 2 has a solution iff it has a solution wherein the regions correspond to maximal compatible index sets.*

Proof The "if direction" is clear. We prove the "only if direction". Consider a solution of Problem 2 with regions $H_1, \ldots, H_q$. For each $1 \leq i \leq q$, there is a maximal compatible index set $I_i = I(c_i)$ such that $H_i \cap \{x_k\}_{k=1}^K \subseteq H(c_i)$. Since $\{x_k\}_{k=1}^K \subseteq \bigcup_{i=1}^q H_i$, it holds that $\{x_k\}_{k=1}^K \subseteq \bigcup_{i=1}^q H(c_i)$. Hence, $H(c_1), \ldots, H(c_q)$, along with affine functions $f_i(x) = A_i x + b_i$ satisfying (b) in Definition 6, provide a solution to Problem 2, concluding the proof. ■

Appendix C. Correctness of Top-Down Algorithm

Proof of Theorem 8 Termination follows from the fact that each index set $I \in \mathcal{I}$ is picked at most once, because when some $I \in \mathcal{I}$ is picked, it is then added to the collection $\mathcal{V}$ of visited index sets, so that it cannot be picked a second time. Since $\mathcal{I}$ is finite, this implies that the algorithm terminates in a finite number of steps.

Now, we prove that, upon termination, any maximal compatible index set is in the output $\mathcal{S}$ of the algorithm. Therefore, let J be a maximal compatible index set. Then, among all sets I picked during the execution of the algorithm and satisfying $J \subseteq I$, let I^* have minimal cardinality. Such an index set exists since $J \subseteq \{1, \ldots, K\}$. We will show that:

Main result. $I^* = J$.

Proof of main result. For a proof by contradiction, assume that $I^* \neq J$. Since J is maximal and $J \subsetneq I^*$, I^* is not compatible. Hence, the index sets $(I_1, \ldots, I_S) = \text{FINDSUBSETS}(I^*)$ were added to $\mathcal{U}$. Using the assumption on FINDSUBSETS, let s be such that $J \subseteq I_s \subsetneq I^*$. Since I_s must have been picked during the execution of the algorithm, this contradicts the minimality of the cardinality of I^*, concluding the proof of the main result.

Thus, J was picked during the execution of the algorithm. Since it is compatible, it was added to $\mathcal{S}$ at the iteration at which it was picked, and since it is maximal, it is not removed at later iterations. Hence, upon termination, $J \in \mathcal{S}$. Since J was arbitrary, this concludes the proof that, upon termination, $\mathcal{S}$ contains all maximal compatible index sets.

Finally, we show that, upon termination, $\mathcal{S}$ contains only maximal compatible index sets. This follows from the fact that, at each iteration of the algorithm, for any distinct $I_1, I_2 \in \mathcal{S}$, it holds that $I_1 \nsubseteq I_2$ and $I_2 \nsubseteq I_1$. Indeed, when I_1 is added to $\mathcal{S}$, all subsets of I_1 are removed from $\mathcal{S}$ and are added to $\mathcal{V}$ so that they are not picked at later iterations. The same holds for I_2. This concludes the proof of the theorem. ■

Appendix D. Correctness of FINDSUBSETS

Lemma 13 *If C is an infeasibility certificate, then every $I \subseteq \{1, \ldots, K\}$ satisfying $C \subseteq I$ is not compatible.*

Proof Straightforward from (b) in Definition 6. ■

Theorem 14 (Correctness of Algorithm 2) *For every non-compatible index set $I \in \mathcal{I}$, the output $I_1, \ldots, I_S$ of Algorithm 2 is consistent w.r.t. I.*

Proof Let $J \subseteq I$ be compatible. Using that $C \nsubseteq J$ (Lemma 13), let s be a component such that $\max_{k \in J} p^s(x_k) < \max_{k \in C} p^s(x_k)$. It holds that $J \subseteq I_s$. Since J was arbitrary, this concludes the proof. ■

Appendix E. Spatially Concentrated Infeasibility Certificates

Given a center point $\bar{x}$ and a non-compatible index set $I \subseteq \{1, \ldots, K\}$, we consider the following Linear Program: with variables $\lambda_k \in \mathbb{R}$, $\forall\, k \in I$,

$$\begin{array}{rl} \text{minimize} & \sum_{k \in I} |\lambda_k| \|x_k - \bar{x}\|^2 \\ \text{s.t.} & \sum_{k \in I} \lambda_k [x_k; [1]_1] = [0]_{d+1} \ \wedge \ \sum_{k \in I} \lambda_k y_k \not\geq -\sum_{k \in I} |\lambda_k| \epsilon. \end{array} \tag{3}$$

From the theorem of alternatives of Linear Programming, it holds that (3) has a feasible solution $\{\lambda_k\}_{k \in I}$ satisfying that at most $d + 2$ variables are nonzero. The objective function of (3) tends to put zero value to λ_k whenever $\|x_k - \bar{x}\|_\infty$ is large. This promotes proximity of the point x_k to $\bar{x}$ when $\lambda_k \neq 0$.[4] In our experiments, we used $\bar{x} = \frac{1}{|I|} \sum_{k \in I} x_k$.

Appendix F. Supplementary material

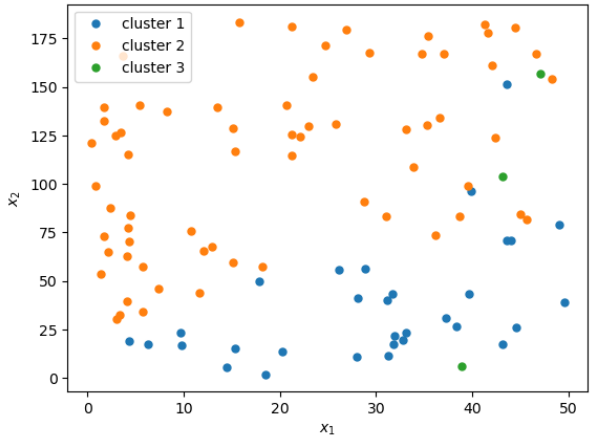


Figure 6: Clusters of data points from SA regression of the data set from the insulin–glucose regulation system in Subsection 5.1.

4. Note that L^1 regularization costs are often used in machine learning to induce sparsity of the optimal solution (Boyd and Vandenberghe, 2004, p. 304). Here, we use a *weighted* L^1 regularization cost to induce a sparsity pattern dictated by the geometry of the problem.

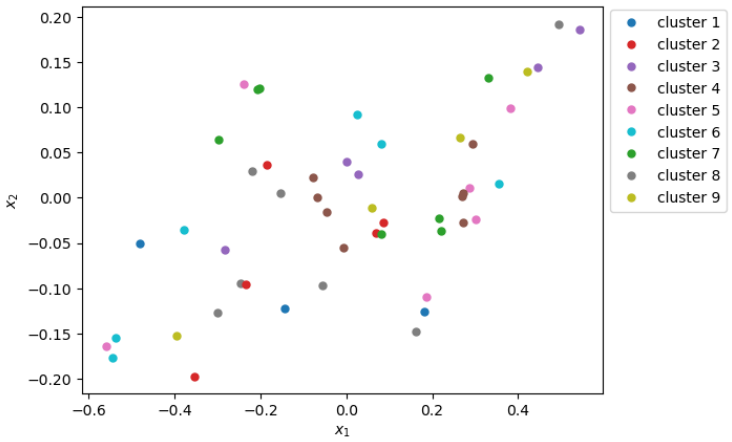


Figure 7: Clusters of data points from SA regression of 51 data points from the inverted double pendulum with soft contacts in Subsection 5.2.