Contents lists available at ScienceDirect

ISA Transactions

journal homepage: www.elsevier.com/locate/isatrans

Research article

Prediction regions based on dissimilarity functions

A.D. Carnerero [a,*], D.R. Ramirez [a], S. Lucia [b], T. Alamo [a]

[a] Departamento de Ingenieria de Sistemas y Automatica, Universidad de Sevilla, Spain
[b] Laboratory of Process Automation Systems, TU Dortmund University, Germany

ARTICLE INFO

Article history:
Received 15 July 2022
Received in revised form 13 March 2023
Accepted 31 March 2023
Available online 5 April 2023

Keywords:
Prediction regions
System identification
Nonlinear systems
Uncertainty

ABSTRACT

This paper presents a new methodology to obtain prediction regions of the output of a dynamical system. The proposed approach uses stored past outputs of the system and it is entirely data-based. Only two hyperparameters are necessary to apply the proposed methodology. These scalars are chosen so that the size of the obtained regions is minimized while fulfilling the desired empirical probability in a validation set. In this paper, methods to optimally estimate both hyperparameters are provided. The provided prediction regions are convex and checking if a given point belongs to a computed prediction region amounts to solving a convex optimization problem. Also, approximation methods to build ellipsoidal prediction regions are provided. These approximations are useful when explicit descriptions of the regions are necessary. Finally, some numerical examples and comparisons for the case of a non-linear uncertain kite system are provided to prove the effectiveness of the proposed methodology.

© 2023 The Author(s). Published by Elsevier Ltd on behalf of ISA. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

1. Introduction

Dynamical models, either linear or nonlinear, suffer from a certain degree of modeling errors and uncertainty that results in inaccurate predictions of the outputs. One way to consider these uncertainties is by using prediction regions in which the future outputs are guaranteed to be contained with a pre-specified probability [1,2]. There are many applications where this kind of strategy is useful. For example, in stochastic programming and chance constrained problems [3,4], some of the parameters are random variables and, thus, it is necessary to characterize such uncertainty. Also, stochastic model predictive control (MPC) and chance constrained MPC [5] require some sort of uncertainty or disturbance characterization within the model of the system in order to solve the control problem.

For the case of single-output systems, interval predictions can be used. This is a particular and easier case of the aforementioned prediction regions. These intervals can be constructed in two different ways, one consisting on following a robust scheme, that is, the prediction will be included in the interval with a probability equal to one. For this uni-variate setting, there are state-of-the-art techniques that work reasonably well. For example, if the uncertainty is bounded, set membership methods can be used to obtain robust intervals [6,7]. However, robust approaches can be overly conservative and thus a second approach has gained the attention of researchers. This approach is the use of stochastic methods. In a stochastic setting, the prediction belongs to the aforementioned interval with a certain probability $1-\tau$. In this context, one could rely on parametric regression techniques, obtaining a quantile regressor (QR) [8,9]. The main limitation of these methods is that a large number of training samples is necessary when τ is close to the extremes of $(0, 1)$. Similarly, as an extension of QR techniques, one could rely on Interval Predictor Model (IPM) approaches [10].

The aforementioned methods can deal with prediction intervals, i.e., single output systems. However, computing prediction regions for multi-variate systems becomes a harder task. The simplest way would be to obtain intervals considering each variable independently and then construct rectangular prediction regions (see Section 2.2.3 in [11]). The main advantage of this method lies in its simplicity. However, the desired probability may not be attained or the size of the regions may be too large. Bootstrap methods are used in [12,13] to construct regions that contain a path of a random variable with at least a certain probability. This means that the obtained regions are actually intervals for a $p-$step prediction. In the case of multi-variable dynamical systems, the literature is rather scarce. For example, in [14], prediction regions for a simple multivariate linear regression model are obtained. On the other hand, [15] proposes a method to calculate prediction regions of a system by computing the Jacobian of the Partial Least-Squares Regression (PLS) parameters. Thus, by means of this local linearization, an ellipsoid-shaped region can be obtained. Also, [16] manages to obtain ellipsoid regions for dynamical systems by means of an Inverse Regression

[*] Correspondence to: Avenida Camino de los descubrimientos s/n, 41092, Sevilla, Spain.
E-mail address: acarnerero@us.es (A.D. Carnerero).

https://doi.org/10.1016/j.isatra.2023.03.048
0019-0578/© 2023 The Author(s). Published by Elsevier Ltd on behalf of ISA. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

(IR) scheme. This IR scheme is more efficient and reliable than the classic regression approaches, when high dimensional data is available. In [17], a data-driven framework to generate and evaluate ellipsoidal prediction regions to characterize the uncertainty of a time-series is proposed. This methodology is applied to the electricity prices as a path forecasting problem. In the field of machine learning, Conformal Prediction techniques [18] are used to obtain prediction regions for any method producing a central prediction of the outputs. Similarly, it is proposed in [19] a framework to obtain a stochastic model based on a deterministic model of a robotic dynamic system. On the other hand, assuming that any finite set of samples follows a multivariate normal distribution, Gaussian processes (GPs) [20] can also be used. The main limitation of this approach is that it relies heavily on the knowledge of the first two moments of the underlying multivariate probability distribution.

In this paper, we propose a new methodology to compute prediction regions of a multivariate dynamical system by means of dissimilarity functions. These functions have been used in [21,22] to obtain uni-variate interval predictors. In those papers, an empirical probability distribution is obtained and then the intervals are computed from the quantiles of such empirical distribution. Here, the dissimilarity functions are used in a different setting to obtain prediction regions for the multidimensional case. The proposed region is defined implicitly through a convex optimization problem, skipping the computation of the empirical distribution, that would be intractable for the multivariate case. Thus, the procedure is much lighter in terms of computational burden than the one presented in [21]. Also, the hyperparameters in which the methodology relies are computed to minimize the size of the regions while achieving the desired empirical probability. Moreover, the proposed approach does not make any assumption on the data or the shape of the region, being able to represent more tightly different types of regions. Also, in order to tackle the case where explicit regions are necessary, a method to obtain ellipsoid approximations of the regions is provided.

The rest of the paper is organized as follows. Section 2 presents the dissimilarity function that will be used throughout the manuscript whereas its role in forecasting the output of a nonlinear system is presented in Section 3. Section 4 defines the predictions regions and proposes a method to tune the hyperparameters. Also, in Section 5 an approximation method based on a quadratic upper bound of the optimization problem is proposed. By means of this approximation, an ellipsoidal region can be obtained. In Section 6, two numerical examples of nonlinear dynamical systems are proposed in order to compare the performance of the region predictor with other baselines. Finally, Section 7 presents the conclusions.

2. Dissimilarity functions

Dissimilarity functions [21] are used to measure the similarity of an element to the elements of a given set. They will play a key role in the remaining sections of the paper, thus a brief summary of the key concepts is given in the following. Consider a set $\mathcal{D} = \begin{bmatrix} d_1 & d_2 & \dots & d_N \end{bmatrix} \in \mathbb{R}^{n\times N}$, where $d_i \in \mathbb{R}^n$. We are interested in determining how similar a given vector $d \in \mathbb{R}^n$ is with respect to the other vectors of the set $\mathcal{D}$. Thus, a dissimilarity function $J(\cdot) : \mathbb{R}^n \times \mathcal{D} \to [0,\infty]$, measures the dissimilarity between a given point d and the data set $\mathcal{D}$. Large values of $J(d,\mathcal{D})$ imply a high dissimilarity whereas small values correspond to a high similarity. A first candidate could be the minimum distance of the point d to each member in the data set $\mathcal{D}$. That is

$$J(d,\mathcal{D}) = \min_{i=1,\dots,N} \|d - d_i\|, \tag{1}$$

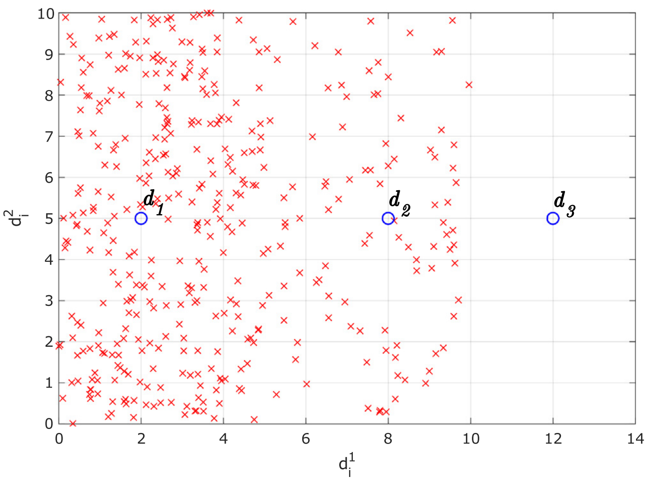


Fig. 1. Elements of set $\mathcal{D} \subset \mathbb{R}^2$ (in red) and three points in $\mathbb{R}^2$ (in blue). Note that d_i^1 and d_i^2 denote the first and second component of each d_i respectively.

where $\|\cdot\|$ is a given norm. Another possibility would be to consider the mean value of the distances of d to each member of $\mathcal{D}$. However, these dissimilarity functions, although simple, are not invariant with respect to affine transformations and thus they depend on the choice of the coordinate system. A better dissimilarity function would be

$$J_\gamma(d,\mathcal{D}) = \min_{\lambda_1,\dots,\lambda_N} (1-\gamma)\sum_{i=1}^{N} w_i\lambda_i^2 + \gamma\sum_{i=1}^{N}|\lambda_i| \tag{2a}$$

$$\text{s.t.}\quad d = \sum_{i=1}^{N}\lambda_i d_i \tag{2b}$$

$$1 = \sum_{i=1}^{N}\lambda_i, \tag{2c}$$

where $\gamma \in [0,1)$ is a tuning parameter and $w_i > 0$ constant weights. This dissimilarity function returns a small value when the vector is very similar to the elements of $\mathcal{D}$ and it returns a large value when the vector is very dissimilar. To illustrate this concept, consider Fig. 1 in which a set $\mathcal{D}$ (red markers) and three element examples (blue markers) denoted as d_1, d_2 and d_3 are plotted. The values of the dissimilarity function with $\gamma = 0$ and $w_i = 1,\ \forall i$ for each regressor are:

$$J_0(d_1,\mathcal{D}) = 0.0021,\quad J_0(d_2,\mathcal{D}) = 0.0085,\quad J_0(d_3,\mathcal{D}) = 0.0226.$$

Thus, d_1 attains the lowest dissimilarity, i.e. it is more similar with respect to the set because the concentration of points around d_1 is larger than around d_2 and d_3. On the other hand, d_3 attains the highest dissimilarity due to the fact that d_3 is actually outside the region in which all the elements of $\mathcal{D}$ lie.

This optimization problem is strictly convex and thus it has a unique solution. In order to guarantee that this solution exists, we assume that $\begin{bmatrix} d_1 & \dots & d_N \\ 1 & \dots & 1 \end{bmatrix}$ is full row rank. To justify that problem (2) is strictly convex, consider that it can be rewritten as

$$J_\gamma(d,\mathcal{D}) = \min_{\lambda} \quad (1-\gamma)\lambda^\top H\lambda + \gamma\|\lambda\|_1 \tag{3a}$$

$$\text{s.t.}\quad \begin{bmatrix} \mathcal{D} \\ \mathbf{1} \end{bmatrix}\lambda = \begin{bmatrix} d \\ 1 \end{bmatrix}, \tag{3b}$$

where $\|\lambda\|_1 = \sum_{i=1}^{N}|\lambda_i|$ and $H > 0$ because the weighting factors w_i are strictly positive. It is easy to see that the quadratic term

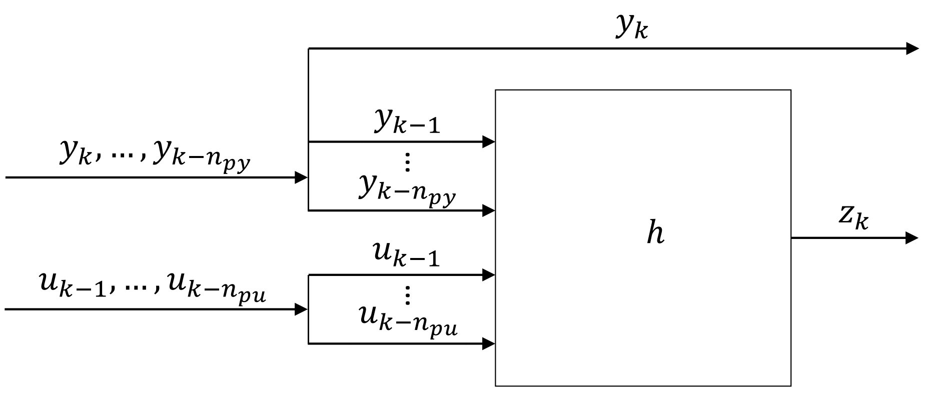


Fig. 2. Relation between past inputs, outputs, the operator $h(\cdot)$ and the regressor z_k.

$(1-\gamma)\lambda^\top H\lambda$ is strictly convex because $H > 0$ and $(1-\gamma) > 0$. Also, the 1-norm term $\|\lambda\|_1$ is convex as well because $\gamma \in [0, 1)$. Thus, the sum of the quadratic term and the absolute value term is a strictly convex function. Therefore, the optimization problem is strictly convex.

Remark 1. This dissimilarity function is invariant to affine transformations, being more convenient than the ones based only on the distance of the point d with respect to the data set $\mathcal{D}$ because they are dependent with respect to the reference system (see [21,23]).

3. Forecasting nonlinear systems using dissimilarity functions

Consider a discrete time nonlinear system

$$\begin{aligned} x_{k+1} &= f(x_k, u_k) \\ y_k &= g(x_k, v_k), \end{aligned} \tag{4}$$

where k is the time instant, $x_k \in \mathcal{R}^{n_x}$ are the states, $u_k \in \mathcal{R}^{n_u}$ are the inputs, $y_k \in \mathcal{R}^{n_y}$ are the outputs, $v_k \in \mathcal{R}^{n_v}$ accounts for measurement noise, $f(\cdot)$ and $g(\cdot)$ are unknown nonlinear functions such that $f(\cdot) : \mathcal{R}^{n_x \times n_u} \to \mathcal{R}^{n_x}$ and $g(\cdot) : \mathcal{R}^{n_x \times n_v} \to \mathcal{R}^{n_y}$. We assume that only past outputs and inputs are accessible.

In this section, we aim to obtain the prediction of the output value denoted as $\hat{y}_k$. This prediction will be computed as a function of a regressor denoted as z_k and a given data set of past regressors and their corresponding output values. As we assumed that only past outputs and inputs are measurable, the regressor z_k is obtained by applying a certain operator $h(\cdot)$ to past outputs and inputs, with $h(\cdot) : \mathcal{R}^{n_u n_{pu} + n_y n_{py}} \to \mathcal{R}^{n_z}$ where n_{pu} and n_{py} are the number of past terms of the input and the output respectively, and n_z is the dimension of the regressor (see Fig. 2). It is also assumed that this operator is known. The operator, included for generality, could be the identity, which would provide the raw past values of the inputs and the outputs without doing any mathematical operation, that is, $z_k = [u_{k-1}{}^\top, \dots, u_{k-n_{pu}}{}^\top, y_{k-1}{}^\top, \dots, y_{k-n_{py}}{}^\top]^\top$. However, more complex operators can be used, such as kernel or neural networks based operators (like in the numerical example of Section 6).

Denoting by $\bar{z}_i \in \mathcal{R}^{n_z}$ and $\bar{y}_i \in \mathcal{R}^{n_y}$ the past samples of the regressor and the output respectively, we assume that the data sets are stored in the form of the matrices

$$\mathcal{Z} = \begin{bmatrix} \bar{z}_1 & \bar{z}_2 & \dots & \bar{z}_N \end{bmatrix}, \; \mathcal{Y} = \begin{bmatrix} \bar{y}_1 & \bar{y}_2 & \dots & \bar{y}_N \end{bmatrix},$$

where N is the number of samples. These data sets will be used to obtain the prediction of the system output for a given regressor z. Here, it is proposed to compute such prediction using a weighted sum of the elements of $\mathcal{Y}$, that is,

$$\hat{y}(z) = \sum_{i=1}^{N} \lambda_i^* \bar{y}_i. \tag{5}$$

We will discuss in the following how to compute the weights λ_i^*. To obtain the desired prediction of the output using the proposed structure, we consider the dissimilarity function given in the previous section written for the forecasting problem as

$$J_\gamma\left(\begin{bmatrix} z \\ y \end{bmatrix}, \begin{bmatrix} \mathcal{Z} \\ \mathcal{Y} \end{bmatrix}\right) = \min_{\lambda_1,\dots,\lambda_N} (1-\gamma)\sum_{i=1}^{N} w_i\lambda_i^2 + \gamma\sum_{i=1}^{N}|\lambda_i| \tag{6a}$$

$$\text{s.t. } z = \sum_{i=1}^{N} \lambda_i \bar{z}_i \tag{6b}$$

$$y = \sum_{i=1}^{N} \lambda_i \bar{y}_i \tag{6c}$$

$$1 = \sum_{i=1}^{N} \lambda_i. \tag{6d}$$

We will compute $\hat{y}(z)$, the expected value for the output y given the value of the regressor z, as the one that gives the minimum value of the dissimilarity function, that is,

$$\hat{y}(z) = \arg\min_{y} J_\gamma\left(\begin{bmatrix} z \\ y \end{bmatrix}, \begin{bmatrix} \mathcal{Z} \\ \mathcal{Y} \end{bmatrix}\right). \tag{7}$$

The following lemma states how to find $\hat{y}(z)$ by solving a simpler optimization problem which in turn will return the optimal values of the weights necessary to compute $\hat{y}(z)$ using the structure of (5).

Lemma 1 (*Computation of the Prediction*)**.** *The prediction $\hat{y}(z)$ can be obtained as:*

$$\hat{y}(z) = \mathcal{Y}\lambda^* = \sum_{i=1}^{N} \lambda_i^* \bar{y}_i, \tag{8}$$

where λ^* is computed as

$$\lambda^* = \arg\min_{\lambda_1,\dots,\lambda_N} (1-\gamma)\sum_{i=1}^{N} w_i\lambda_i^2 + \gamma\sum_{i=1}^{N}|\lambda_i| \tag{9a}$$

$$\text{s.t. } z = \sum_{i=1}^{N} \lambda_i \bar{z}_i \tag{9b}$$

$$1 = \sum_{i=1}^{N} \lambda_i. \tag{9c}$$

Proof. Let us consider $J_\gamma(z, \mathcal{Z})$, that is,

$$J_\gamma(z, \mathcal{Z}) = \min_{\lambda_1,\ldots,\lambda_N} (1-\gamma)\sum_{i=1}^{N} w_i\lambda_i^2 + \gamma \sum_{i=1}^{N} |\lambda_i| \tag{10a}$$

$$\text{s.t. } z = \sum_{i=1}^{N} \lambda_i \bar{z}_i \tag{10b}$$

$$1 = \sum_{i=1}^{N} \lambda_i. \tag{10c}$$

As this optimization problem is a less constrained version of that in (6), the optimal value of the cost of the more constrained problem is always greater or equal to the optimal value of the cost of the less constrained one, that is,

$$J_\gamma (z, \mathcal{Z}) \leq J_\gamma \left(\begin{bmatrix} z \\ y \end{bmatrix}, \begin{bmatrix} \mathcal{Z} \\ \mathcal{Y} \end{bmatrix} \right). \tag{11}$$

Now, denote the vector of weights λ that minimizes the optimization problem in (10) as λ^*. Then, it is easy to see that

$$J_\gamma (z, \mathcal{Z}) = J_\gamma \left(\begin{bmatrix} z \\ \mathcal{Y}\lambda^* \end{bmatrix}, \begin{bmatrix} \mathcal{Z} \\ \mathcal{Y} \end{bmatrix} \right).$$

Because of this and (11), it is not possible to find a value of y improving the value of the dissimilarity function. Thus, $\hat{y}(z) = \mathcal{Y}\lambda^* = \sum_{i=1}^{N} \lambda_i^* \bar{y}_i$, computed using the optimal weights from (10), is the one that minimizes the dissimilarity function for a fixed z in Eq. (7). ∎

An important property is the uniqueness of the prediction. The following remark states that $\hat{y}(z)$ is unique.

Remark 2 (*Uniqueness Of $\hat{y}(z)$*)**.** From the results stated in the previous section problem (6) is strictly convex and because $\begin{bmatrix} \mathcal{Z}^T & \mathcal{Y}^T & \mathbf{1}^T \end{bmatrix}^T$ is assumed to be full rank, its solution exists. Therefore, for a certain z, $\mathcal{Z}$ and $\mathcal{Y}$, the dissimilarity function (6) has a unique minimum at $\hat{y}(z) = \mathcal{Y}\lambda^*$, where λ^* corresponds to the optimal solution of problem (10), and therefore the prediction $\hat{y}(z)$ is unique.

The numerical computation of problem (9) can be addressed by means of a dual formulation. For this particular optimization problem, the number of dual decision variables corresponds to the number of equality constraints, i.e. (n_z+1) which is obviously much smaller than the number of data points (N). Also, note that once the dual variables are fixed, the primal variables can be obtained by solving N unidimensional problems which have explicit solutions. In this paper, the numerical examples have been computed by using an accelerated gradient method in the dual variables [24,25].

For the univariate case ($n_y = 1$) and for the particular choice $\gamma = 0$, it was proved in [21] that this forecasting corresponds to the one obtained by means of least-squares regression. As we explain in what follows, strictly positive values of γ encourage the components of λ to be positive (which means that the central estimation is often obtained from an interpolation of points). When every $\lambda_i \geq 0$, $\forall i = 1,\ldots,N$, $\|\lambda\|_1 = 1$. However, when some of the components of λ are negative, we infer that $\|\lambda\|_1 > 1$. In other words, the cost term $\|\lambda\|_1$ becomes larger when extrapolating points (i.e. using negative values of λ_i). This means that convex combinations of λ_i are encouraged and thus interpolation is preferred, which may improve the predictions. The choice of γ will be further discussed in the context of the proposed methodology in Section 4.4.

Remark 3. The weighting factors w_i offer a way to add flexibility to the forecasting scheme. They could be used to consider locality in the data, by making $w_i = \max\{\|z - z_i\|_2^2, \epsilon\}$, where ϵ is a small positive constant. In this way, local estimation can be implemented (see [21]).

4. Region estimation

Besides obtaining a forecasting for the next output, the aforementioned dissimilarity function allows us to compute regions that will contain the output with a pre-specified probability. This is an extension of the method proposed in [21] where only one-dimensional outputs were considered and, thus, interval predictions were obtained. In the present work, we address the multidimensional case, providing prediction regions for the multidimensional output. The scheme proposed in [21] is based on a numerical integration in a unidimensional space. The generalization of [21] to the multidimensional setting is not trivial because of the well-known complexity of high-dimensional numerical integration. Therefore, that technique is not useful in practice for multivariable systems. In what follows, we show how to obtain prediction regions in multidimensional spaces without resorting to numerical integration, providing a fundamental advantage with respect to the results presented in [21]. In order to characterize such a region, two issues must be tackled: the choice of the region center and the computation of the region itself.

4.1. Choosing the center

In the previous section, we denoted $\hat{y}(z)$ as the value of y that minimizes the dissimilarity function given z (7). This optimal value can be obtained as $\hat{y}(z) = \mathcal{Y}\lambda^*$ where λ^* corresponds to the optimal solution of the strictly convex problem (10).

As the prediction $\hat{y}(z)$ is unique (see Remark 2), it will be used to define the center of the region that will contain the real value of the predicted output with a given probability. From now on, in order to make the manuscript more readable, we simplify the notation by removing $\mathcal{Z}$ and $\mathcal{Y}$ from the dissimilarity function, since they are assumed to be fixed. That is, we use the notation

$$J_\gamma \left(\begin{bmatrix} z \\ y \end{bmatrix}, \begin{bmatrix} \mathcal{Z} \\ \mathcal{Y} \end{bmatrix} \right) = J_\gamma (z, y).$$

Moreover, the minimal value of the dissimilarity function given z is denoted as $J_\gamma^*(z)$. That is,

$$J_\gamma^*(z) = J_\gamma(z, \hat{y}(z)) = \min_y J_\gamma(z, y).$$

4.2. Computing the regions

The dissimilarity function can be seen as a sort of surrogate of the probability distribution of y given z [21]. Thus, the prediction region will be defined in relation to the value of $J_\gamma^*(z)$. We consider that this probability distribution has a maximum at $\hat{y}(z)$ and decreases as the dissimilarity function increases. For that reason, we consider prediction regions that are defined as those points for which the dissimilarity function does not exceed more than a given factor α the one corresponding to the central prediction $\hat{y}(z)$. This is formally stated in the following definition.

Definition 1 (*Prediction Region*)**.** For a given z, γ, data sets $\mathcal{Z}$, $\mathcal{Y}$ and a tunable parameter $\alpha > 1$, we define a prediction region as the set

$$\Delta(z) = \left\{ y : J_\gamma(z, y) \leq \alpha J_\gamma^*(z) \right\}, \tag{12}$$

that is, the points y that obtain a dissimilarity less or equal to $\alpha J_\gamma^*(z)$.

Table 1
Numerical results of the clarifying example.

	Area	Empirical probability
$\gamma = 0$	1.1423	0.9600
$\gamma = 0.2$	1.0077	0.9580
$\gamma = 0.4$	0.9834	0.9640
$\gamma = 0.8$	0.9561	0.9520

We note that the provided regions are defined in an implicit way (we address explicit ellipsoidal regions in the next section). Since the dissimilarity function is convex in y, the obtained implicit regions are convex and can be used in different settings, e.g. in chance-constrained optimization. For example, in many cases, it is not necessary to compute a prediction region, but to verify if a certain point $\bar{y}$ belongs to it. This is an affordable task as it suffices to compute the dissimilarity $J_\gamma(z, \bar{y})$ and check that (12) holds.

4.3. Clarifying example: multivariate uniform distribution

To illustrate the role of γ, consider the easier case of a multivariate uniform distribution. As we are not considering a dynamic system, conditioned distributions are not taken into account. A number of samples $N = 500$ are extracted from a uniform distribution in $\mathcal{R}^2$ and gathered into a data set $\mathcal{Y}$. This uniform distribution is generated considering that both dimensions are independent one with respect the other, with values ranging from 0 to 1. Then, the obtained distribution is rotated with an angle of $\frac{\pi}{6}$ rad in order to misalign the previously computed distribution. We will approximate the support of the generated distribution by means of prediction regions of the form

$$\Delta = \left\{ y : J_\gamma\left(y, \mathcal{Y}\right) \le \alpha J_\gamma^* \right\} .$$

Here, we consider a finite family of four possible values of γ, denoted as $\Gamma = \{0, 0.2, 0.4, 0.8\}$. As we stated in the previous sections, $\gamma = 0$ corresponds to the least-squares solution [21] and thus an ellipsoid will be obtained. However, more appropriate shapes can be obtained with different values of γ. Also, note that the empirical expectation turns out to be the center of the region. In this example, we consider $\tau = 0.05$, that is, we would like the probability of the samples falling within the region to be 95%. The results are shown in Fig. 3. The blue crosses correspond to the points falling inside the region whereas the red crosses correspond to the points falling outside. It is easy to see that $\gamma = 0.8$ is the element of Γ that captures best the shape of the proposed probability distribution, leading to a tighter region.

Also, it can be seen in Table 1 that, for this example, increasing the value of γ provides a smaller area (tighter approximation) while still fulfilling the desired empirical probability. Even though the difference might seem small at first, we note that this difference may increase when considering regions of greater dimensions.

4.4. Tuning the hyperparameters α and γ

Regarding which value of γ should be used, consider a validation set

$$\mathcal{V} = \{(\tilde{z}_1, \tilde{y}_1), (\tilde{z}_2, \tilde{y}_2), \dots, (\tilde{z}_{N_v}, \tilde{y}_{N_v})\},$$

composed of pairs $(\tilde{z}_j, \tilde{y}_j)$ gathered from the system. The problem of determining the best value of $\gamma \in [0, 1)$ can be addressed in different ways. Suppose that, in order to reduce the complexity of the problem, γ is constrained to belong to a set $\Gamma \subseteq [0, 1)$ of finite cardinality (e.g. a set of equidistant points in the interval $[0, 1)$), i.e., a finite family of γ. Then, a reasonable choice would be the value that minimizes the error of the central predictions $\{\hat{y}(\tilde{z})\}$ with respect to the real outputs $\{\tilde{y}\}$ in the validation set $\mathcal{V}$, that is:

$$\gamma^* = \arg\min_{\gamma\in\Gamma} \frac{1}{N_v} \sum_{j=1}^{N_v} \left\| \hat{y}(\tilde{z}_j) - \tilde{y}_j \right\|_2^2 .$$

Note that minimizing the errors of the predictions is related to minimizing the size of the regions. This is due to the fact that reducing the gap between $\hat{y}(\tilde{z}_j)$ and $\tilde{y}_j$ leads to a smaller value of α needed to fulfill the desired probability $1 - \tau$.

Given γ, smaller values of α make the regions smaller. This can be easily seen by looking at Definition 1, where the equation of the prediction region is presented. There, we note that α appears multiplying the optimum value $J_\gamma^*(z)$, i.e. the optimum value of the dissimilarity function given z. Therefore, α is a parameter that implicitly defines the size of the region. That is, for the case when $\alpha = 1$, we have that the region is composed of just the point $\hat{y}(\tilde{z}_j)$ whereas for larger values of α, the number of points included in the region increases. The procedure to obtain α presented in the following aims to obtain the smallest possible region that guarantees that a point $\tilde{y}_j$ taken from the validation set $\mathcal{V}$ is contained in the computed region $\Delta(\tilde{z}_j)$ with a pre-specified probability $1-\tau$. In order to circumvent the difficulty of generating i.i.d. samples $\{(z, y)\}$, we consider that the discrete probability described in the validation set $\mathcal{V}$ is assumed to represent the true probability distribution. Then, α is chosen so that it fulfills

$$\text{Prob}_{\mathcal{V}}(y \in \Delta(z)) \ge 1 - \tau,$$

that is, guaranteeing that the fraction of points y in the validation set that fall out of the corresponding prediction region does not exceed τ. Denote

$$\tilde{\alpha}_j = \frac{J_\gamma(\tilde{z}_j, \tilde{y}_j)}{J_\gamma^*(\tilde{z}_j)},\ j = 1, \dots, N_v.$$

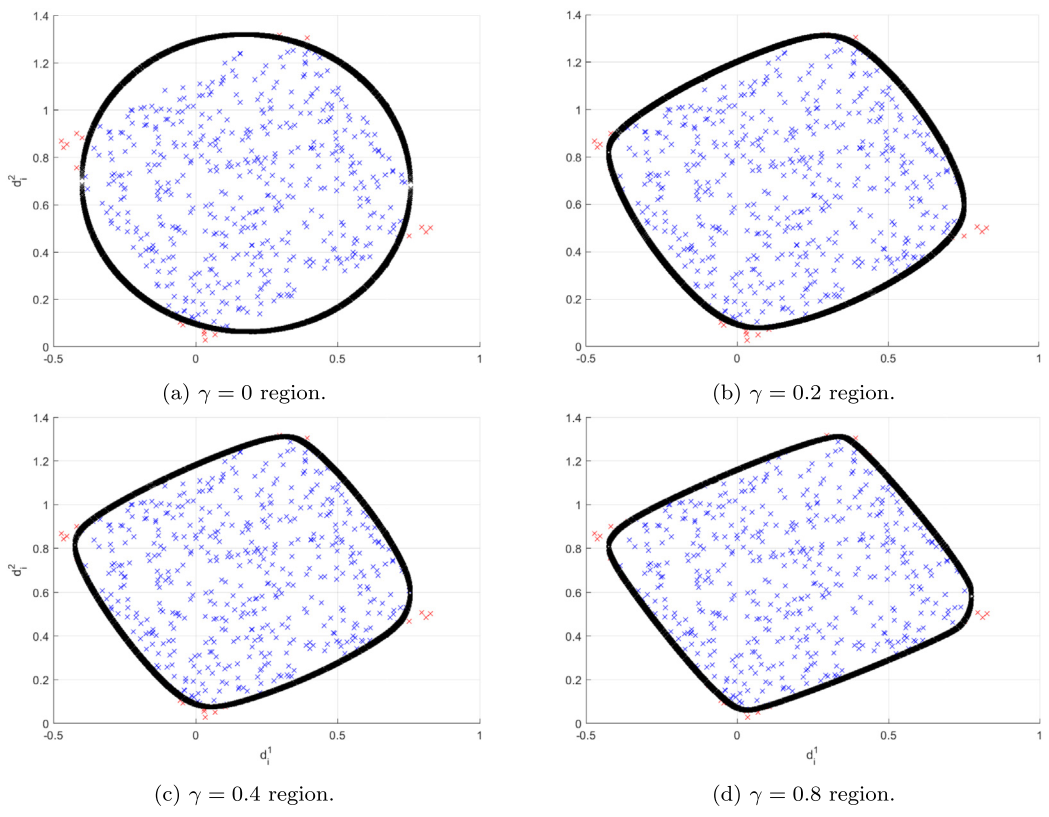
Then, the value of α satisfying the probabilistic specification in the set $\mathcal{V}$ is the one corresponding to the $r_\tau = \lceil \tau N_v \rceil$ largest value of $\{\tilde{\alpha}_j\}_{j=1}^{N_v}$. This choice makes the number of points that fall out of the prediction region equal to $r_\tau - 1$ which implies that the fraction of such points out of the region is equal to $\frac{r_\tau - 1}{N_V} = \frac{\lceil \tau N_v \rceil - 1}{N_v} \le \frac{\tau N_v}{N_v} = \tau$. Thus, the number of points out of the region do not exceed τ as desired. On the other hand, if i.i.d. samples are available, one could resort to the concept of probabilistic scaling to choose the value of r_τ and N_v (see [26–28]). In that case, two levels of probability results are obtained that apply to any pair $(\bar{z}, \bar{y})$ generated according to the underlying distribution of (z, y).

4.5. Comparison with the approach presented in [21]

In this subsection, we show that the proposed approach is superior to the one presented in [21] in terms of computational complexity. For this purpose, we consider the example presented in [21] corresponding to the Lorenz attractor. This is a chaotic system following the set of differential equations:

$$\begin{aligned} \frac{\mathrm{d}o}{\mathrm{d}t} &= \sigma(p - o) \\ \frac{\mathrm{d}p}{\mathrm{d}t} &= o(\rho - q) - p \\ \frac{\mathrm{d}q}{\mathrm{d}t} &= op - \beta q , \end{aligned}$$

where σ, ρ and β are real scalar parameters. In this example, these parameters take the values $\sigma = 10$, $\rho = 28$ and $\beta = 8/3$. Considering a sample time of $T_s = 0.1$ s, we tackle the problem of obtaining 1−step ahead interval predictions of the output of the system, that is $y_k = o_k$. The regressor z_k is composed of the



(a) $\gamma = 0$ region. (b) $\gamma = 0.2$ region. (c) $\gamma = 0.4$ region. (d) $\gamma = 0.8$ region.

Fig. 3. Prediction regions for different values of γ.

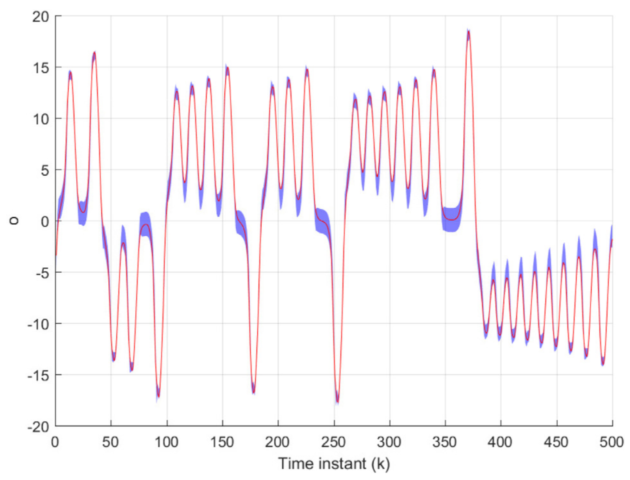


Fig. 4. Computed intervals for the proposed approach ($\tau = 0.2$).

two previous values of the outputs, i.e. $z_k = [y_{k-1}, y_{k-2}]^\top$. The size of the data set, validation set and test set is 300, 500 and 500 respectively. The considered finite family of γ, that is Γ, is composed of the values [0, 0.1, 0.2]. Also, we consider the same grid on the probability distribution as in [21].

The results are shown in Fig. 4 and Table 2. There, it can be seen that the new approach attains a much smaller computational time. Also, the obtained interval widths and empirical probabilities are better (that is, the interval is smaller and the empirical probability higher). Note that the computational time corresponds to the total amount of required time to calculate the hyperparameters and compute the intervals for all the points in the test set.

Table 2
Numerical results of the example in [21].

	Interval width	Empirical probability	Time (s)
Approach in [21]	2.0352	0.8900	296.47
Proposed	1.9773	0.8980	6.9201

5. Ellipsoidal regions

Due to the term $\gamma \sum_{i=1}^{N} |\lambda_i|$ appearing in the definition of the dissimilarity function, the prediction regions cannot be computed explicitly. Instead, in order to check if a point y belongs to a prediction region, a convex optimization problem must be solved. We provide in this section a quadratic upper bound to the dissimilarity function that leads to ellipsoidal regions that can be computed in an explicit way. The idea is to upper bound the absolute values $\{|\lambda_i|\}_{i=1}^{N}$ with scalar quadratic functions. This transforms the functional into a quadratic one and thus the optimization problem can be solved explicitly since the minimization of a convex quadratic function subject to linear equality constraints has an explicit solution. In order to upper bound the absolute values, we use the following lemma.

Lemma 2 (*Quadratic Upper Bound Of* $|\lambda_i|$). *Given a scalar* λ_i^c *and* $\nu > 0$,

$$|\lambda_i| \leq \frac{|\lambda_i^c| + \nu}{2} \left(\left(\frac{|\lambda_i|}{|\lambda_i^c| + \nu} \right)^2 + 1 \right), \quad \forall \lambda_i \in \mathcal{R}. \tag{13}$$

Proof. See Appendix B in [29]. ■

We choose λ_i^c as the λ_i^* used to obtain the central prediction $\hat{y}(z)$ (8). We note that the upper bound provided by Lemma 2 is tight when $\lambda_i = \lambda_i^c = \lambda_i^*$ and $\nu \to 0$. Thus, we choose ν as a small constant. Applying Eq. (13) to the optimization problem in (6), we obtain

$$J_\gamma(z,y) \leq Q_\gamma(z,y),$$

where

$$Q_\gamma(z,y) = \min_{\lambda_1,\ldots,\lambda_N} \frac{1}{2}\lambda^T H_\gamma \lambda + c_\gamma \tag{14a}$$

$$\text{s.t.} \begin{bmatrix} \mathcal{Z} \\ \mathbf{1} \end{bmatrix} \lambda = \begin{bmatrix} z \\ 1 \end{bmatrix} \tag{14b}$$

$$\mathcal{Y}\lambda = y, \tag{14c}$$

H_γ is a diagonal matrix with entries

$$H_{\gamma|i,i} = 2(1-\gamma)w_i + \frac{\gamma}{|\lambda_i^c| + \nu},\ i = 1,\ldots,N$$

and $c_\gamma = \gamma \sum_{i=1}^N \frac{|\lambda_i^c|+\nu}{2}$. The following lemma characterizes the new dissimilarity function $Q_\gamma(\cdot)$.

Lemma 3. *Assume that $H_\gamma > 0$ and that the matrix $\begin{bmatrix} \mathcal{Z}^T & \mathcal{Y}^T & \mathbf{1}^T \end{bmatrix}^T$ is full rank. Denote*

$$Q_\gamma(z,y) = \min_{\lambda_1,\ldots,\lambda_N} \frac{1}{2}\lambda^\top H_\gamma \lambda + c_\gamma$$
$$\text{s.t.} \begin{bmatrix} \mathcal{Z} \\ \mathcal{Y} \\ \mathbf{1} \end{bmatrix} \lambda = \begin{bmatrix} z \\ y \\ 1 \end{bmatrix},$$

$$y^* = \arg\min_y\ Q_\gamma(z,y),$$
$$Q_\gamma^*(z) = Q_\gamma(z,y^*).$$

Then,

$$y^* = {\Gamma_{1,2}}^\top \Gamma_{1,1}^{-1} \begin{bmatrix} z \\ 1 \end{bmatrix},\ Q_\gamma^*(z) = \frac{1}{2}\begin{bmatrix} z \\ 1 \end{bmatrix}^\top \Gamma_{1,1}^{-1} \begin{bmatrix} z \\ 1 \end{bmatrix} + c_\gamma$$

$$Q_\gamma(z,y) = Q_\gamma^*(z) + \frac{1}{2}(y-y^*)^\top \Phi_{2,2}(y-y^*).$$

where

$$\Gamma_{1,1} = \begin{bmatrix} \mathcal{Z} \\ \mathbf{1} \end{bmatrix} H_\gamma^{-1} \begin{bmatrix} \mathcal{Z} \\ \mathbf{1} \end{bmatrix}^\top,\ \Gamma_{1,2} = \begin{bmatrix} \mathcal{Z} \\ \mathbf{1} \end{bmatrix} H_\gamma^{-1}\mathcal{Y}^\top,$$
$$\Gamma_{2,2} = \mathcal{Y}H_\gamma^{-1}\mathcal{Y}^\top,\ \Phi_{2,2} = \left(\Gamma_{2,2} - {\Gamma_{1,2}}^\top \Gamma_{1,1}^{-1}\Gamma_{1,2}\right)^{-1}.$$

Proof. The proof is shown in Appendix. ■

We now define, by means of the new dissimilarity function $Q_\gamma(z,y)$, the following ellipsoidal regions

$$\begin{aligned}\mathcal{E}_{\gamma,\alpha}(z) &= \left\{y : Q_\gamma(z,y) \leq \alpha Q_\gamma^*(z)\right\} \\ &= \left\{y : (y-y^*)^\top \Phi_{2,2}(y-y^*) \leq 2(\alpha-1)Q_\gamma^*(z)\right\},\end{aligned}$$

where, the values for y^*, $\Phi_{2,2}$ and $Q_\gamma^*(z)$ are given in Lemma 3.

We note that, given $\gamma \geq 0$, the optimal value for α could be obtained by means of a validation set $\mathcal{V}$ as in the case of implicit regions. In this case, the scalars $\tilde{\alpha}_j$ are defined as

$$\tilde{\alpha}_j = \frac{Q_\gamma(\tilde{z}_j,\tilde{y}_j)}{Q_\gamma^*(\tilde{z}_j)},\ j = 1,\ldots,N_v.$$

The optimal value for α would be the r_τ largest one, where $r_\tau = \lceil \tau N_v \rceil$. We also remark that, in this case, one could obtain γ in such a way that a measure of the size of the obtained ellipsoids is minimized.

6. Numerical example

In this section, we propose as an example the application of the proposed methodology to compute prediction regions for the predictions of the outputs of two different multivariable systems.

6.1. Towing kite

In this section, we propose as an example the application of the proposed methodology to compute prediction regions for the predictions of the outputs of a multivariable system. As the system to be forecasted, we consider the non-linear model of a towing kite presented in [30]. It has three states, which corresponds to the angles of the kite, θ, ϕ and φ, a control input u and two uncertain parameters. The equations governing the system are the following

$$\dot{\theta} = \frac{v_a}{L_T}\left(\cos\varphi - \frac{\tan\theta}{E}\right),$$
$$\dot{\phi} = -\frac{v_a}{L_T \sin\theta}\sin\varphi,$$
$$\dot{\varphi} = \frac{v_a}{L_T}u + \dot{\phi}\cos\theta,$$

where

$$v_a = v_0 E\cos\theta,$$
$$E = E_0 - 0.028u.$$

From this system of equations, the parameter L_T (length of the tether) is considered to be known whereas v_0 (wind speed) and E_0 (base glide ratio) are considered to be uncertain parameters. Note that these uncertainties affect directly v_a, which can be described as the effect of the wind and the glide ratio E, and thus it also affects the states θ, ϕ and φ. Furthermore, only the states θ and ϕ are considered as outputs with an additive correlated gaussian noise so that

$$\mathcal{N}(\mu,\Sigma):\ \mu = \begin{bmatrix} 0 \\ 0 \end{bmatrix},\ \Sigma = \begin{bmatrix} 0.01 & -0.0085 \\ -0.0085 & 0.01 \end{bmatrix}.$$

We assume that the control input u is computed following a certain stabilizing control policy $\kappa(\theta,\phi,\varphi)$ whose exact nature is irrelevant for this paper (see [31] for more details on a possible control law). In fact, we treat the kite as an autonomous closed loop system. Thus, it is assumed that we have some stable trajectories of the closed-loop system for different values of the base glide ratio E_0 and the wind speed v_0 that are used to build the data sets $\mathcal{Z}$, $\mathcal{Y}$ with a sample time of $T_s = 0.15$ s. We consider two different data sets in this example, one with $N = 250$ samples and another one with $N = 500$ samples. Also, a validation set $\mathcal{V}$ composed of $N_v = 500$ samples to compute the optimal values of γ and α is considered. On the other hand, we have a test set $\mathcal{T}$ of $N_T = 1000$ samples to compare the obtained results with other baseline strategies. We consider a finite family of γ denoted as Γ. This Γ is composed of the values $\Gamma = \{0,\ 0.2,\ 0.4,\ 0.6,\ 0.8\}$.

In this example, the operator $h(\cdot)$ is chosen as a Neural Network (NN). That is, the NN is working as an encoder network. This means that the feature space of the whole data sets $\mathcal{Z}$ and $\mathcal{Y}$ are transformed by means of this given NN. In this case, the NN is comprised of 2 hidden layers of 10 neurons each one. Furthermore, the inputs of the NN are the last 5 outputs of the system. The outputs of the last layer are used as the regressor z. Note that this regressor z is the same for the proposed approach and the baselines and thus the following numerical comparisons are fair. Also, it is assumed that this NN is given. This means that the problems related to training appropriately the NN, finding the optimal number of layers, etc. are outside the scope of this paper.

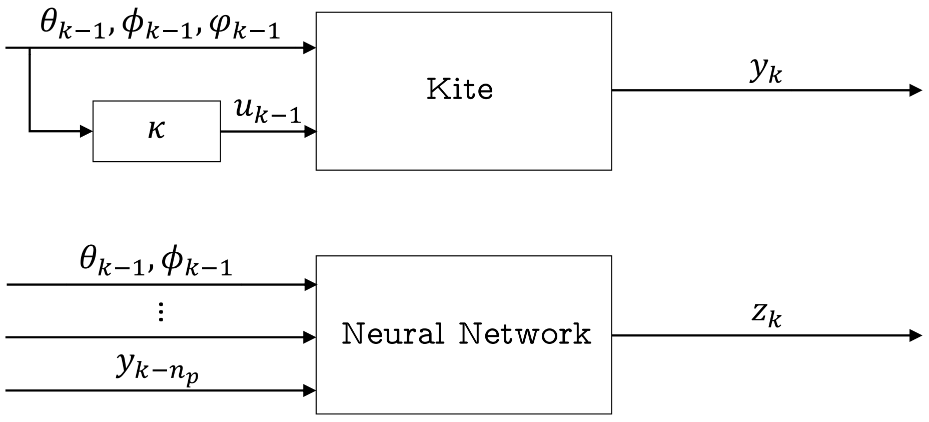

$\theta_{k-1}, \phi_{k-1}, \varphi_{k-1}$
κ
u_{k-1}
Kite
y_k
θ_{k-1}, ϕ_{k-1}
y_{k-n_p}
Neural Network
z_k

Fig. 5. Obtaining the data sets $\mathcal{Z}$ and $\mathcal{Y}$ for the kite example. Note that y_{k-j} corresponds to $(\theta_{k-j}, \phi_{k-j})$.

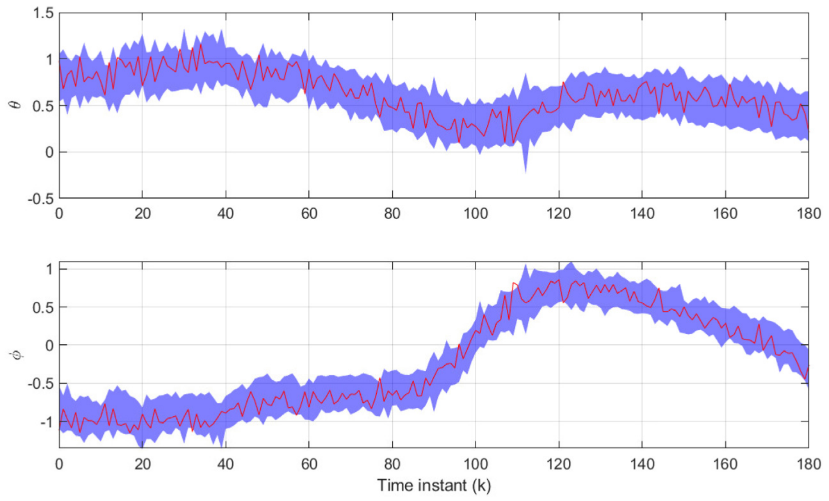


Fig. 6. Projected regions for a representative trajectory ($\tau = 0.2$).

Fig. 5 shows the flow of the data used to build the data sets $\mathcal{Z}$ and $\mathcal{Y}$ in this example. There, we note that the previous values of the output go into the neural network, whose output is the regressor z_k. On the other hand, the previous state along with the control input given by the control law $\kappa(\theta_{k-1}, \phi_{k-1}, \varphi_{k-1})$ go into the system, producing the next value of the output y_k. Then, the pair (z_k, y_k) is stored in the data sets $\mathcal{Z}$ and $\mathcal{Y}$.

Three baseline strategies are considered for the sake of comparison. First, a quantile regression (QR) approach. This technique is well established for the single output case. In order to obtain regions, a probabilistic interval of probability τ is obtained for each output independently and then combined to form a box-shaped region. However, combining two intervals of probability τ does not turn out to be a prediction region of probability τ. Actually, a lower probability will be attained. To tackle this problem, a back-off parameter is calculated so that the computed boxes fulfill the empirical probability in the validation set $\mathcal{V}$. The second one is based on GPs. Here, the matlab function “fitgrp” is used to obtain a model for each output independently. This function automatically finds an appropriate Kernel function for the input–output data and optimizes the values of the hyperparameters. Then, using “predict”, it is possible to obtain intervals of a specified probability τ. Same as before, it is necessary to compute a back-off parameter in the validation set $\mathcal{V}$ to achieve the desired empirical probability. Finally, an implementation of the prediction regions based on Inverse Regression (IR) of [16] is considered.

Fig. 6 shows the obtained predictions projected as intervals using the proposed approach for the case of a representative trajectory of the test set employing $\tau = 0.2$ and $N = 250$. It can be seen how the noisy trajectory (in red) is enclosed by the computed region most of the times, as it was expected.

The results for $\tau = 0.1$ and $\tau = 0.2$ are shown in Table 3. It can be seen how the implicit regions of the proposed approach and the approximation ellipsoids outperform the baselines considered for both values of τ and different data set sizes. That is, the proposed approach obtains regions of considerably smaller area while still fulfilling the specified probability given by τ (i.e. with an empirical probability (E.P.) no smaller than $1 - \tau$). This means that the proposed approach provides regions of smaller uncertainty in comparison to the baselines considered. It is clear that obtaining smaller regions will allow us to take better decisions if we implement the proposed methodology in an optimization setting.

The average time required for $N = 250$ to check if a certain point belongs to the region is 5.813 ms for the proposed approach (we allude here to the true region not the approximation). On the other hand, for the case of GPs, QR and IR, the values are 0.011 ms, 0.001 ms and 0.023 ms respectively.

Table 3
Area of the regions and empirical probabilities obtained for the proposed methodology, the approximation with ellipsoidal regions, Gaussian processes (GPs), quantile regression (QR) and inverse regression (IR).

		Proposed		Approximation		GPs		QR		IR	
	N	Area	E.P.	Area	E.P.	Area	E.P.	Area	E.P.	Area	E.P.
$\tau = 0.1$	250	0.1695	0.9200	0.1885	0.9170	0.2235	0.9250	0.5839	0.9120	0.4331	0.9031
	500	0.1551	0.9260	0.1656	0.9170	0.2092	0.9240	0.2645	0.8960	0.4101	0.9043
$\tau = 0.2$	250	0.1119	0.8070	0.1176	0.8070	0.1518	0.8450	0.2341	0.8550	0.2466	0.7446
	500	0.1053	0.8150	0.1142	0.8260	0.1403	0.8310	0.1590	0.8080	0.2335	0.7454

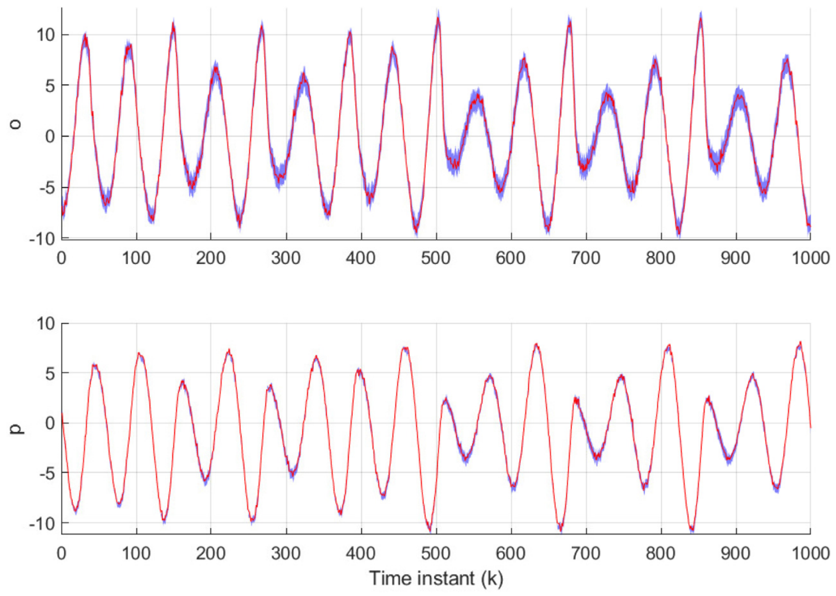


Fig. 7. Projected regions for a representative trajectory ($\tau = 0.2$).

6.2. Rössler attractor

As a different example, consider the system described by the following differential equations

$$\frac{\mathrm{d}o}{\mathrm{d}t} = -p - l$$
$$\frac{\mathrm{d}p}{\mathrm{d}t} = o + ap$$
$$\frac{\mathrm{d}l}{\mathrm{d}t} = b + l(o - c)\,,$$

which is known as the Rössler attractor. In this example, the parameters take the values $a = 0.2$, $b = 0.2$, $c = 5.7$ and the sample time is 0.1 s. The aforementioned system along with the proposed parameters show a chaotic behavior. Two different sizes (150 and 250 samples) for the data set ($\mathcal{Z}$ and $\mathcal{Y}$) are available. For the validation set $\mathcal{V}$ and the test set $\mathcal{T}$, we consider a total of 1000 samples in each one. The regressor is comprised of the past output of the system, i.e. $z_k = [o_{k-1}, p_{k-1}]^\top$. Note that this corresponds to choosing $h(\cdot)$ as the identity operator. Also, the output of the system is affected by some gaussian noise so that

$$\mathcal{N}(\mu, \Sigma):\ \mu = \begin{bmatrix} 0 \\ 0 \end{bmatrix},\ \Sigma = \begin{bmatrix} 0.05 & 0.025 \\ 0.025 & 0.05 \end{bmatrix}.$$

The baselines used for comparison will be the same as in the previous example: GPs, QR and IR. However, in this example, the IR method fails to obtain the desired regions (the obtained regions are several orders of magnitude larger), leading to unacceptable results. For this reason, we removed IR from the table of results.

The results are shown in Fig. 7, and Table 4. Fig. 7 shows the real evolution of the output and the prediction region obtained with the proposed approach and projected as in the previous example. Table 4 summarizes the numerical results of the experiment. It can be seen that the proposed strategy achieves the smallest areas. On the other hand, the ellipsoidal approximation obtains similar areas with respect to GPs when considering the small data set (150 samples) and smaller areas when considering the larger data set (250 samples).

7. Conclusions

This work presents a new data-based approach to obtain prediction regions to be used in nonlinear systems. The hyperparameters in which the methodology relies are computed so that the empirical probability is at least higher than the desired one in a validation set. Also, it is shown that checking if a point belongs to the computed region is as simple as solving a convex optimization problem. Finally, the proposed method has been used to generate prediction regions of a non-linear system, which have been compared favorably with the ones obtained by means of other baselines strategies based on Gaussian processes, quantile regression and inverse regression. As a future work, we consider the application of the proposed strategy in the context of developing novel stochastic or chance constrained MPC approaches, where the output of the prediction model is a whole region which heavily depend on uncertainty quantification and therefore could leverage the results of this paper. Another field of application would be failure detection, where the proposed technique could be used to obtain the regions of normal operation of the system. These regions could be then used to detect abnormal behaviors, which could be attributed to failures of the system.

Table 4
Area of the regions and empirical probabilities (E.P.) obtained for the proposed methodology, the approximation with ellipsoidal regions, Gaussian processes (GPs) and quantile regression (QR).

		Proposed		Approximation		GPs		QR	
	N	Area	E.P.	Area	E.P.	Area	E.P.	Area	E.P.
$\tau = 0.1$	150	2.3008	0.9010	2.5061	0.9100	2.4443	0.9050	33.914	0.9090
	250	2.0605	0.9020	2.0612	0.9030	2.3636	0.9110	31.966	0.8990
$\tau = 0.2$	150	1.4753	0.8000	1.5897	0.8040	1.5769	0.8040	9.1792	0.8060
	250	1.4080	0.7960	1.3504	0.8050	1.5424	0.8190	8.9740	0.7980

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported by the Agencia Estatal de Investigación (AEI)- Spain under Grant PID2019-106212RB-C41/AEI/10.13039/501100011033.

Appendix. Proof of Lemma 3

We first note that the optimization problem that defines $Q(z, y)$ is always feasible because of the full rank assumption on $\begin{bmatrix} \mathcal{Z}^T & \mathcal{Y}^T & \mathbf{1}^T \end{bmatrix}^T$. Moreover, because of the definite positiveness of H_γ, the problem has a unique minimum. For notational convenience, we define

$$A_1 = \begin{bmatrix} \mathcal{Z} \\ \mathbf{1} \end{bmatrix}, \; A_2 = \mathcal{Y}, \; A = \begin{bmatrix} A_1 \\ A_2 \end{bmatrix}, \; b = \begin{bmatrix} z \\ 1 \end{bmatrix}.$$

Consider now a related optimization problem in which the equality constraint $A_2\lambda = y$ is removed:

$$Q^*_\gamma(z) = \min_{\lambda_1,\ldots,\lambda_N} \frac{1}{2}\lambda^\top H_\gamma \lambda \quad \text{s.t } A_1\lambda = b.$$

It is clear that $Q^*_\gamma(z) \leq Q_\gamma(z, y)$, for every y. The optimal vector λ^* corresponding to the optimization problem that defines $Q^*_\gamma(z)$ can be directly obtained from the the Karush-Kuhn–Tucker (KKT) conditions (see Subsection 10.1.1 in [32]):

$$\lambda^* = H_\gamma^{-1}A_1^T(A_1H_\gamma^{-1}A_1^T)^{-1}b.$$

This leads to

$$Q^*_\gamma(z) = \frac{1}{2}(\lambda^*)^\top H_\gamma \lambda^* = \frac{1}{2}b^\top(A_1H_\gamma^{-1}A_1^T)^{-1}b = \frac{1}{2}b^\top \Gamma_{1,1}^{-1}b.$$

Denote now

$$y^* = A_2\lambda^* = A_2H_\gamma^{-1}A_1^T(A_1H_\gamma^{-1}A_1^T)^{-1} = {\Gamma_{1,2}}^\top \Gamma_{1,1}^{-1}b.$$

From $A_1\lambda^* = b$, we have that λ^* is also a feasible solution for the optimization problem corresponding to $Q_\gamma(z, y^*) = Q_\gamma(z, A_2\lambda^*)$. Thus,

$$Q_\gamma(z, y^*) = Q^*_\gamma(z).$$

From this and the inequality $Q^*_\gamma(z) \leq Q_\gamma(z, y)$, $\forall y$, we obtain the first two claims of the lemma.

Denote λ^*_y as the optimal value for λ in the optimization problem that provides $Q_\gamma(z, y)$. Using again the Karush-Kuhn–Tucker (KKT) conditions, we obtain

$$\lambda^*_y = H_\gamma^{-1}A^T(AH_\gamma^{-1}A^T)^{-1}\begin{bmatrix} b \\ y \end{bmatrix}.$$

Thus,

$$Q_\gamma(z, y) = \frac{1}{2}{\lambda^*_y}^\top H_\gamma \lambda^*_y = \frac{1}{2}\begin{bmatrix} b \\ y \end{bmatrix}^\top (AH_\gamma^{-1}A^T)^{-1}\begin{bmatrix} b \\ y \end{bmatrix}$$
$$= \frac{1}{2}\begin{bmatrix} b \\ y \end{bmatrix}^\top \begin{bmatrix} \Gamma_{1,1} & \Gamma_{1,2} \\ {\Gamma_{1,2}}^\top & \Gamma_{2,2} \end{bmatrix}^{-1}\begin{bmatrix} b \\ y \end{bmatrix}$$

Now, making

$$\begin{bmatrix} \Gamma_{1,1} & \Gamma_{1,2} \\ {\Gamma_{1,2}}^\top & \Gamma_{2,2} \end{bmatrix}^{-1} = \begin{bmatrix} \Phi_{1,1} & \Phi_{1,2} \\ {\Phi_{1,2}}^\top & \Phi_{2,2} \end{bmatrix},$$

we obtain

$$Q_\gamma(z, y) = \frac{1}{2}\begin{bmatrix} b \\ y \end{bmatrix}^\top \begin{bmatrix} \Phi_{1,1} & \Phi_{1,2} \\ {\Phi_{1,2}}^\top & \Phi_{2,2} \end{bmatrix}\begin{bmatrix} b \\ y \end{bmatrix}.$$

It is well known that every quadratic function $q(y)$ with hessian Φ_q can be rewritten as $q(y) = q(y^*) + \frac{1}{2}(y - y^*)^\top \Phi_q(y - y^*)$. Thus,

$$Q_\gamma(z, y) = Q^*_\gamma(z) + \frac{1}{2}(y - y^*)^\top \Phi_{2,2}(y - y^*).$$

The matrix $\Phi_{2,2}$ can be obtained from the inverse of the partitioned matrix Γ using Schur complements (see e.g. Subsection 4.3.4 in [33]), that is

$$\Phi_{2,2} = \left(\Gamma_{2,2} - {\Gamma_{1,2}}^\top \Gamma_{1,1}^{-1}\Gamma_{1,2}\right)^{-1}. \quad \blacksquare$$

References

[1] Mc Clarren RG, Mc Clarren P, Penrose. Uncertainty quantification and predictive computational science. Springer; 2018.

[2] Villaverde AF, Raimúndez E, Hasenauer J, Banga JR. A comparison of methods for quantifying prediction uncertainty in systems biology. IFAC-PapersOnLine 2019;52(26):45–51.

[3] Birge JR, Louveaux F. Introduction to stochastic programming. Springer Science & Business Media; 2011.

[4] Charnes A, Cooper WW. Chance-constrained programming. Manage Sci 1959;6(1):73–9.

[5] Farina M, Giulioni L, Scattolini R. Stochastic linear model predictive control with chance constraints–a review. J Process Control 2016;44:53–67.

[6] Milanese M, Novara C. Set membership identification of nonlinear systems. Automatica 2004;40(6):957–75.

[7] Milanese M, Novara C. Unified set membership theory for identification, prediction and filtering of nonlinear systems. Automatica 2011;47(10):2141–51.

[8] Koenker R, Bassett G. Regression quantiles. Econometrica 1978;33–50.

[9] Davino C, Furno M, Vistocco D. Quantile regression: theory and applications, Vol. 988. John Wiley & Sons; 2013.

[10] Garatti S, Campi M, Care A. On a class of interval predictor models with universal reliability. Automatica 2019;110:108542.

[11] Lütkepohl H. Introduction to multiple time series analysis. Berlin: Springer; 1991.

[12] Wolf M, Wunderli D. Bootstrap joint prediction regions. J Time Series Anal 2015;36(3):352–76.

[13] Kim JH. Bias-corrected bootstrap prediction regions for vector autoregression. J Forecast 2004;23(2):141–54.

[14] Olive DJ. Applications of hyperellipsoidal prediction regions. Statist Papers 2018;59(3):913–31.

[15] Lin W, Zhuang Y, Zhang S, Martin E. On estimation of multivariate prediction regions in partial least squares regression. J Chemometr 2013;27(9):243–50.

[16] Devijver E, Perthame E. Prediction regions through inverse regression. J Mach Learn Res 2020;21(1):4532–55.

[17] Golestaneh F, Pinson P, Azizipanah-Abarghooee R, Gooi HB. Ellipsoidal prediction regions for multivariate uncertainty characterization. IEEE Trans Power Syst 2018;33(4):4519–30.
[18] Shafer G, Vovk V. A tutorial on conformal prediction. J Mach Learn Res 2008;9(3).
[19] Karydis K, Poulakakis I, Sun J, Tanner HG. Probabilistically valid stochastic extensions of deterministic models for systems with uncertainty. Int J Robot Res 2015;34(10):1278–95.
[20] Rasmussen CE, Williams CKI. Gaussian processes for machine learning. MIT Press; 2006.
[21] Carnerero AD, Ramirez DR, Alamo T. Probabilistic interval predictor based on dissimilarity functions. IEEE Trans Automat Control 2021. http://dx.doi.org/10.1109/TAC.2021.3136137.
[22] Alfonso G, Carnerero AD, Ramirez DR, Alamo T. Stock forecasting using local data. IEEE Access 2020;9:9334–44.
[23] Carnerero AD, Ramirez DR, Alamo T. State-space kriging: A data-driven method to forecast nonlinear dynamical systems. IEEE Control Syst Lett 2022.
[24] Beck A. First-order methods in optimization. SIAM; 2017.
[25] Beck A, Teboulle M. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. SIAM J Imaging Sci 2009;2(1):183–202.
[26] Alamo T, Tempo R, Luque A, Ramirez D. Randomized methods for design of uncertain systems: Sample complexity and sequential algorithms. Automatica 2015;52:160–72.
[27] Mirasierra V, Mammarella M, Dabbene F, Alamo T. Prediction error quantification through probabilistic scaling. IEEE Control Syst Lett 2022;6:1118–23.
[28] Mammarella M, Mirasierra V, Lorenzen M, Alamo T, Dabbene F. Chance-constrained sets approximation: A probabilistic scaling approach. Automatica 2022;137:110108.
[29] Alamo T, Cepeda A, Limon D. Improved computation of ellipsoidal invariant sets for saturated control systems. In: Proceedings of the 44th IEEE conference on decision and control. IEEE; 2005, p. 6216–21.
[30] Erhard M, Strauch H. Control of towing kites for seagoing vessels. IEEE Trans Control Syst Technol 2012;21(5):1629–40.
[31] Karg B, Alamo T, Lucia S. Probabilistic performance validation of deep learning-based robust NMPC controllers. Internat J Robust Nonlinear Control 2021;31(18):8855–76, [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/rnc.5696.
[32] Boyd S, Vandenberghe L. Convex optimization. Cambridge University Press; 2004.
[33] Murphy KP. Machine learning: a probabilistic perspective. MIT Press; 2012.