2024 IEEE 63rd Conference on Decision and Control (CDC)
December 16-19, 2024. MiCo, Milan, Italy

A data-driven CLF controller based on a kriged predictor

A. Daniel Carnerero, Daniel R. Ramirez, Takeshi Hatanaka, Teodoro Alamo

Abstract— **In this paper, we present a data-driven methodology to predict and control the behaviour of nonlinear and non-autonomous systems based on kernel functions. The technique computes the forecasting by means of a linear combination of past data. The weights used to compute the prediction are obtained by solving a convex optimization problem that stems from a novel kriging formulation. A Control Lyapunov Function (CLF) based controller using the presented predictor is also built. Finally, numerical examples of both prediction and control are presented, showing the efficacy of the proposed approach.**

I. Introduction

Recently, data-driven approaches have gained the attention of many researchers. In this kind of control method, a controller is obtained without any previous knowledge of the system dynamics. For this purpose, some experimental data is supposed to be available. Then, there are mainly two different ways of handling the aforementioned problem [1], namely the direct approach and the indirect approach.

In the direct approach, a control law compatible with the experimental data is directly obtained without identifying a model of the system. Examples of this approach are the data-driven controllers presented in [2], [3], the data-driven model predictive control schemes proposed in [4], [5], the behavioral approach in [6], [7] or the Reinforcement Learning approaches reviewed in [8], [9], among others.

On the other hand, carrying out a system identification process to obtain a model and, finally, synthesize a controller based on the obtained model is usually called the indirect approach. There are many techniques that result in different types of models like nonlinear autoregressive exogenous (NARX) models [10], Takagi-Sugeno models [11], Gaussian Processes [12], Neural Networks [13], the Koopman Operator [14], Reservoir Computing strategies [15] and the Kriging method [16], [17], [18] (also known in the literature as Direct Weight Optimization [19] and dissimilarity functions [20]).

Once a model of the system is obtained, different control strategies can be applied. For example, Model Predictive Control (MPC) [21] uses a system model to forecast the future behaviour of the system and compute the control actions that minimize a certain optimization criteria over a prediction horizon. However, the need of solving the MPC problem for an arbitrarily long prediction horizon might make the problem untractable for nonlinear systems. A possible way to leverage this burden would be to rely on Control Lyapunov Functions (CLF) [22]. In CLF controllers, an optimal control action fulfilling that the value of the Lyapunov function is strictly decreasing at the next step is computed. This works as if we are demanding the system to constantly dissipate a certain amount of energy. For example, see [23], [24] for some recent applications of CLFs in the context of data-driven control.

In this paper, we present a new data-driven control scheme based on a novel kriged prediction method (we refer the interested reader to [16], [17], [18] for a discussion about the prediction capabilities of these methods and some applications in the context of interval prediction, model predictive control, etc.). Here, we introduce a nonlinear transformation for the regressor, making it possible to tackle nonlinear systems from a different perspective. We demonstrate that explicit knowledge of this operator is not required due to the fact that the cross-products of the transformed regressors can be computed using a reproducing kernel function. Thanks to these changes, the nonlinearity is not handled by means of local data but using kernel functions, improving the prediction capabilities. Finally, a data-driven CLF controller built upon the proposed predictor is presented. The performance of the predictor and the controller is illustrated through a simulated example. There, it can be seen that the predictor can replicate satisfactorily trajectories of the nonlinear system and the data-driven CLF controller attains similar results than a CLF controller with perfect knowledge of the system.

The paper is organized as follows: Section II presents the predictor to be used alongside the paper whereas, in Section III, it is shown how the proposed predictor can be used to build a data-driven Control Lyapunov Function (CLF) controller. Section IV presents both prediction and control examples and, finally, Section V shows the conclusions and the expected future work.

II. Proposed Predictor

Consider a non-autonomous discrete-time nonlinear system

$$y_{k+1} = h(x_k, u_k) + v_k, \tag{1}$$

where $x_k \in \mathbf{R}^{n_x}$, $u_k \in \mathbf{R}^{n_u}$, $y_k \in \mathbf{R}^{n_y}$, $v_k \in \mathbf{R}^{n_y}$ are the state, input, output and measurement noise of the system at time k respectively and $h(\cdot,\cdot) : \mathbf{R}^{n_x \times n_u} \to \mathbf{R}^{n_y}$ is an unknown nonlinear function. We assume, as in the NARX model framework [25], that the state of the system x_k can

A.D. Carnerero is with the Graduate School of Engineering Science, Osaka University, Toyonaka 560-8531, Japan (e-mail: carnerero.daniel.es@osaka-u.ac.jp)

T. Hatanaka is with the Department of Systems and Control Engineering School of Engineering, Tokyo Institute of Technology, Tokyo, 152-8550, Japan (e-mail: hatanaka@sc.e.titech.ac.jp).

D.R. Ramirez and T. Alamo are with Departamento de Ingeniería de Sistemas y Automática, Universidad de Sevilla, Escuela Superior de Ingenieros, Camino de los Descubrimientos s/n, 41020 Sevilla, Spain (e-mail: {danirr,talamo}@us.es).

be expressed as

$$x_k = \left[\begin{array}{cccccc} y_k & \dots & y_{k-n_a} & u_{k-1} & \dots & u_{k-n_b} \end{array} \right], \quad (2)$$

with $n_a \in \mathbf{Z}$ and $n_b \in \mathbf{Z}$.

In this section, given a realization of x_k and u_k, a new data-driven method to estimate y_{k+1} by means of past data without explicitly building a model is proposed.

First, we assume that some past data of the system is stored in a database according to the following matrix structure

$$\begin{aligned} D &= \left[\begin{array}{cccc} \bar{x}_1 & \bar{x}_2 & \dots & \bar{x}_N \end{array} \right], \\ D^+ &= \left[\begin{array}{cccc} \bar{y}_1^+ & \bar{y}_2^+ & \dots & \bar{y}_N^+ \end{array} \right], \\ U &= \left[\begin{array}{cccc} \bar{u}_1 & \bar{u}_2 & \dots & \bar{u}_N \end{array} \right], \end{aligned}$$

where N is the number of data points, $U \in \mathbf{R}^{n_u \times N}$ is the database of control actions, $D \in \mathbf{R}^{n_z \times N}$ contains past samples of the state and $D^+ \in \mathbf{R}^{n_z \times N}$ contains the next output of D. Assuming that x is given, it is possible to obtain a prediction of the next output $\tilde{y}^+$ by solving the following optimization problem [16], [17]

$$\lambda^* = \arg\min_{\lambda} (1-\gamma) \sum_{i=1}^{N} \lambda_i^2 + \gamma \sum_{i=1}^{N} |\lambda_i| \quad (3a)$$

$$\text{s.t. } x = \sum_{i=1}^{N} \lambda_i \bar{x}_i \quad (3b)$$

$$u = \sum_{i=1}^{N} \lambda_i \bar{u}_i \quad (3c)$$

$$1 = \sum_{i=1}^{N} \lambda_i, \quad (3d)$$

where $\gamma \in [0, 1)$ is a tuning parameter. Then, the prediction $\tilde{y}^+$ can be computed by means of the optimal value of λ^* and the data set, that is

$$\tilde{y}^+ = D^+ \lambda^*. \quad (4)$$

To tackle the nonlinearity of the system, the easiest method would be to consider a slightly different cost function

$$\sum_{i=1}^{N} \omega_i \lambda_i^2 + \gamma \sum_{i=1}^{N} |\lambda_i| \quad (5)$$

where ω_i corresponds to a certain weight that can be computed as

$$\omega_i = ||x - \bar{x}_i||, \quad (6)$$

that is, locality is now weighted in the cost function. We refer the reader to [17] for more details about the original predictor and the dissimilarity functions involved.

On the other hand, in the context of State-Space Kriging (SSK) (which involves similar optimization problems), it was shown in [18] that the use of kernel functions might attain better prediction accuracy and computation times. However, the kernel function in the SSK presented in [18] does not consider the inclusion of past inputs in the kernel. This implies that some nonlinearities of the system might remain unmodeled.

In order to address the above issue, we introduce kernel functions in the aforementioned optimization problem for both the state and the input, improving the prediction capabilities of such predictors. First, let us denote z as

$$z = \left[\begin{array}{c} x \\ u \end{array} \right] \in \mathbf{R}^{n_z}, \quad n_z = n_x + n_u \quad (7)$$

and thus we can define a new data set

$$Z = \left[\begin{array}{c} D \\ U \end{array} \right] = \left[\begin{array}{cccc} \bar{x}_1 & \bar{x}_2 & \dots & \bar{x}_N \\ \bar{u}_1 & \bar{u}_2 & \dots & \bar{u}_N \end{array} \right]. \quad (8)$$

Then, (3) can be rewritten as

$$\lambda^* = \arg\min_{\lambda} \ (1-\gamma)\lambda^\top \lambda + \gamma \|\lambda\|_1 \quad (9a)$$

$$\text{s.t. } Z\lambda = z \quad (9b)$$

$$\mathbf{1}^\top \lambda = 1, \quad (9c)$$

where $\|\lambda\|_1 = \sum_{i=1}^{N} |\lambda_i|$. Now, we propose to transform the hard constraint (9b) into a soft constraint in a probably high dimensional space $\mathbf{H}$. Note that the transformation of the hard constraint into a soft constraint makes sense in this context since the hard constraint might become unfeasible in $\mathbf{H}$. Then, the following optimization problem is obtained

$$\lambda^* = \arg\min_{\lambda} \ (1-\gamma)\lambda^\top H_1 \lambda + \gamma \|\lambda\|_1 + \left\| \sum_{i=1}^{N} \varphi_{\bar{z}_i} \lambda_i - \varphi_z \right\|_{\Sigma_\varphi^{-1}}^2 \quad (10a)$$

$$\text{s.t. } \mathbf{1}^\top \lambda = 1\,, \quad (10b)$$

where H_1 is a weighting matrix, $\varphi(\cdot) : \mathbf{R}^{n_z} \to \mathbf{H}$ refers to a nonlinear operator mapping $\mathbf{R}^{n_z}$ into a probably high dimensional space $\mathbf{H}$, $\varphi_{\bar{z}_i}$ and φ_z denote $\varphi(\bar{z}_i)$ and $\varphi(z)$ respectively and Σ_φ is a positive definite matrix of appropriate dimensions. Note that, for example, $\varphi(\cdot)$ could be a function such that, given $z \in \mathbf{R}^{n_z}$, it returns the products and cross-products of the elements of z up to order p, converting the previous problem to one similar of polynomial regression, improving the potential to tackle nonlinear correlations within the regressor. Now, denoting $\varphi_{\bar{z}}$ as

$$\varphi_{\bar{z}} = \left[\begin{array}{cccc} \varphi_{\bar{z}_1} & \varphi_{\bar{z}_2} & \dots & \varphi_{\bar{z}_N} \end{array} \right],$$

it is possible to write the previous optimization problem in a more compact form as follows:

$$\lambda^* = \arg\min_{\lambda} \ (1-\gamma)\lambda^\top H_1 \lambda + \gamma \|\lambda\|_1 + \|\varphi_{\bar{z}}\lambda - \varphi_z\|_{\Sigma_\varphi^{-1}}^2 \quad (11a)$$

$$\text{s.t. } \mathbf{1}^\top \lambda = 1\,. \quad (11b)$$

Then, the term $\|\varphi_{\bar{z}}\lambda_i - \varphi_z\|_{\Sigma_\varphi^{-1}}^2$, can be rewritten as

$$\lambda^\top \varphi_{\bar{z}}^\top \Sigma_\varphi^{-1} \varphi_{\bar{z}} \lambda - 2\varphi_z^\top \Sigma_\varphi^{-1} \varphi_{\bar{z}} \lambda + \varphi_z^\top \Sigma_\varphi^{-1} \varphi_z. \quad (12)$$

The constant term $\varphi_z^\top \Sigma_\varphi^{-1} \varphi_z$ does not depend on λ and thus it will not affect the value of λ^*, which means that we can

get rid of it. Now, taking into account the first term in the cost function of (11), that is, $\lambda^\top H_1 \lambda$, and making

$$H = 2(\varphi_{\bar{z}}^\top \Sigma_\varphi^{-1} \varphi_{\bar{z}}) + 2(1-\gamma)H_1,\ f^\top = -2\varphi_z^\top \Sigma_\varphi^{-1} \varphi_{\bar{z}}, \tag{13}$$

optimization problem (11) can be written as

$$\lambda^* = \arg\min_\lambda \ \frac{1}{2}\lambda^\top H \lambda + f^\top \lambda + \gamma \|\lambda\|_1 \tag{14a}$$

$$\text{s.t. } \mathbf{1}^\top \lambda = 1. \tag{14b}$$

In case that $\gamma = 0$, this becomes a quadratic programming problem (QP) with only equality constraints and the value of λ^* can be computed by means of the following equation

$$\lambda^* = -H^{-1}f + H^{-1}\mathbf{1}(\mathbf{1}^\top H^{-1}\mathbf{1})^{-1}(1 + \mathbf{1}^\top H^{-1} f), \tag{15}$$

where the above equation corresponds to the solution obtained from the Karush-Kuhn-Tucker conditions [26]. Note that in the case of $\gamma > 0$, the problem is still a strict convex optimization problem subject to convex constraints and thus it has a unique solution [26]. Then, the numerical computation can be addressed using a dual formulation and a accelerated gradient method [27].

In what follows, we show that we do not need precise knowledge about $\varphi(\cdot)$. As $\varphi(\cdot)$ is an operator mapping a vector in a space $\mathbf{R}^{n_z}$ to a reproducing kernel Hilbert space $\mathbf{H}$, it is well known that we can compute the cross-product of two elements in such space by means of a reproducing kernel function, which is commonly known in the literature as the kernel trick [28]. Thus, if we can pose every element in the optimization problem in (14) as cross-products, the problem can be solved without explicit knowledge of $\varphi(\cdot)$.

First, for a given pair $a \in \mathbf{R}^{n_z}$ and $b \in \mathbf{R}^{n_z}$, we denote $\langle \varphi_a, \varphi_b \rangle = \varphi_a \Sigma_\varphi^{-1} \varphi_b$. Then, it is clear that $\varphi_{\bar{z}}^\top \Sigma_\varphi^{-1} \varphi_{\bar{z}}$ and $\varphi_z^\top \Sigma_\varphi^{-1} \varphi_{\bar{z}}$ can be written as

$$\varphi_{\bar{z}}^\top \Sigma_\varphi^{-1} \varphi_{\bar{z}} = \begin{bmatrix} \langle \varphi_{\bar{z}_1}, \varphi_{\bar{z}_1} \rangle & \langle \varphi_{\bar{z}_1}, \varphi_{\bar{z}_2} \rangle & \cdots & \langle \varphi_{\bar{z}_1}, \varphi_{\bar{z}_N} \rangle \\ \langle \varphi_{\bar{z}_2}, \varphi_{\bar{z}_1} \rangle & \langle \varphi_{\bar{z}_2}, \varphi_{\bar{z}_2} \rangle & \cdots & \langle \varphi_{\bar{z}_2}, \varphi_{\bar{z}_N} \rangle \\ \vdots & \vdots & & \vdots \\ \langle \varphi_{\bar{z}_N}, \varphi_{\bar{z}_1} \rangle & \langle \varphi_{\bar{z}_N}, \varphi_{\bar{z}_2} \rangle & \cdots & \langle \varphi_{\bar{z}_N}, \varphi_{\bar{z}_N} \rangle \end{bmatrix},$$

$$\varphi_z^\top \Sigma_\varphi^{-1} \varphi_{\bar{z}} = \begin{bmatrix} \langle \varphi_z, \varphi_{\bar{z}_1} \rangle & \langle \varphi_z, \varphi_{\bar{z}_2} \rangle & \cdots & \langle \varphi_z, \varphi_{\bar{z}_N} \rangle \end{bmatrix}.$$

As only cross-products appear in the above matrices, every element can be computed by means of a kernel function without explicitly knowing the operator $\varphi(\cdot)$. There are different kernel functions that can be used with the proposed methodology [29].

Thus, given x_k, it is possible to build the appropriate kernel matrices and solve the optimization problem in equation (14) to obtain λ_k^*. Then, by means of λ_k^*, it is possible to compute the prediction of the next output using equation (4) and, finally, the state can be built as

$$\tilde{x}_{k+1} = \begin{bmatrix} \tilde{y}_{k+1} & \cdots & y_{k-n_a+1} & u_k & \cdots & u_{k-n_b+1} \end{bmatrix}. \tag{16}$$

For the sake of simplicity, we denote the previously obtained prediction as

$$\tilde{x}_{k+1} = g(x_k, u_k), \tag{17}$$

where $g(\cdot,\cdot) : \mathbf{R}^{n_x \times n_u} \to \mathbf{R}^{n_x}$ is a function encapsulating the proposed predictor.

III. Control Lyapunov Function

Control Lyapunov Functions (CLF) correspond to the generalization of the traditional concept of Lyapunov functions for autonomous systems to manipulable systems.

Definition 3.1 (Control Lyapunov Function): A function $V : \mathbf{R}^{n_x} \to \mathbf{R}_+$ is said to be a CLF of the system

$$x^+ = f(x,u)$$

where $x \in \mathbf{R}^{n_x}, u \in \mathbf{R}^{n_u}$, $f(\cdot,\cdot) : \mathbf{R}^{n_x \times n_u} \to \mathbf{R}^{n_x}$, if it is positive definite and, for a certain set $\mathcal{B}$, it exists a control input u so that

$$V(f(x,u)) - V(x) \leq 0, \quad \forall x \in \mathcal{B}.$$

Note that a system is asymptotically controllable to the origin only if and only if it is possible to find an input u satisfying the above condition. Also, for that purpose, it is needed to upper bound the decrease rate of the CLF as follows

$$V(f(x,u)) - V(x) \leq -\theta(||x||), \quad \forall x \in \mathcal{B},$$

where $\theta(\cdot)$ is a $\mathcal{K}$ function (see definition 3.3 in [30]). Note that if the condition on the decrease rate cannot be guaranteed, it is possible to obtain stability but not asymptotic stability properties.

Then, in order to find an input u satisfying the CLF condition, we can solve an optimization problem like the following

$$\min_u \ u^\top R u \tag{18a}$$

$$\text{s.t. } V(f(x,u)) - V(x) \leq -\theta(||x||) \tag{18b}$$

$$u \in \mathcal{U}, \tag{18c}$$

where R is a weighting matrix and $\mathcal{U}$ is the set of feasible inputs. In case we would like to track the system to a certain reference r, the previous optimization problem is modified as follows:

$$\min_u \ (u - u_{\text{ref}})^\top R (u - u_{\text{ref}}) \tag{19a}$$

$$\text{s.t. } V(f(x,u) - r) - V(x - r) \leq -\theta(||x - r||) \tag{19b}$$

$$u \in \mathcal{U}, \tag{19c}$$

where the pair $\{r, u_{\text{ref}}\}$ fulfills the equilibrium condition

$$r = f(r, u_{\text{ref}}). \tag{20}$$

In this paper, we consider using the proposed predictor as the model of the system, obtaining the following optimization problem

$$\min_u \ (u - u_{\text{ref}})^\top R (u - u_{\text{ref}}) \tag{21a}$$

$$\text{s.t. } V(g(x,u) - r) - V(x - r) \leq -\theta(||x - r||) \tag{21b}$$

$$u \in \mathcal{U} \tag{21c}$$

where $g(\cdot,\cdot)$ corresponds to the predictor in equation (17). In order to preserve the recursive feasibility of the optimization

problem, the following relaxation is commonly used

$$\min_{u,\sigma} \ (u-u_{\text{ref}})^\top R(u-u_{\text{ref}}) + p\,\sigma^\kappa \tag{22a}$$

$$\text{s.t. } V(g(x,u)-r) - V(x-r) \leq -\theta(||x-r||) + \sigma \tag{22b}$$

$$u \in \mathcal{U}, \tag{22c}$$

where $\kappa \in \mathbf{Z}$, σ is a slack variable and p is a high-valued constant. Depending on the application, different values of κ can be considered. For example, in the context of penalty functions and exact approximations, the l_1 norm is used frequently thanks to its good properties (see chapter 6 in [26]) whereas, in the literature of CLF and Control Barrier Function (CBF) controllers, the most popular choice is $\kappa = 2$ (see [31]). Here, we opt to follow the tendency of CBF and CLF controllers and thus make $\kappa = 2$.

Remark 1: We point out that there are many ways to address the problem of finding an appropriate Lyapunov function for the nonlinear system to be controlled. For example, in [32], the difference between the true nonlinear system and a linear system is bounded by means of the disturbance term. Then, it is possible to build a quadratic Lyapunov function for the linear system. Another similar way would be to obtain a linearized model of the plant and build a linear controller for this model. Then, we can use this Lyapunov function with our proposed approach to extend its domain of attraction thanks to its good prediction capabilities.

IV. Numerical examples

In this section, the effectiveness of the proposed data-driven approach is tested in both a prediction and a control example. As a benchmark, the quadruple-tank process [33] will be used. The system consists of four interconnected water deposits whose dynamics are given by the following set of differential equations:

$$\frac{dh_1}{dt} = -\frac{a_1}{A_1}\sqrt{2gh_1} + \frac{a_3}{A_1}\sqrt{2gh_3} + \frac{\gamma_1 k_1}{A_1}v_1, \tag{23a}$$

$$\frac{dh_2}{dt} = -\frac{a_2}{A_2}\sqrt{2gh_2} + \frac{a_4}{A_2}\sqrt{2gh_4} + \frac{\gamma_2 k_2}{A_2}v_2, \tag{23b}$$

$$\frac{dh_3}{dt} = -\frac{a_3}{A_3}\sqrt{2gh_3} + \frac{(1-\gamma_2)k_2}{A_3}v_2, \tag{23c}$$

$$\frac{dh_4}{dt} = -\frac{a_4}{A_4}\sqrt{2gh_4} + \frac{(1-\gamma_1)k_1}{A_4}v_1, \tag{23d}$$

where $h^\top = \begin{bmatrix} h_1 & h_2 & h_3 & h_4 \end{bmatrix}^\top$ is the state of the system, $v^\top = \begin{bmatrix} v_1 & v_2 \end{bmatrix}^\top$ is the input of the system and everything else are parameters whose values are shown in table I. Also, the considered sample time is $T_s = 30$s and $\gamma = 0$.

A. Prediction example

In this subsection, the accuracy of the proposed approach will be tested against the predictions obtained by using the predictor from Remark 1 in [16] to show the corresponding improvements of the predictions. For this purpose, a random trajectory of the system, that is, given a certain sequence of N_t inputs, the evolution of the system from time $k = 0$ until N_t will be computed in an open loop manner (only the initial state is given and the output will not be measured again).

Parameter	Meaning	Value
A_1,A_2,A_3,A_4	Cross section of the tanks	0.03
a_1	Discharge constant	1.3104×10^{-4}
a_2	Discharge constant	1.5074×10^{-4}
a_3	Discharge constant	9.2673×10^{-5}
a_4	Discharge constant	8.8164×10^{-5}
g	Gravity acceleration	9.81
γ_1	Valve aperture	0.3
γ_2	Valve aperture	0.4
k_1, k_2	Heat transfer coefficient	1/3600

TABLE I
Parameters of the model

The data set, validation set and test set contain a number of $N = 300$, $N_v = 300$ and $N_t = 300$ samples respectively. Note that the validation set is used to tune the parameters of the proposed approach, i.e. the kernel function and the matrix H_1. For the kernel function, we have chosen a quadratic kernel function so that:

$$K(a,b) = (\alpha + a^\top b)^2 \tag{24}$$

where α is a tunable parameter. On the other hand, H_1 is chosen to be a diagonal matrix

$$H_1 = \beta\, \mathrm{I}_N \tag{25}$$

where I_N is the $N \times N$ identity matrix and β is a tunable parameter. The considered criteria to optimize the parameters α and β is the following:

$$\min_{\alpha,\beta} \ \frac{1}{N_{\text{v}}}\sum_{j=1}^{N_{\text{v}}}(\tilde{h}_j(\alpha,\beta) - \bar{h}_j)^2 \tag{26a}$$

$$\text{s.t. } \alpha \geq 0, \quad \beta \geq 0. \tag{26b}$$

That is, minimizing the Mean-Squared Error (MSE) of the predictions in the validation set.

First, we assume that the state is measurable and no measurement error is present, i.e. $y_k = x_k$. The results are shown in Figure 1. Note that the whole trajectory is computed without any kind of feedback given the initial condition. The obtained MSE is 1.1755×10^{-4}. On the other hand, the chosen baseline predictor using local data obtained a MSE of 8.4882×10^{-4}, which is almost 8 times bigger than the previous one.

Next, we assume that the output is noisy. In particular, we consider an additive random noise with an uniform probability distribution ranging in the interval $[-0.15, 0.15]$, which can be considered quite high taking into account the order of magnitude of the state of the system. The results are shown in Figure 2. There, it can be seen that the addition of noise to the measurements does not change drastically the performance of the predictor. The new values of the MSE for the proposed approach and the baseline are 0.0127 and 0.0142 respectively.

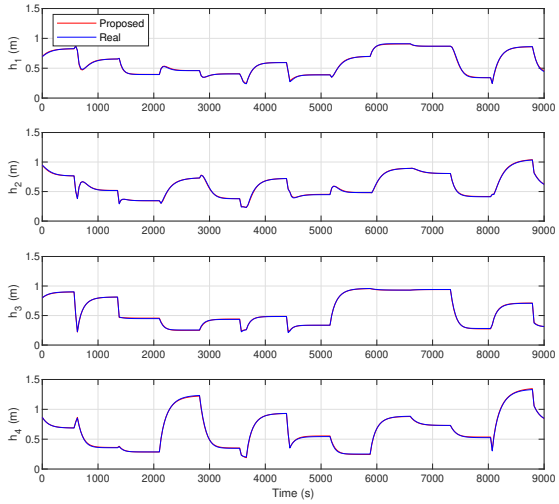

Fig. 1. Noiseless test trajectory.

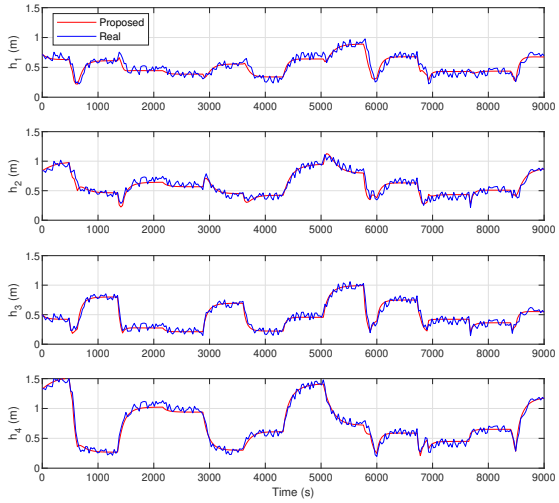

Fig. 2. Noisy test trajectory.

B. Control example

In this subsection, the proposed predictor will be used within a CLF controller as it was shown in section III. As the Lyapunov function $V(x)$, we choose the tracking error, that is $(x-r)^\top(x-r)$. The output is affected by a random uniform noise within the range [-0.025,0.025] meters.

Here, the sampling time is $T_s = 10$ seconds and the number of data samples is $N = 1000$. The simulations lasts for 4000 seconds, i.e. 400 time instants. The desired reference changes every 1000 seconds, making a total of four reference changes during the simulation. Along with the proposed approach, a baseline controller is also shown. Not only this controller is a CLF controller with perfect knowledge of the system, but it can also measure the state without being affected by the measurement noise.

The results are shown in figures 3, 4 and 5. Figure 3 shows the evolution of the output of the system during the simulation. It is easy to see that, not only the desired reference is tracked, but also the performance is very similar to the one obtained by the controller with perfect knowledge, as it was expected due to the good prediction capabilities of the kernel-based proposed data-driven predictor.

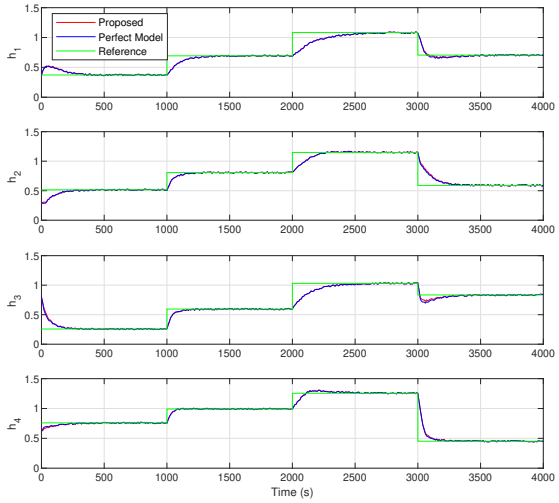

Fig. 3. Output of the system for the closed loop simulation.

In figure 4, the values of the input applied to the system are shown. Similarly, the computed control actions by the proposed approach are almost identical to the ones obtained by the baseline controller in spite of the potential model mismatches and the measurement noise.

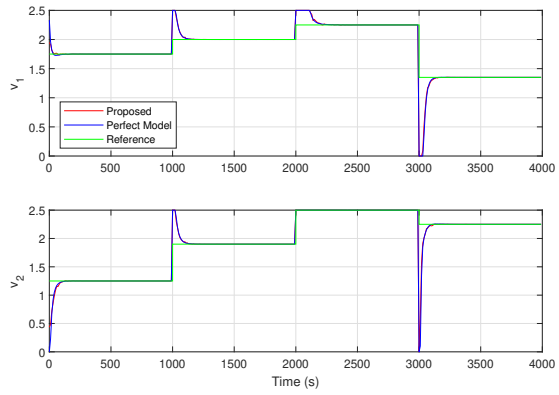

Fig. 4. Input applied to the system for the closed loop simulation.

Finally, the evolution of the Lyapunov function is shown in figure 5. Note that the peaks in the Lyapunov function correspond to the changes in the reference. Also, we notice that the strictly decreasing behaviour of the Lyapunov cannot be achieved for every time instant due to the presence of noise, as it can be expected.

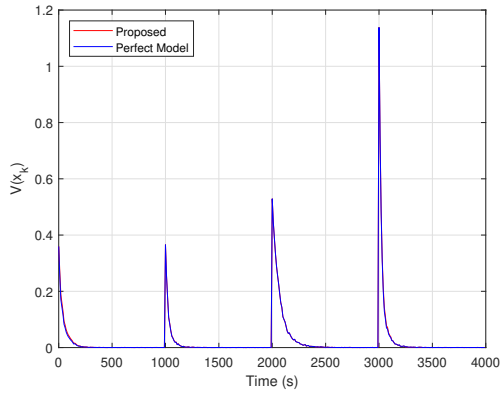

Fig. 5. Evolution of the Lyapunov function for the closed loop simulation.

V. Conclusions

In this paper, we presented a data-driven predictor based on past data and kernel functions in order to forecast nonlinear systems. By means of a set of examples involving the quadruple tank process nonlinear system, it was shown that the prediction capabilities of the proposed approach are satisfactory with and without measurement noise. Also, it was shown that the proposed approach can be applied in the context of control easily, achieving good results. As future work, we consider a detailed discussion about the asymptotic stability and recursive feasibility of the proposed data-driven CLF controller.

Acknowledgments

This work is supported by the Ministry of the Environment, Government of Japan, and JSPS KAKENHI under Grant 21J13956. D. R. Ramirez and T. Alamo also gratefully acknowledge the support from grant PID2022-141159OB-I00, funded by MCIN/AEI/ 10.13039/501100011033 and by ERDF A way of making Europe.

References

[1] C. De Persis and P. Tesi, "Learning controllers for nonlinear systems from data," *Annual Reviews in Control*, p. 100915, 2023.

[2] M. Tanaskovic, L. Fagiano, C. Novara, and M. Morari, "Data-driven control of nonlinear systems: An on-line direct approach," *Automatica*, vol. 75, pp. 1–10, 2017.

[3] T. Martin, T. B. Schön, and F. Allgöwer, "Guarantees for data-driven control of nonlinear systems using semidefinite programming: A survey," *Annual Reviews in Control*, p. 100911, 2023.

[4] J. Berberich, J. Köhler, M. A. Müller, and F. Allgöwer, "Data-driven model predictive control with stability and robustness guarantees," *IEEE Transactions on Automatic Control*, vol. 66, no. 4, pp. 1702–1717, 2020.

[5] ——, "Data-driven model predictive control: closed-loop guarantees and experimental results," *at-Automatisierungstechnik*, vol. 69, no. 7, pp. 608–618, 2021.

[6] T. M. Maupong and P. Rapisarda, "Data-driven control: A behavioral approach," *Systems & Control Letters*, vol. 101, pp. 37–43, 2017.

[7] I. Markovsky and F. Dörfler, "Behavioral systems theory in data-driven analysis, signal processing, and control," *Annual Reviews in Control*, vol. 52, pp. 42–64, 2021.

[8] B. Kiumarsi, K. G. Vamvoudakis, H. Modares, and F. L. Lewis, "Optimal and autonomous control using reinforcement learning: A survey," *IEEE transactions on neural networks and learning systems*, vol. 29, no. 6, pp. 2042–2062, 2017.

[9] Z. Wang and T. Hong, "Reinforcement learning for building controls: The opportunities and challenges," *Applied Energy*, vol. 269, p. 115036, 2020.

[10] S. Chen and S. A. Billings, "Representations of non-linear systems: the NARMAX model," *International journal of control*, vol. 49, no. 3, pp. 1013–1032, 1989.

[11] T. Takagi and M. Sugeno, "Fuzzy identification of systems and its applications to modeling and control," *IEEE transactions on systems, man, and cybernetics*, no. 1, pp. 116–132, 1985.

[12] S. Eleftheriadis, T. Nicholson, M. Deisenroth, and J. Hensman, "Identification of Gaussian process state space models," *Advances in neural information processing systems*, vol. 30, 2017.

[13] D. Gedon, N. Wahlström, T. B. Schön, and L. Ljung, "Deep state space models for nonlinear system identification," *IFAC-PapersOnLine*, vol. 54, no. 7, pp. 481–486, 2021.

[14] A. Mauroy, Y. Susuki, and I. Mezić, *Koopman operator in systems and control*. Springer, 2020.

[15] K. Nakajima and I. Fischer, *Reservoir Computing*. Springer, 2021.

[16] A. D. Carnerero, D. R. Ramirez, and T. Alamo, "Probabilistic interval predictor based on dissimilarity functions," *IEEE Transactions on Automatic Control*, vol. 67, no. 12, pp. 6842–6849, 2021.

[17] A. D. Carnerero, D. R. Ramirez, S. Lucia, and T. Alamo, "Prediction regions based on dissimilarity functions," *ISA transactions*, 2023.

[18] A. D. Carnerero, D. R. Ramirez, D. Limon, and T. Alamo, "Kernel-based state-space kriging for predictive control," *IEEE/CAA Journal of Automatica Sinica*, vol. 10, no. 5, pp. 1263–1275, 2023.

[19] J. Roll, A. Nazin, and L. Ljung, "Nonlinear system identification via direct weight optimization," *Automatica*, vol. 41, no. 3, pp. 475–490, 2005.

[20] S. T. Wierzchoń and M. Kłopotek, *Modern algorithms of cluster analysis*. Springer, 2018.

[21] E. Camacho and C. Bordons, *Model Predictive Control*, 2nd ed. Springer Verlag, 1999.

[22] E. D. Sontag, "A Lyapunov-like characterization of asymptotic controllability," *SIAM journal on control and optimization*, vol. 21, no. 3, pp. 462–471, 1983.

[23] J. Umlauft, L. Pöhler, and S. Hirche, "An uncertainty-based control lyapunov approach for control-affine systems modeled by gaussian process," *IEEE Control Systems Letters*, vol. 2, no. 3, pp. 483–488, 2018.

[24] Z. Wang, G. O. Berger, and R. M. Jungers, "Data-driven control of unknown switched linear systems using scenario optimization," *IEEE Transactions on Automatic Control*, 2024.

[25] I. J. Leontaritis and S. A. Billings, "Input-output parametric models for non-linear systems part I: deterministic non-linear systems," *International journal of control*, vol. 41, no. 2, pp. 303–328, 1985.

[26] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.

[27] A. Beck, *First-order methods in optimization*. SIAM, 2017.

[28] J. Shawe-Taylor and N. Cristianini, *Kernel methods for pattern analysis*. Cambridge University Press, 2004.

[29] T. Hofmann, B. Schölkopf, and A. J. Smola, "Kernel methods in machine learning," *The Annals of Statistics*, vol. 36, no. 3, pp. 1171–1220, 2008.

[30] H. Khalil, "Nonlinear systems," *3rd edition*, 2002.

[31] A. D. Ames, S. Coogan, M. Egerstedt, G. Notomista, K. Sreenath, and P. Tabuada, "Control barrier functions: Theory and applications," in *2019 18th European control conference (ECC)*. IEEE, 2019, pp. 3420–3431.

[32] H. J. van Waarde, M. K. Camlibel, and M. Mesbahi, "From noisy data to feedback controllers: Nonconservative design via a matrix s-lemma," *IEEE Transactions on Automatic Control*, vol. 67, no. 1, pp. 162–175, 2020.

[33] K. H. Johansson, "The quadruple-tank process: A multivariable laboratory process with an adjustable zero," *IEEE Transactions on control systems technology*, vol. 8, no. 3, pp. 456–465, 2000.