

MIT Open Access Articles

*Online Support Vector Regression Based Adaptive
NARMA-L2 Controller for Nonlinear Systems*

The MIT Faculty has made this article openly available. **Please share**
how this access benefits you. Your story matters.

As Published: <https://doi.org/10.1007/s11063-020-10403-8>

Publisher: Springer US

Persistent URL: <https://hdl.handle.net/1721.1/132045>

Version: Author's final manuscript: final author's manuscript post peer review, without publisher's formatting or copy editing

Terms of Use: Article is made available in accordance with the publisher's policy and may be subject to US copyright law. Please refer to the publisher's site for terms of use.



Online Support Vector Regression Based Adaptive NARMA-L2 Controller for Nonlinear Systems

Cite this article as: Kemal Uçak and Gülay Öke Günel, Online Support Vector Regression Based Adaptive NARMA-L2 Controller for Nonlinear Systems, Neural Processing Letters <https://doi.org/10.1007/s11063-020-10403-8>

This Author Accepted Manuscript is a PDF file of an unedited peer-reviewed manuscript that has been accepted for publication but has not been copyedited or corrected. The official version of record that is published in the journal is kept up to date and so may therefore differ from this version.

Terms of use and reuse: academic research for non-commercial purposes, see here for full terms. <https://www.springer.com/aam-terms-v1>

Author accepted manuscript

Noname manuscript No.
(will be inserted by the editor)

Online Support Vector Regression based Adaptive NARMA-L2 Controller for Nonlinear Systems

Kemal UÇAK^{1,2} · Gülay ÖKE GÜNEL³

Received: / Accepted:

Abstract NARMA model is a simple and effective way to represent nonlinear systems, based on the NARMA model, NARMA-L2 controller is designed and has been successfully applied in the literature. Success of NARMA-L2 controller is directly related to the precision with which controlled systems' dynamics can be estimated. In this paper, online SVR is utilized to obtain controlled plant's subdynamics and consequently this information is used in the construction of NARMA-L2 controller. Hence functionality of NARMA-L2 controllers and high generalization capability of SVR are combined. Also, SVR formulates a convex optimization problem and therefore guarantees global optimum solution. The proposed method is assessed by performing simulations on a nonlinear CSTR system, the robustness of the designed controller is also tested under noisy and uncertainty conditions.

Keywords Model based adaptive control · NARMA-L2 SVR model · NARMA-L2 SVR controller · Online support vector regression · System identification

1 Introduction

In model based control design, the most crucial point is the accurate determination of the controlled system model. Model type which will be used in the control algorithm must have adequate complexity to capture the essential features of the actual system, yet it should not complicate the design with too much computational load. Appropriate selection of the system model leads to successful control

Kemal UÇAK (✉)

¹Massachusetts Institute of Technology Cambridge, MA 02139

¹E-mail: ucakk@mit.edu

²Muğla Sıtkı Koçman University Kötekli 48000, Muğla, Turkey

²E-mail: ucak@mu.edu.tr

Gülay ÖKE GÜNEL

³Istanbul Technical University Maslak 34469, İstanbul, Turkey

³E-mail: gulay.oke@itu.edu.tr

and good performance. NARMA-L2 model proposed by Narendra and Mukhopadhyay [1] is a simple but effective way to represent nonlinear systems. It is based on expressing the dynamical equations of the nonlinear system in Taylor series expansion form so that the control input is separated from the nonlinear dynamics and appears linearly [2, 3, 4, 5]. Once an accurate model is obtained, it is straightforward to compute the control input using NARMA-L2 controller methodology.

In the technical literature, there are various studies performed on NARMA-L2 controllers. In [2], Majstorovic et al. derived the NARMA model for a two tank system offline and designed a controller based on this model. In [3], Pedro et al. controlled slip in an anti-lock braking system using NARMA-L2 method. In [6], Hagan, Demuth and De Jesus applied NARMA-L2 controller, model predictive control and model reference control, all based on neural networks on a continuously stirred tank reactor (CSTR), a magnetic levitation system and a single-link robot arm and compared the results. A weakness of NARMA-L2 based control is that it produces an oscillatory output. In order to suppress the oscillation in [5], Pukrittayakamee, De Jesus and Hagan proposed to use an additional term consisting of the output of the system and a scaled version of the reference signal. Also in [7], Wahyudi, Mokri and Shafie, controlled a permanent magnet brushed dc motor driven single-link robot arm and smoothed NARMA L2 to alleviate oscillations. In [8], Akbarimajd and Kia controlled a 2-dof underactuated planar robot with a NARMA-L2 controller. In [9], Vesselenyi et al. used NARMA-L2 controller for the position control of a pneumatic actuator. In [10], Awwad, Abu-Rub and Toliyat employed a NARMA-L2 based adaptive controller for the tracking control of rotor speed for an induction motor. In [11], Pedro and Ekoru used NARMA-L2 control method for a nonlinear four-degrees of freedom servo-hydraulic vehicle suspension system and carried out a comparison with a passive suspension system. In [12], Lutfy and Salamat controlled nonlinear systems using internal model control (IMC) methodology where they obtained the models with NARMA-L2 technique using wavelet neural networks. In order to avoid trapping in local minima and to speed up convergence, they employed artificial immune system technique to obtain optimal neural network weights. In [13], Paul and Chokkadi controlled a shell and tube heat exchanger (STHE) temperature process system with NARMA-L2 controller and they trained it using Bayesian regularization algorithm. In [14], Al-Dunainawi, Abbod, and Jizany implemented an ANFIS based NARMA-L2 controller and tested its performance on a binary distillation column. In order to optimize ANFIS parameters they employed particle swarm optimisation (PSO) method. They also compared their results with those obtained by PID, with PD, PI and PID-like fuzzy logic controller tuned by PSO as well as GA, and also with ANN based NARMA-L2. In [15], Yang et al proposed multiple NARMA-L2 approach to identify and control nonlinear systems and evaluated their method on several benchmark problems, also they performed experiments on a modified DC motor.

The success of NARMA-L2 controller directly depends on correct estimation of system model. In most of the applications of NARMA-L2 controller, identification of the model has been carried out with backpropagation based techniques such as neural networks (NN) and adaptive neuro-fuzzy inference systems (ANFIS). However, backpropagation algorithm which depends on gradient descent method has the risk of getting trapped at local minimum. In support vector regression (SVR) method, a convex objective function is formulated, so the global optimum solution

is assured. Hence, SVR has been effectively applied to control and identify dynamical systems due to its generalization property and unique global solution [16]. In [17], Uçak and Günel used SVR technique to obtain NARX models of nonlinear systems in an online manner and then proposed a method to convert from SVR based NARX model to NARMA-L2 submodels, they assessed the performance on a bioreactor system. In [18], Şen and Öke Günel extended this method for MIMO systems and used online least square support vector regression (LSSVR). Apart from these examples, to abstain from local minima, global optimization techniques have been employed to find optimal parameters of neural networks in a few recent applications [12, 14].

There exist numerous applications of SVR in control architectures in the technical literature. In [19], Wanfeng, Shengdun and Yajing employed an adaptive PID controller with weights tuned using gradient descent, they obtained the Jacobian with online LSSVR. Usage of online LSSVR involves the selection of a sliding window, to improve training performance. In [20], Zhao, Li, and Wang proposed to change the sliding window size. In [21], Yuan, Wang, and Wu suggested using a controller with both feedforward and feedback parts. In the feedback part, they employed a conventional PID controller while in the feedforward path, information about system Jacobian obtained by SVR was used. In [22], İplikçi proposed a PID controller employing SVR. In this work SVR is used in estimating the model of the system as well as the Jacobian. PID parameters are adapted using Levenberg Marquardt method by predicting K-step ahead system Jacobian, also a correction term is formulated using the Taylor expansion of the control input. In [23], Takao, Yamamoto and Hinamoto employed support vector classifier (SVC) in a decision tree architecture. The operating region is divided into small regions, for each region a separate robust PID controller and a SVC to determine whether the operating condition belongs to that region is designed. All SVC's are combined to give a final decision about the operation region and the corresponding PID is utilized. Differently from the previously mentioned publications, in [24], Liu, Yi and Zhao used SVR to estimate the inverse dynamics of a nonlinear plant and based on this model they designed an inverse controller. Similarly, in [25] Wang, Pi and Sun designed an inverse controller for a nonlinear system by utilizing an online SVR. In [26], Yuan, Wang and Wu proposed a control method where they employed separate SVR's as the forward system model and the inverse controller. They optimized system model and controller parameters by backpropagation algorithm. In [27, 28, 29, 30] nonlinear internal model control (NIMC) approach has been combined with the powerful generalization and modeling capacity of SVR. In [31, 32, 33, 34, 35] and [36] SVR methodology is utilized in model predictive control, in these studies mainly the models of the controlled systems have been obtained by SVR.

In [37], Uçak and Günel have utilized online SVR directly as a controller for the first time. They defined the “closed-loop margin” concept based on tracking error and they optimized this margin to minimize the tracking error. They employed a second SVR structure to estimate the controlled system dynamics. In [38], they used the closed-loop margin concept to optimize the parameters of self-tuning regulators, in particular they have implemented an adaptive PID and adaptive fuzzy PID with this methodology to control a nonlinear bioreactor benchmark system. In [39], again the closed-loop margin concept of [37] was employed to propose a model free control architecture which did not need the computation of

a model estimate. In [40], the same online SVR based methodology was adopted to update the parameters of a sliding mode controller.

In this paper, a new method is proposed where online SVR is directly employed to attain NARMA-L2 model of nonlinear systems, consequently this obtained model is adopted to construct a NARMA-L2 controller and compute the control input signal. Here, main novelty is the direct identification of NARMA-L2 subdynamics by utilizing the online SVR. In the literature generally gradient based techniques such as ANN or ANFIS have frequently been used to estimate the NARMA model of nonlinear systems. Nonetheless, the main drawback of gradient based methods is their convergence to local minima, this leads to only locally valid models. In contrast, SVR defines a convex cost function, minimizing it leads to the global optimum solution. **The main difficulty in employing SVR method in NARMA-L2 modeling is to formulate how to utilize the input-output data in SVR to estimate the subdynamic terms separately. To the best of the authors' knowledge, the only application of a SVR based approach to obtain NARMA-L2 model was proposed in [17] however it was an indirect approach where NARX model was obtained first, and NARMA model was calculated through defining certain relations between NARX and NARMA SVR parameters. The main contribution of our proposed approach here is the direct identification of SVR parameters without appealing to any intermediate formulation. This leads to the straightforward estimation of the subdynamical terms of the NARMA-L2 model which are consequently used to compute the control input.** Proposed method is evaluated with simulations performed on the benchmark CSTR chemical reaction system. Robustness is also evaluated by repeating the simulations with noise and parametric uncertainty. **Additionally, a detailed comparison with the previously published work in [17] is also accomplished and the superiority of the proposed adaptive NARMA-L2 controller is verified.**

Organization of the paper can be summarized as: In Section 2, proposed NARMA-L2 controller is presented together with a brief review of online SVR method. Adaptation rules for the SVR based NARMA-L2 controller are also given. In Section 3, the CSTR system is described, **simulation results are given in detail, also a detailed comparison is performed with a SVR based NARMA-L2 controller from the literature.** Finally, Section 4 consists of a brief conclusion.

2 Proposed SVR based NARMA-L2 Controller

2.1 Review of Online Support Vector Regression

Support vector regression is a prominent data sampled regression technique in machine learning. A significant feature of SVR is that it ensures global minimum of the optimization problem by means of converting the **primal form of the problem into a convex one in dual form.** Assume there is a data set for the SVR model given by:

$$\mathbf{T} = \{\mathbf{X}_i, y_{i+1}\}_{i=1}^N \quad \mathbf{X}_i \in \mathbf{X} \subseteq R^n, y_{i+1} \in R \quad (1)$$

In (1), n and N represent input dimension and number of training samples respectively, \mathbf{X}_i shows input vector, y_{i+1} is the desired outcome or target output. Mapping between the input and output for the data given in (1) can be estimated

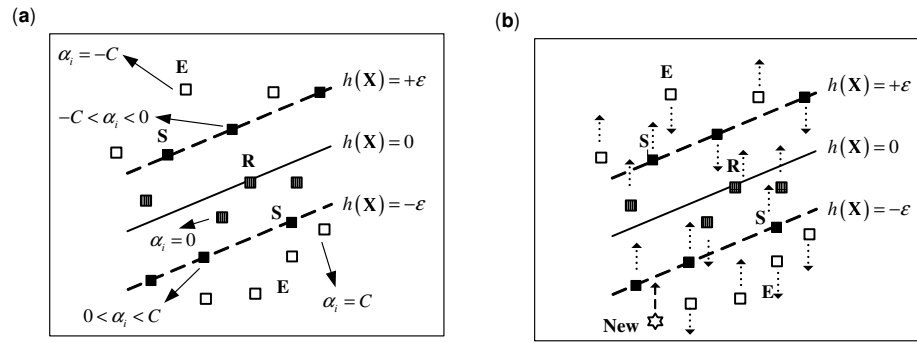


Fig. 1 (a) **E, S, R** subsets (b) Migration of samples among subsets **E, S** and **R**[40].

using the following dual regression function:

$$y_{i+1} = f(\mathbf{X}_i) = \sum_{k \in SV} \alpha_k K(\mathbf{X}_k, \mathbf{X}_i) + b, \quad \alpha_k = \eta_k - \eta_k^* \quad (2)$$

In (2), η_k and η_k^* stand for the nonnegative Lagrange multipliers for the data on either side of the separating hyperplane used in SVR formulations, they also represent the support vectors (SV), b is a bias term. α_k, b are regression parameters to be obtained. The dual optimization problem with convex objective function is formulated as:

$$J_D = \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N (\eta_i - \eta_i^*)(\eta_j - \eta_j^*) K_{ij} + \varepsilon \sum_{i=1}^N (\eta_i + \eta_i^*) - \sum_{i=1}^N y_i (\eta_i - \eta_i^*) \quad (3)$$

with the constraints given in (4):

$$0 \leq \eta_i \leq C, \quad 0 \leq \eta_i^* \leq C, \quad \sum_{i=1}^N (\eta_i - \eta_i^*) = 0, \quad i = 1, 2, \dots, N \quad (4)$$

Here, $K_{ij} = \Phi(\mathbf{x}_i)^T \Phi(\mathbf{x}_j)$, where $\Phi(\mathbf{x}_j)$ is the kernel function that maps the input vector onto the feature space, ε is training error tolerance parameter initialized at the begining of the optimization procedure and C determines the size of the search space for Lagrange multipliers which also constrains the maximum and minimum values of the Lagrange values. C also acts as a penalty term for the training examples outside the ε -tube. This formulation is more favorable for offline problems. As a result of convergence to optimal point, the training samples are classified into three main subsets (**E, S, R**) with respect to their Lagrange and error margin values as illustrated in Figure 1 a. If new data is introduced as depicted in Figure 1 (b), in order to learn the dynamics of the incoming data, it is required to assign a new Lagrange parameter corresponding to the new sample and update all regression parameters including the most recent training data. Depending on inclusion of new sample, the structure of the regression surface changes. As a result, the samples may change their subsets and immigration may occur among training samples. Therefore, the update rules are derived by taking into account all possible migrations. The details are given in [17, 37, 41, 42, 43, 44].

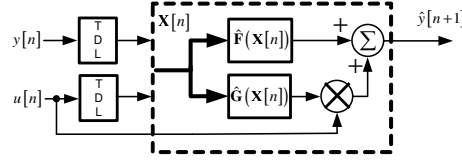


Fig. 2 NARMA-L2 system model[17].

2.2 An Overview of NARMA-L2 Model and Controller

This section briefly reviews NARMA-L2 system model and derivation of control law for nonlinear systems via NARMA-L2 model. The generic representation of a nonlinear system is in the following form:

$$y[n+d] = \mathbf{F}_{actual}(u[n], \dots, u[n-k+1], y[n], \dots, y[n-k+1]) \quad (5)$$

where $y[n]$ is system output, $u[n]$ symbolizes control input and d denotes relative degree. The model given in (5) is not practical to use in control applications since it is not easy to extract control input $u[n]$ from it. NARMA-L2 model is obtained from (5) by expanding it in the form of Taylor series expansion and retaining first order terms. Nonlinear system dynamics can be identified using NARMA-L2 model as:

$$y[n+d] = f(x[n]) + g(x[n])u[n] \quad (6)$$

where $f(\cdot)$ and $g(\cdot)$ are nonlinear functions and $x[\cdot]$ represents system state. The main advantage of NARMA-L2 model is that the control term $u[n]$ appears linearly and can be easily solved for, as can be seen from equation (6). Hence it can be practically used in control design applications. If the system dynamics is not known ahead, it is possible to approximate it with:

$$y[n+d] = \hat{\mathbf{F}}(\mathbf{X}[n]) + \hat{\mathbf{G}}(\mathbf{X}[n])u[n] \quad (7)$$

Here, $\hat{\mathbf{F}}(\cdot)$ and $\hat{\mathbf{G}}(\cdot)$ represent nonlinear system dynamics which will be estimated, $\mathbf{X}[n] = \begin{bmatrix} u[n-1] \cdots u[n-n_u], y[n] \cdots y[n-n_y+1] \end{bmatrix}^T$ is input vector. n_u, n_y represent the number of previous control inputs and system outputs used in system model. $\hat{\mathbf{F}}(\cdot)$ and $\hat{\mathbf{G}}(\cdot)$ submodels can be obtained by using numerical or intelligent techniques [17]. Figure 2 illustrates NARMA-L2 system model. In Figures 2-4 the blocks labeled as “TDL” represent a delay of one sampling time, in signal processing terminology, z^{-1} . “TDL” stands for Tap Delay Line which represents a discrete element that allows a signal to be delayed by a number of samples. In system identification, model output has to correctly estimate the actual output of the system ($\hat{y}[n+1] = y[n+1]$). Therefore, the NARMA-L2 model parameters are optimized to decrease modeling error ($\hat{e}[n+1] = \hat{y}[n+1] - y[n+1]$). In control design, controller dynamics minimizes the tracking error ($\hat{e}_{tr}[n+1] = r[n+1] - y[n+1]$), hence system dynamics are enforced to pursue the reference input. Therefore, by replacing output of NARMA-L2 model ($\hat{y}[n+1]$), by the reference input signal

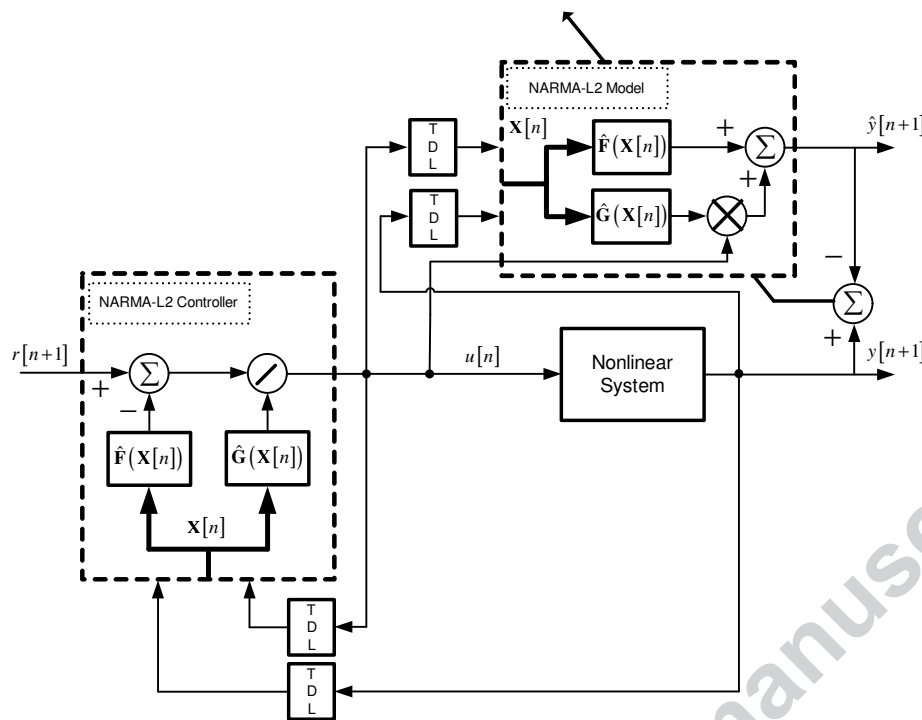


Fig. 3 Schematic representation of NARMA-L2 controller[17].

$(r[n+1])$ the control law can be easily acquired as follows for $d = 1$:

$$\begin{aligned} r[n+1] &\cong \hat{y}[n+1] = \hat{F}(\mathbf{X}[n]) + \hat{G}(\mathbf{X}[n])u[n] \\ u[n] &\cong \frac{r[n+1] - \hat{F}(\mathbf{X}[n])}{\hat{G}(\mathbf{X}[n])} \end{aligned} \quad (8)$$

Thus, schematic representation of the NARMA-L2 controller derived as described above is illustrated in Figure 3. As can be seen from (8), NARMA-L2 controller is very easy to derive, contains no mathematical complications and control input signal $u[n]$ is easy to compute. These are the main advantages of NARMA-L2 controller design. It must be noted that the success of the controller lies in the correct estimation of the dynamics of the controlled system. During the course of control, as the number of iterations increase, the adaptation mechanism will tune the estimated subdynamics terms $\hat{F}(\cdot)$ and $\hat{G}(\cdot)$ so that they converge to actual terms $f(\cdot)$ and $g(\cdot)$, which will lead to the convergence of the modeling output to actual system output ($\hat{y}[n+1] \cong y[n+1]$). Eventually, the modeling error $\hat{e} = \hat{y}[n+1] - y[n+1]$ will converge to zero, which means that the obtained model behaves the same as the actual system. As a result, accurate identification of subdynamics $\hat{F}(\cdot)$ and $\hat{G}(\cdot)$ automatically leads to the effective design of NARMA-L2 controller. Thus, it is possible to interpret the control law for NARMA-L2 controller as an adaptive controller with two adjustable parameters($\hat{F}(\cdot)$ and $\hat{G}(\cdot)$)

as follows:

$$u[n] = f_{\text{law}}(r[n+1], \hat{\mathbf{F}}(\mathbf{X}[n]), \hat{\mathbf{G}}(\mathbf{X}[n])) \cong \frac{r[n+1] - \hat{\mathbf{F}}(\mathbf{X}[n])}{\hat{\mathbf{G}}(\mathbf{X}[n])} \quad (9)$$

According to (9), accurate estimation of subdynamics $\hat{\mathbf{F}}(\cdot)$ and $\hat{\mathbf{G}}(\cdot)$ plays a critical role in obtaining good tracking performance. By considering the control law as an adaptive controller, in this paper online SVR has been deployed to estimate the tunable controller parameters.

2.3 SVR based adaptive NARMA-L2 controller

In NARMA-L2 controller design, control input $u[n]$ is computed by (8) as given in Section 2.2. As discussed above, the main advantage of NARMA-L2 is its mathematical simplicity and ease of implementation. However, the key issue in design is the correct estimation of the system dynamics. In this study, the subdynamical terms $\hat{\mathbf{F}}(\cdot)$ and $\hat{\mathbf{G}}(\cdot)$ have been estimated using the online SVR methodology. An overview of the proposed adaptive NARMA-L2 controller structure is illustrated in Figure 4. As can be seen in Figure 4, the control architecture consists of three main blocks: NARMA-L2 model, adaptive NARMA-L2 controller and the actual system to be controlled. NARMA-L2 model block contains the online SVR based models of the subdynamical terms $\hat{\mathbf{F}}(\cdot)$ and $\hat{\mathbf{G}}(\cdot)$, the model with the tuned parameters computes an estimate of the model output $\hat{y}[n+1]$, this is compared with the actual output of the system to be controlled, $y[n+1]$. The model output is subtracted from the actual output to obtain the modeling error: $\hat{e} = \hat{y}[n+1] - y[n+1]$. Modeling error $\hat{e}[n+1]$ is utilized to tune the parameters of the SVR models for $\hat{\mathbf{F}}(\cdot)$ and $\hat{\mathbf{G}}(\cdot)$. Adaptive NARMA-L2 controller block takes as inputs the estimated subdynamical terms $\hat{\mathbf{F}}(\cdot)$ and $\hat{\mathbf{G}}(\cdot)$, and the reference input signal $r[n+1]$. These are substituted into (9), and the control input signal $u[n]$ is computed, which is consequently fed into the actual system to be controlled. The subdynamics of NARMA-L2 model, $\hat{\mathbf{F}}(\cdot)$ and $\hat{\mathbf{G}}(\cdot)$ can be estimated via SVR as follows:

$$\begin{aligned} \hat{\mathbf{F}}(\boldsymbol{\lambda}[n], \mathbf{X}[n]) &= \sum_{i=1}^N \lambda_i[n] K(\mathbf{X}_{\text{SV}_i}, \mathbf{X}[n]) + b_F \\ \hat{\mathbf{G}}(\boldsymbol{\beta}[n], \mathbf{X}[n]) &= \sum_{j=1}^N \beta_j[n] K(\mathbf{X}_{\text{SV}_j}, \mathbf{X}[n]) + b_G \end{aligned} \quad (10)$$

In (10) $\boldsymbol{\lambda}[n]$ and $\boldsymbol{\beta}[n]$ are model regressor parameters and $\mathbf{X}[n]$ represents input vector. Thus, NARMA-L2 model output is computed with:

$$\hat{y}[n+1] = \hat{\mathbf{F}}(\boldsymbol{\lambda}[n], \mathbf{X}[n]) + \hat{\mathbf{G}}(\boldsymbol{\beta}[n], \mathbf{X}[n])u[n] \quad (11)$$

By using online SVR, the subdynamics $\hat{\mathbf{F}}(\boldsymbol{\lambda}[n], \mathbf{X}[n])$ and $\hat{\mathbf{G}}(\boldsymbol{\beta}[n], \mathbf{X}[n])$ can be adjusted depending on alternation of the system dynamics, which enables to introduce adaptation ability to the NARMA-L2 controller in response to internal and external factors which affect system response.

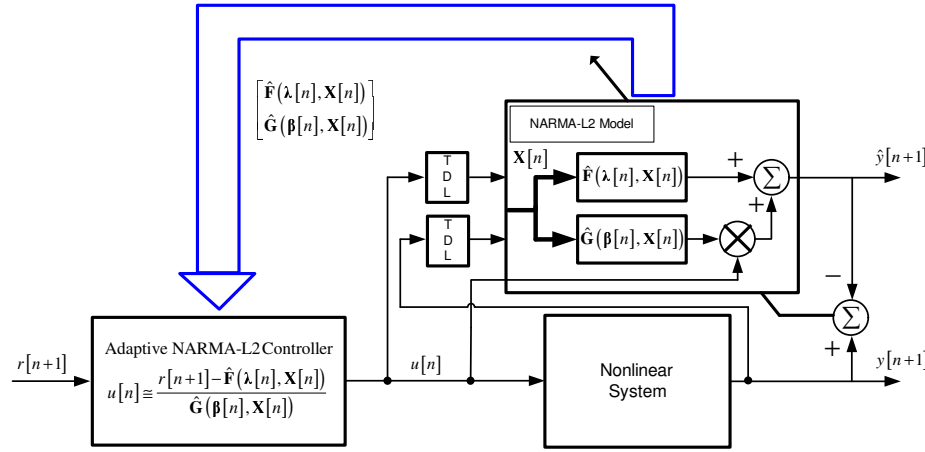


Fig. 4 Adaptive NARMA-L2 controller based on SVR.

2.4 Adaptation Rules of SVR for Adaptive NARMA-L2 controller

To formulate the adaptation rules for the online SVR, let us define open-loop error margin function of calculated for the i th input \mathbf{X}_i :

$$h(\mathbf{X}_i) = \hat{y}[i+1] - y[i+1] = f_{\text{model}}(\mathbf{X}_i) - y[i+1] \quad (12)$$

where

$$\begin{aligned} \hat{y}[i+1] &= \hat{\mathbf{F}}(\boldsymbol{\lambda}, \mathbf{X}_i) + \hat{\mathbf{G}}(\boldsymbol{\beta}, \mathbf{X}_i)u[i] \\ \hat{\mathbf{F}}(\boldsymbol{\lambda}, \mathbf{X}_i) &= \sum_{k=1}^N \lambda_k K(\mathbf{X}_{\text{SV}_k}, \mathbf{X}_i) + b_F \\ \hat{\mathbf{G}}(\boldsymbol{\beta}, \mathbf{X}_i) &= \sum_{z=1}^N \beta_z K(\mathbf{X}_{\text{SV}_z}, \mathbf{X}_i) + b_G \\ \mathbf{X}_i &= [u[i-1] \cdots u[i-n_u], y[i] \cdots y[i-n_y+1]]^T \end{aligned} \quad (13)$$

The data is going to converge and migrate according to the following conditions:

$$\begin{aligned} h(\mathbf{X}_i) &\geq \varepsilon_\lambda, \lambda_i = -C_\lambda \\ h(\mathbf{X}_i) &= \varepsilon_\lambda, -C_\lambda < \lambda_i < 0 \\ -\varepsilon_\lambda &\leq h(\mathbf{X}_i) \leq \varepsilon_\lambda, \lambda_i = 0 \\ h(\mathbf{X}_i) &= -\varepsilon_\lambda, 0 < \lambda_i < C_\lambda \\ h(\mathbf{X}_i) &\leq -\varepsilon_\lambda, \lambda_i = C_\lambda \\ h(\mathbf{X}_i) &\geq \varepsilon_\beta, \beta_i = -C_\beta \\ h(\mathbf{X}_i) &= \varepsilon_\beta, -C_\beta < \beta_i < 0 \\ -\varepsilon_\beta &\leq h(\mathbf{X}_i) \leq \varepsilon_\beta, \beta_i = 0 \\ h(\mathbf{X}_i) &= -\varepsilon_\beta, 0 < \beta_i < C_\beta \\ h(\mathbf{X}_i) &\leq -\varepsilon_\beta, \beta_i = C_\beta \end{aligned} \quad (14)$$

Assume a new input vector $\mathbf{X}[n]$ is incorporated to the training data set. Using existing model parameters, error margin function will be as follows:

$$h^{old}(\mathbf{X}[n]) = \hat{\mathbf{F}}(\boldsymbol{\lambda}^{old}[n], \mathbf{X}[n]) + \hat{\mathbf{G}}(\boldsymbol{\beta}^{old}[n], \mathbf{X}[n])u[n] - y[n+1] \quad (15)$$

Dynamics of the new added data has to be learnt and to achieve this, SVR parameters must be adjusted, and new values of model parameters must be computed. Therefore, the new error margin can be defined as:

$$h^{new}(\mathbf{X}[n]) = \hat{\mathbf{F}}(\boldsymbol{\lambda}^{new}[n], \mathbf{X}[n]) + \hat{\mathbf{G}}(\boldsymbol{\beta}^{new}[n], \mathbf{X}[n])u[n] - y[n+1] \quad (16)$$

If an input vector belongs to either \mathbf{E} or \mathbf{R} subsets, value of Lagrange multiplier (λ_c, β_c) equals to either "0" or "C". In case support vectors have to stay in \mathbf{S} , then $\Delta h(\mathbf{X}_i)$, $i \in \mathbf{S}$ [45]. Thus, $\Delta h(\mathbf{X}_i)$ can be calculated as

$$\begin{aligned} \Delta h(\mathbf{X}_i) &= h^{new}(\mathbf{X}_i) - h^{old}(\mathbf{X}_i) = \left(\mathbf{F}(\boldsymbol{\lambda}^{new}, \mathbf{X}_i) + \mathbf{G}(\boldsymbol{\beta}^{new}, \mathbf{X}_i)u[n] - y[n+1] \right) \\ &\quad - \left(\mathbf{F}(\boldsymbol{\lambda}^{new}, \mathbf{X}_i) + \mathbf{G}(\boldsymbol{\beta}^{new}, \mathbf{X}_i)u[n] - y[n+1] \right) \\ \Delta h(\mathbf{X}_i) &= \left(\mathbf{F}(\boldsymbol{\lambda}^{new}, \mathbf{X}_i) - \mathbf{F}(\boldsymbol{\lambda}^{old}, \mathbf{X}_i) \right) + \left(\mathbf{G}(\boldsymbol{\beta}^{new}, \mathbf{X}_i) - \mathbf{G}(\boldsymbol{\beta}^{old}, \mathbf{X}_i) \right)u[n] \\ \Delta h(\mathbf{X}_i) &= \sum_{k=1}^N (\lambda_k^{new} - \lambda_k^{old})K(\mathbf{X}_{SV_k}, \mathbf{X}_i) + (b_F^{new} - b_F^{old}) \\ &\quad + \left(\sum_{z=1}^N (\beta_z^{new} - \beta_z^{old})K(\mathbf{X}_{SV_z}, \mathbf{X}_i) + (b_G^{new} - b_G^{old}) \right)u[n] \end{aligned} \quad (17)$$

By substituting the following identities in (17)

$$\begin{aligned} \Delta h(\mathbf{X}_i) &= h^{new}(\mathbf{X}_i) - h^{old}(\mathbf{X}_i) \\ \Delta \lambda_k &= \lambda_k^{new} - \lambda_k^{old} \\ \Delta b_F &= b_F^{new} - b_F^{old} \\ \Delta \beta_k &= \beta_z^{new} - \beta_z^{old} \\ \Delta b_G &= b_G^{new} - b_G^{old} \end{aligned} \quad (18)$$

$\Delta h(\mathbf{X}_i)$ can be acquired as

$$\Delta h(\mathbf{X}_i) = \sum_{k=1}^N \Delta \lambda_k K(\mathbf{X}_{SV_k}, \mathbf{X}_i) + \Delta b_F + \left(\sum_{z=1}^N \Delta \beta_z K(\mathbf{X}_{SV_z}, \mathbf{X}_i) + \Delta b_G \right)u[n] \quad (19)$$

For the condition $\Delta h(\mathbf{X}_i)$, $i \in \mathbf{S}$ to be satisfied, each summation term must be zero as follows:

$$\begin{aligned} \Delta h(\mathbf{X}_i) &= \Delta h_1(\mathbf{X}_i) + \Delta h_2(\mathbf{X}_i)u[n] \\ &= \sum_{k=1}^N \Delta \lambda_k K(\mathbf{X}_{SV_k}, \mathbf{X}_i) + \Delta b_F + \left(\sum_{z=1}^N \Delta \beta_z K(\mathbf{X}_{SV_z}, \mathbf{X}_i) + \Delta b_G \right) u[n] = 0 \\ \text{for } u[n] \neq 0 &\begin{cases} \Delta h_1(\mathbf{X}_i) = \sum_{k=1}^N \Delta \lambda_k K(\mathbf{X}_{SV_k}, \mathbf{X}_i) + \Delta b_F = 0 \\ \Delta h_2(\mathbf{X}_i) = \sum_{z=1}^N \Delta \beta_z K(\mathbf{X}_{SV_z}, \mathbf{X}_i) + \Delta b_G = 0 \end{cases} \end{aligned} \quad (20)$$

In case a new input with influences λ_c and β_c is added to the data set and there is no migration between \mathbf{S} , \mathbf{E} and \mathbf{R} subsets, variations in $\sum_{k=1}^N \Delta \lambda_k K(\mathbf{X}_{SV_k}, \mathbf{X}_i) + \Delta b_F$ and $\sum_{z=1}^N \Delta \beta_z K(\mathbf{X}_{SV_z}, \mathbf{X}_i) + \Delta b_G$ are derived as follows:

$$\begin{aligned} \Delta h_1(\mathbf{X}_i) &= K_{ic} \Delta \lambda_c + \sum_{k=1}^N K_{ik} \Delta \lambda_k + \Delta b_F = 0, \quad K_{ik} = K(\mathbf{X}_{SV_k}, \mathbf{X}_i) \\ \Delta h_2(\mathbf{X}_i) &= K_{ic} \Delta \beta_c + \sum_{z=1}^N K_{iz} \Delta \beta_z + \Delta b_G = 0, \quad K_{iz} = K(\mathbf{X}_{SV_z}, \mathbf{X}_i) \end{aligned} \quad (21)$$

From dual constraints, summation of the all Lagrange multipliers must be zero as follows:

$$\begin{aligned} \lambda_c + \sum_{k=1}^N \lambda_k &= 0 \\ \beta_c + \sum_{z=1}^N \beta_z &= 0 \end{aligned} \quad (22)$$

Since the derivation of the rules for both $\{\lambda, b_F\}$ and $\{\beta, b_G\}$ are the same, by assigning a new variable $\{\alpha, b\}$ in place of $\{\lambda, b_F\}$ and $\{\beta, b_G\}$, the valid adaptation rules for both $\{\lambda, b_F\}$ and $\{\beta, b_G\}$ can be derived as given in (23–28). As described in the parts above, for data in \mathbf{S} , $\Delta h(\mathbf{X}_i) = 0, i \in \mathbf{S}$ must be satisfied [45]. So, the variations of Lagrange multipliers of support vectors in \mathbf{S} can be formulated in (23) [44,45]:

$$\begin{aligned} \sum_{k=1}^N K_{ik} \Delta \alpha_k + \Delta b &= -K_{ic} \Delta \alpha_c \\ \sum_{k=1}^N \Delta \alpha_k &= -\Delta \alpha_c \end{aligned} \quad (23)$$

Let us define indices for the data in \mathbf{S} as:

$$\mathbf{S} = \{s_1, s_2, s_3, \dots, s_k\} \quad (24)$$

Then it is possible to rewrite (23) in matrix form:

$$\begin{bmatrix} 0 & 1 & \cdots & 1 \\ 1 & K_{s_1 s_1} & \cdots & K_{s_1 s_k} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & K_{s_k s_1} & \cdots & K_{s_k s_k} \end{bmatrix} \begin{bmatrix} \Delta b \\ \Delta \alpha_{s_1} \\ \vdots \\ \Delta \alpha_{s_k} \end{bmatrix} = - \begin{bmatrix} 1 \\ K_{s_1 c} \\ \vdots \\ K_{s_k c} \end{bmatrix} \Delta \alpha_c \quad (25)$$

So,

$$\begin{bmatrix} \Delta b \\ \Delta \alpha_{s_1} \\ \vdots \\ \Delta \alpha_{s_k} \end{bmatrix} = \Phi \Delta \alpha_c \quad (26)$$

where

$$\Phi = \begin{bmatrix} \Phi \\ \Phi_{s_1} \\ \vdots \\ \Phi_{s_k} \end{bmatrix} = -\Psi \begin{bmatrix} 1 \\ K_{s_1 c} \\ \vdots \\ K_{s_k c} \end{bmatrix}, \quad \Psi = \begin{bmatrix} 0 & 1 & \cdots & 1 \\ 1 & K_{s_1 s_1} & \cdots & K_{s_1 s_k} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & K_{s_k s_1} & \cdots & K_{s_k s_k} \end{bmatrix}^{-1} \quad (27)$$

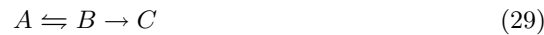
as given in [41]. This means that it is possible to update Lagrange multipliers and bias values for a given $\Delta \alpha_c$, for the data samples in \mathbf{S} using equations (23–27). Also, it is possible to calculate non-support data margins by (24,26–28):

$$\begin{bmatrix} \Delta h(\mathbf{x}_{z_1}) \\ \Delta h(\mathbf{x}_{z_2}) \\ \vdots \\ \Delta h(\mathbf{x}_{z_m}) \end{bmatrix} = \gamma \Delta \alpha_c, \quad \gamma = \begin{bmatrix} K_{z_1 c} \\ K_{z_2 c} \\ \vdots \\ K_{z_m c} \end{bmatrix} + \begin{bmatrix} 1 & K_{z_1 s_1} & \cdots & K_{z_1 s_l} \\ 1 & K_{z_2 s_1} & \cdots & K_{z_2 s_l} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & K_{z_m s_1} & \cdots & K_{z_m s_l} \end{bmatrix} \Phi \quad (28)$$

In (28), $z_1 \cdots z_m$ denote non-support data indices and γ represents margin sensitivities. It must also be noted that for the data in \mathbf{S} , $\gamma = 0$. Using these update rules $h(\mathbf{X}_i)$ and α_i can be computed for a given value of $\Delta \alpha_c$ for support set and non-support set data [41]. Detailed information on the algorithm, as well as the update of Φ matrix is available in [41,42,44,45]. Also, how to find an appropriate $\Delta \alpha_c$ is described in detail in [17,37,38,39,40,41,42,45].

3 Simulation Results

The proposed control architecture has been assessed on a Continuously Stirred Tank Reactor System (CSTR). CSTR is a chemical reactor model used to carry out isothermal, liquid-phase, successive multicomponent chemical reactions [46, 47] where the contents are uniform and well stirred [48]. Also it is known as a vat or backmix reactor [48,49]. Consider the given chemical reaction:



The goal is to control the concentration of reaction output C by changing molar feed rate of reactant B. The differential equations given in (30) describe the dynamical process in the CSTR [46]:

$$\begin{aligned}\dot{x}_1(t) &= 1 - x_1(t) - Da_1x_1(t) + Da_2x_2^2(t) \\ \dot{x}_2(t) &= -x_2(t) + Da_1x_1(t) - Da_2x_2^2(t) - Da_3d_2(t)x_2^2(t) + u(t) \\ \dot{x}_3(t) &= -x_3(t) + Da_3d_2(t)x_2^2(t)\end{aligned}\quad (30)$$

Here, $x_1(t)$, $x_2(t)$ and $x_3(t)$ represent the concentrations of reactant A, B and C, respectively. $x_3(t)$ is the state that is to be controlled and $u(t)$ is the control input. Also, $d_2(t)$ is a time-varying parameter. In the simulations, parameters are taken as $Da_1 = 3$, $Da_2 = 0.5$, $Da_3 = 1$. The nominal value of $d_2(t)$ is selected as $d_{2nominal}(t) = 1$ [22, 31, 47, 50]. The control signal is restricted between $u_{min} = 0$ and $u_{max} = 1$; and its duration is set as $\tau_{min} = \tau_{max} = 1$ s. Fourth order Runge Kutta integration method is employed in simulating the model and sampling period is designated as 0.1 seconds. In the design of online SVR structure, there are two important parameters: ε and C . ε is the maximum tolerable error and its value is critical, since if it is chosen too large the modeling error will be high and the dynamics of the controlled system will not be estimated correctly. On the other hand, if it is taken too small, the number of support vectors will be high and consequently the computational load will increase. So, the value of ε must be determined by taking into account the trade-off between modeling error and computational load. The second parameter C basically determines the size of the search space for Lagrange multipliers and eventually the maximum and minimum values of the Lagrange values are constrained by the search space. C also acts as a penalty term for the training examples outside the ε -tube. In simulating the online SVR, training error tolerance parameter ε is selected as $\varepsilon = 10^{-3}$, C_α and C_β are set as 1000, the kernel function is chosen as $K(x, y) = e^{\frac{-||x-y||}{2\sigma^2}}$. In realizing the kernel function, the parameter σ is taken as $\sigma = 1$. Small values of σ lead to oscillations in system response while large values result in a loss of nonlinearity of kernel function, which means that SVR structure will not effectively estimate the nonlinear system dynamics. Simulations are implemented to assess the conduct of the proposed controller for tracking control problem (nominal case). Consequently, the robustness of the method has also been evaluated when noise and uncertainty have been added. Finally, the performance of the proposed method has been compared with another SVR based NARMA-L2 controller from the technical literature [17]. For all cases, input vector has been utilized as $\mathbf{X} = \begin{bmatrix} u[n-1] \cdots u[n-3] y[n] \cdots y[n-2] \end{bmatrix}^T$.

3.1 Nominal case

The proposed control architecture is employed for the tracking control of the CSTR system briefly described above. Simulations have been repeated for two different reference signals, namely for a variable step signal (staircase) and a sinusoidal signal. Figure 5 depicts the achievement of the proposed controller in trajectory tracking problem for both types of reference signals and the computed control input. It can be clearly seen in Figure 5 (a) and (c) that the system output converges

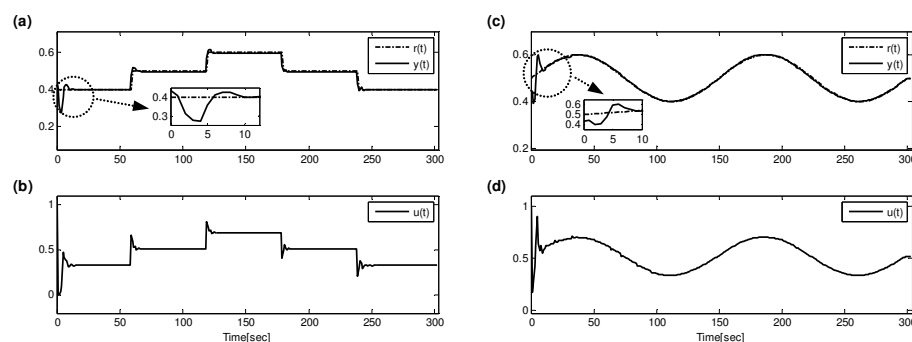


Fig. 5 (a,c) System output, (b,d) control input for variable step input and sinusoidal input (Nominal case).

to the reference input very fast, after a short duration of oscillation, the system response settles with a minimal amount of steady state error. The most conspicuous feature of the obtained system response is that it is very fast in achieving tracking. So, it is justified in the figures that the designed controller provides successful tracking of the reference input. Figures 6 (a), (b), (c) show the first two Lagrange multipliers and bias values of online SVR for the subdynamics $\hat{\mathbf{F}}$ for variable step input, Figures 6 (e), (f) and (g) depict the same parameters for subdynamics $\hat{\mathbf{G}}$. These figures illustrate how fast the SVR parameters adjust themselves to obtain an accurate tracking performance for the case of a changing reference input signal. Figure 6 (d) illustrates a comparison of the estimated model output versus actual system output for variable step input and Figure 6 (h) provides the same plot for sinusoidal input. It is clearly seen in the figures that the NARMA-L2 model block successfully estimates the system output with very small modeling error.

3.2 Measurement Noise

In the second set of simulations, robustness of the introduced control method is evaluated. To this end, the measured system output is contaminated with a Gaussian noise of zero mean and a signal-to-noise ratio (SNR) of 30 dB. SNR is defined as

$$\text{SNR} = 10 \log_{10} \left(\frac{\sigma_y^2}{\sigma_v^2} \right) \text{ dB} \quad (31)$$

Here, σ_y^2 denotes system output variance and σ_u^2 is the additive noise variance [22]. In Figure 7 (a) the output of the controlled system is depicted for variable step input and in (c) system output is shown for sinusoidal input. It can be clearly seen that although the response is oscillatory due to the introduced measurement noise, the system output can successfully track the reference input. In Figures 7 (b) and (d) the control input signal applied to the system is illustrated for variable step and sinusoidal type of reference signals, respectively. It is observed that the control signal changes abruptly and in an oscillatory manner to compensate for the noise in the system, the relatively larger magnitude oscillations in the control

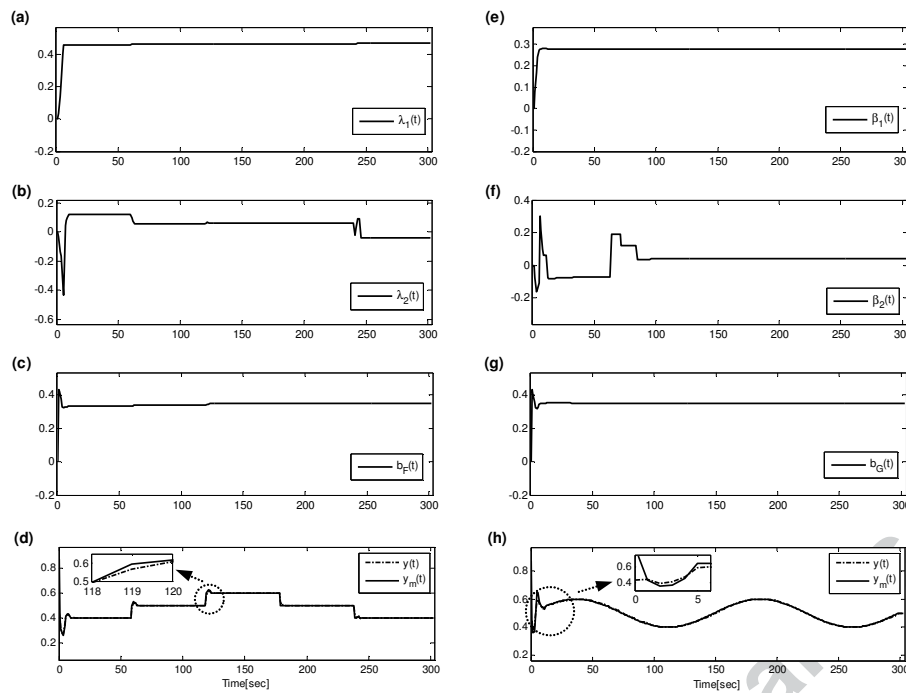


Fig. 6 (a,b,c) Regression parameters of subdynamic $\hat{\mathbf{F}}(\lambda[n], \mathbf{X}[n])$, (d) NARMA-L2 model output ($y_m(t)$) versus actual output ($y(t)$) for variable step input, (e,f,g) Regression parameters of the subdynamic $\hat{\mathbf{G}}(\beta[n], \mathbf{X}[n])$ (h) NARMA-L2 model output ($y_m(t)$) versus actual output ($y(t)$) for sinusoidal input (Nominal case).

input signal successfully suppress the magnitudes of the oscillations in the system output. In Figure 8 (a) and (b), the first two Lagrange multipliers λ_1 and λ_2 are graphed and Figure 8 (c) shows the bias value b_F , all calculated for the estimated model of the subdynamics $\hat{\mathbf{F}}$ for the case of variable step reference input. Similarly, Figure 8 (e), (f) and (g) illustrate the Lagrange multipliers β_1 and β_2 and bias value b_G , computed by the online SVR model of the subdynamics $\hat{\mathbf{G}}$ for variable step input. In all these figures, it is clearly observed that the parameters adapt themselves continuously to account for the effect of the measurement noise and suppress it successfully to lead to good tracking of the reference input with small oscillations. Finally, Figure 8 (d) represents a comparison of the actual system output and the output of the model as estimated by the online SVR for variable step input, similarly Figure 8 (h) shows the same plot for sinusoidal input. As it is clearly seen from Figure 8 (g) and (h), the model of the system is successfully estimated by the online SVR model block, the enlarged parts in the figure show how the modeling error is small in the case of added measurement noise to the system.

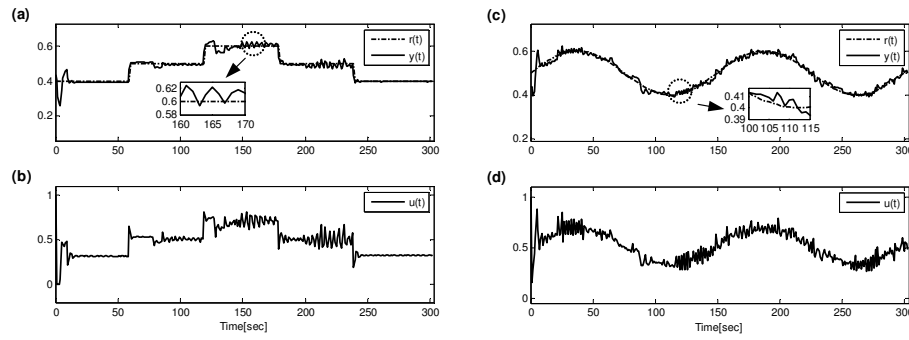


Fig. 7 (a,c) System output (b,d) control input for variable step input and sinusoidal input (Measurement noise case).

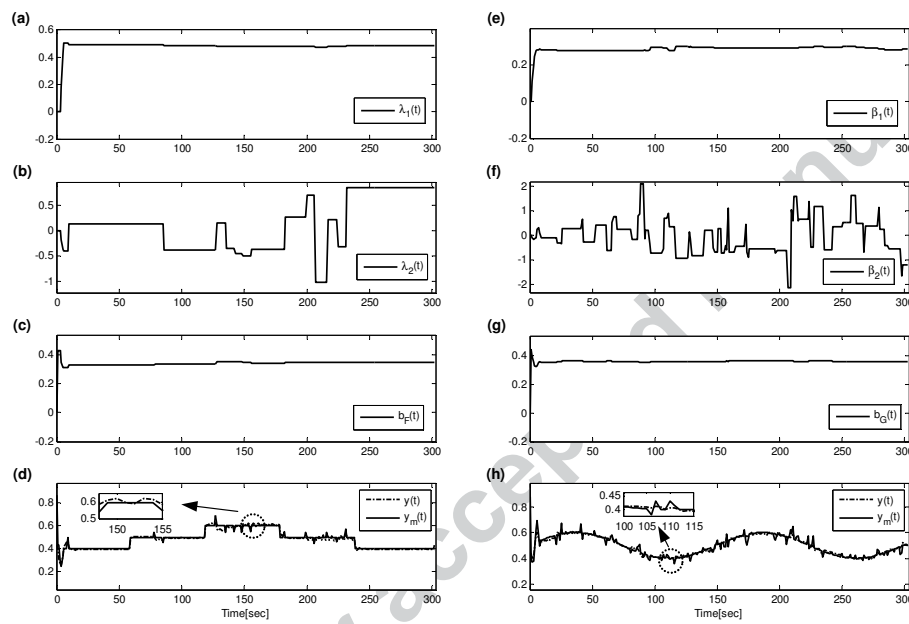


Fig. 8 (a,b,c) Regression parameters of subdynamic $\hat{\mathbf{F}}(\lambda[n], \mathbf{X}[n])$, (d) NARMA-L2 model output ($y_m(t)$) versus actual output ($y_m(t)$) for variable step input, (e,f,g) Regression parameters of the subdynamic $\hat{\mathbf{G}}(\beta[n], \mathbf{X}[n])$ (h) NARMA-L2 model output ($y_m(t)$) versus actual output ($y(t)$) for sinusoidal input (Measurement noise case).

3.3 Parametric uncertainty

Simulations are performed also to evaluate the robustness of suggested control technique against parametric uncertainty. For this, time varying parameter $d_2(t)$ is manipulated to fluctuate about its initial value with $d_2(t) = 1 + 0.1 \sin(2\pi \frac{1}{30} t)$. This is the most difficult case to control and the results must be interpreted carefully to verify the performance of the proposed controller. Figure 9 (a) shows the

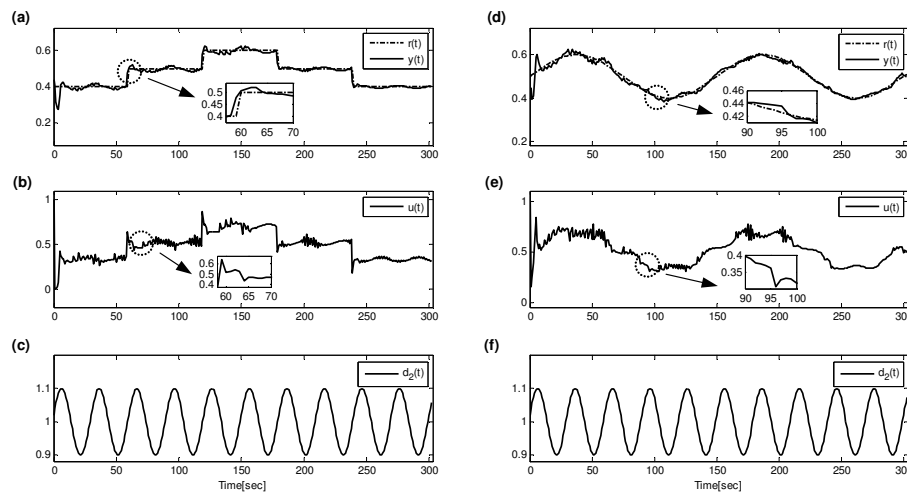


Fig. 9 (a,d) System output (b,e) control input, (c,f) uncertain parameter for variable step input and sinusoidal input(Parametric uncertainty case).

output of the controlled system for the variable step reference input and (b) depicts the control input signal computed for this case. Figure 9 (c) and (d) illustrate the system output and control input for sinusoidal type of reference input, respectively. A detailed analysis of the graphs justifies how the control action changes abruptly and continuously to act against the effect of the changing system parameter and smoothes the system response. The enlarged parts of the graph depict in detail the adaptation of the control signal and its effect on the system response. Figure 9 (c) and (f) are the graphs of the uncertain parameter $d_2(t)$, for the variable step input and the sinusoidal input, respectively. Figure 10 illustrates some of the parameters regarding the model estimation process. Figure 10 (a), (b) show the Lagrange parameters λ_1 and λ_2 for the estimated subdynamics $\hat{\mathbf{F}}$ by the online SVR model estimation block, for the case of variable step reference input, (c) depicts the bias term b_F . In a similar way, Figure 10 (e), (f) and (g) are the graphs of the first two Lagrange multipliers β_1 and β_2 and bias value b_G for the subdynamics $\hat{\mathbf{G}}$ for variable step input, respectively. Again, the fast adaptation of the model parameters can be observed from the figures to estimate the system model precisely in the presence of fluctuating system parameter. Finally, Figure 10 (d) and (h) show the output estimated by the model of the system for the variable step and sinusoidal reference inputs, respectively. It is clearly seen from these two figures that the NARMA-L2 model block in Figure 3, realized by the online SVR estimators can successfully obtain the system model. The enlarged parts show in detail, how the induced oscillations in the model output due to the fluctuating system parameter $d_2(t)$ are suppressed and the model output follows the actual output smoothly.

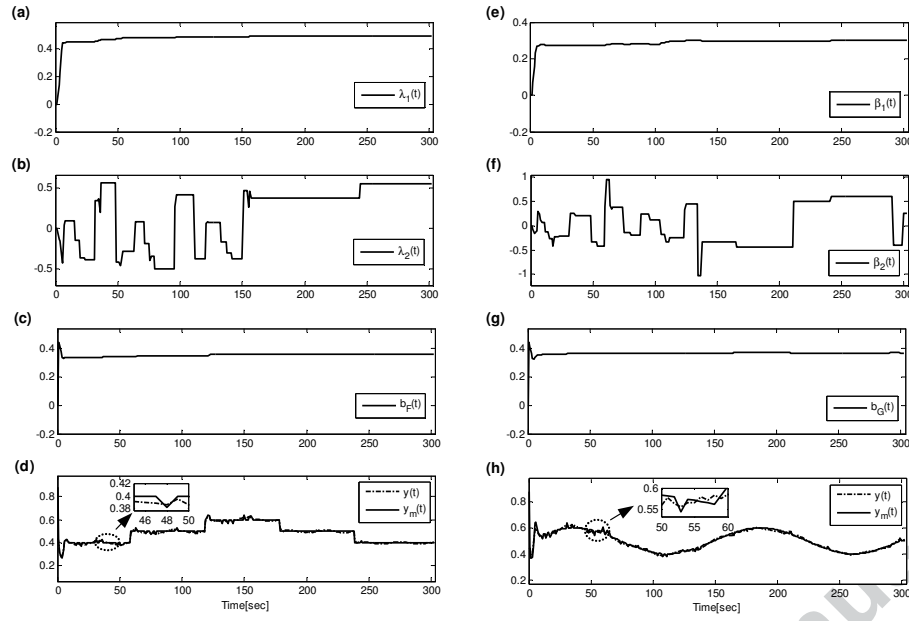


Fig. 10 (a,b,c) Regression parameters of subdynamic $\hat{\mathbf{F}}(\boldsymbol{\lambda}[n], \mathbf{X}[n])$, (d) NARMA-L2 model output ($y_m(t)$) versus actual output ($y(t)$) for variable step input, (e,f,g) Regression parameters of the subdynamic $\hat{\mathbf{G}}(\boldsymbol{\beta}[n], \mathbf{X}[n])$ (h) NARMA-L2 model output ($y_m(t)$) versus actual output ($y(t)$) for sinusoidal input (Parametric uncertainty case).

3.4 Comparison with an SVR based NARMA-L2 controller from literature

The performance of the proposed adaptive NARMA-L2 controller has been compared with another NARMA-L2 type controller based on SVR methodology that was described in [17]. In [17], firstly, the NARX model of the nonlinear plant to be controlled is obtained using an online SVR estimator. The generic NARX model of a nonlinear system is given by:

$$y_{NARX}[n+d] = \mathbf{F}(u[n], \dots, u[n-k+1], y[n], \dots, y[n-k+1]) \quad (32)$$

This NARX model is estimated using online SVR as:

$$\hat{y}_{NARX} = \sum_{i=1}^N \alpha_i K(\mathbf{X}_i, \mathbf{X}_c) + b_\alpha \quad (33)$$

The NARX model given in (33) is decomposed into NARMA-L2 model by:

$$\begin{aligned} \hat{\mathbf{F}}(\mathbf{X}_n) &= \sum_{i=1}^N \mu_i K(\mathbf{X}_i, \mathbf{X}_n) + b_\mu \\ \hat{\mathbf{G}}(\mathbf{X}_n) &= \sum_{i=1}^N \theta_i K(\mathbf{X}_i, \mathbf{X}_n) + b_\theta \end{aligned} \quad (34)$$

So, the NARMA-L2 model of the nonlinear plant can be expressed as:

$$\begin{aligned}\hat{y}_{NARMA}[n+1] &= \hat{\mathbf{F}}(\mathbf{X}_n) + \hat{\mathbf{G}}(\mathbf{X}_n)u[n] \\ \hat{y}_{NARMA}[n+1] &= \sum_{i=1}^N \mu_i K(\mathbf{X}_i, \mathbf{X}_n) + b_\mu + \left(\sum_{i=1}^N \theta_i K(\mathbf{X}_i, \mathbf{X}_n) + b_\theta \right) u[n] \\ \hat{y}_{NARMA}[n+1] &= \sum_{i=1}^N (\mu_i + \theta_i u[n]) K(\mathbf{X}_i, \mathbf{X}_n) + b_\mu + b_\theta u[n]\end{aligned}\quad (35)$$

Since both the NARX model given in (33) and the NARMA-L2 model in (35) represent the same plant they can be equated to obtain the subdynamics terms $\hat{\mathbf{F}}(\mathbf{X}_n)$ and $\hat{\mathbf{G}}(\mathbf{X}_n)$:

$$\begin{aligned}\hat{y}_{NARX} &\cong \hat{y}_{NARMA} \\ \sum_{i=1}^N \alpha_i K(\mathbf{X}_i, \mathbf{X}_n) + b_\alpha &= \sum_{i=1}^N (\mu_i + \theta_i u[n]) K(\mathbf{X}_i, \mathbf{X}_n) + b_\mu + b_\theta u[n]\end{aligned}\quad (36)$$

From (36), it can be deduced that:

$$\begin{aligned}\alpha_i &= \mu_i + \theta_i u[n] \\ b_\alpha &= b_\mu + b_\theta u[n]\end{aligned}\quad (37)$$

In order to obtain unique subdynamical terms $\hat{\mathbf{F}}(\mathbf{X}_n)$ and $\hat{\mathbf{G}}(\mathbf{X}_n)$, the following assumption is made:

$$\begin{aligned}\mu_i &= \vartheta_1(\cdot)\theta_i \\ b_\mu &= \vartheta_2(\cdot)b_\theta\end{aligned}\quad (38)$$

Then the following relations can be formulated:

$$\begin{aligned}\hat{\mathbf{F}}(\mathbf{X}_n) &= \vartheta_1(\cdot)\hat{\mathbf{G}}(\mathbf{X}_n) + [\vartheta_2(\cdot) - \vartheta_1(\cdot)]b_\theta \\ \theta_i &= \frac{\alpha_i}{\vartheta_1(\cdot) + u[n]}, \mu_i = \vartheta_1(\cdot)\theta_i \\ b_\theta &= \frac{b_\alpha}{\vartheta_2(\cdot) + u[n]}, b_\mu = \vartheta_2(\cdot)b_\theta\end{aligned}\quad (39)$$

After $\hat{\mathbf{F}}(\mathbf{X}_n)$ and $\hat{\mathbf{G}}(\mathbf{X}_n)$ are obtained, they are substituted into (9) to calculate the control input. Since the selection of $\vartheta_1(\cdot)$ and $\vartheta_2(\cdot)$ is not unique, in [17], their optimal values are determined by formulating a cost function and optimizing it using Levenberg-Marquardt algorithm. More detailed information about the design can be found in [17]. The proposed adaptive NARMA-L2 controller and the design approach given in [17] are comparable, since they are both based on NARMA-L2 controller principle and SVR modeling methodology. However, it can be seen that the proposed adaptive NARMA-L2 controller we put forward here is a more direct approach where the subdynamical terms $\hat{\mathbf{F}}(\mathbf{X}_n)$ and $\hat{\mathbf{G}}(\mathbf{X}_n)$ are directly calculated without defining any intermediate variables. The strength of our approach comes from obtaining the SVR models of $\hat{\mathbf{F}}(\mathbf{X}_n)$ and $\hat{\mathbf{G}}(\mathbf{X}_n)$ in a single step. When we compare the two approaches it will be obvious that the computational load of the proposed method is lower. Also, since there is no need

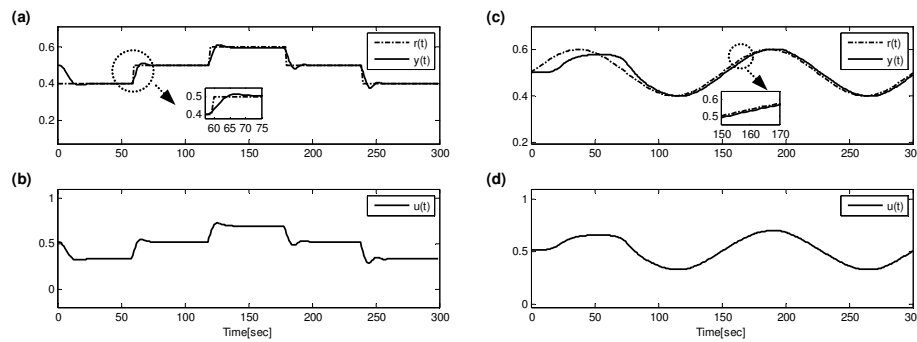


Fig. 11 (a,d) System output (b,e) control input, (c,f) uncertain parameter for variable step input and sinusoidal input for the NARMA-L2 controller in [17] (Nominal case).

to calculate the auxiliary variables $\vartheta_1(\cdot)$ and $\vartheta_2(\cdot)$, the design process is significantly simpler. Simulations have been repeated for the CSTR system with the implementation of NARMA-L2 controller described in [17]. For a fair comparison prediction horizon(K) in [17] is taken as “1” for $\vartheta_1(\cdot)$ and $\vartheta_2(\cdot)$ parameters in our simulations. Figure 11 (a) illustrates the system output for variable step input and (c) depicts the system output for sinusoidal input obtained by the NARMA-L2 controller of [17]. When these figures are compared with the plots in Figure 5 (a) and (c), it can apparently be seen that the proposed adaptive NARMA-L2 controller of this study performs much better than the controller of [17], the output settles fastly and tracks the reference successfully. Figure 11 (c) clearly shows a slower settling time. Figure 11 (b) and (d) show the control input computed by the controller in [17]. It must be noted that one of the reasons of the apparent delay observed in Figure 11 (a) and (c) is that the Levenberg-Marquardt algorithm needs some time to search for the optimal auxiliary variables $\vartheta_1(\cdot)$ and $\vartheta_2(\cdot)$. Figure 12 (a) and (c) show the system output obtained by the controller of [17] for variable step input and sinusoidal input respectively, when measurement noise is added to the system. Figure 12 (b) and (d) depict the control input obtained by [17], for variable step and sinusoidal type of reference inputs. When compared with Figure 7, it can be deduced that with the adaptive NARMA-L2 controller we propose the settling time is shorter, overshoot is smaller and tracking performance is better, which is especially obvious for the sinusoidal reference. Less oscillations are observed in Figure 12 (a), (b), (c) and (d) since a filter has been utilized in the design implemented in [17]. The comparison between the adaptive NARMA-L2 controller proposed here and the NARMA-L2 controller of [17] has also been performed when parametric uncertainty is added to the system. For the CSTR system described in detail above, again $d_2(t)$ is chosen as the uncertain parameter and fluctuated with $d_2(t) = 1 + 0.1\sin(2\pi\frac{1}{30}t)$. The results obtained with the controller of [17] are depicted in Figure 13. Figure 13 (a) and (d) illustrate the system output for variable step input and sinusoidal input. In Figure 13 (b) and (e), the computed control input signal is shown and (c) and (f) depict the fluctuating parameter $d_2(t)$, for variable step input and sinusoidal input, respectively. When the results in Figure 9 and Figure 13 are compared, it is apparent that when the adaptive NARMA-L2 controller proposed in this paper

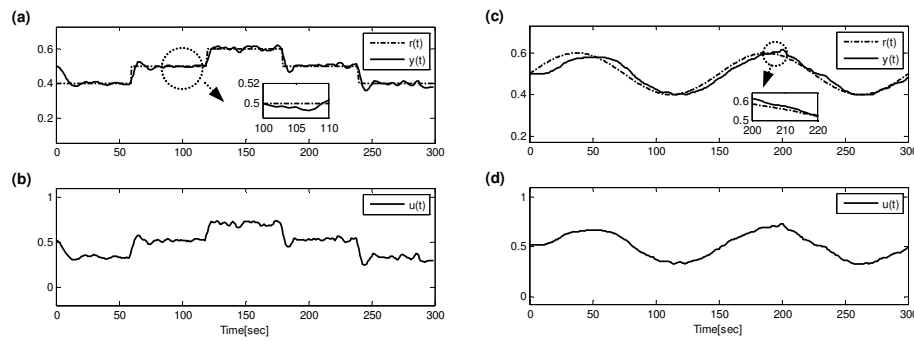


Fig. 12 (a,d) System output (b,e) control input, (c,f) uncertain parameter for variable step input and sinusoidal input for the NARMA-L2 controller in [17] (Measurement noise case).

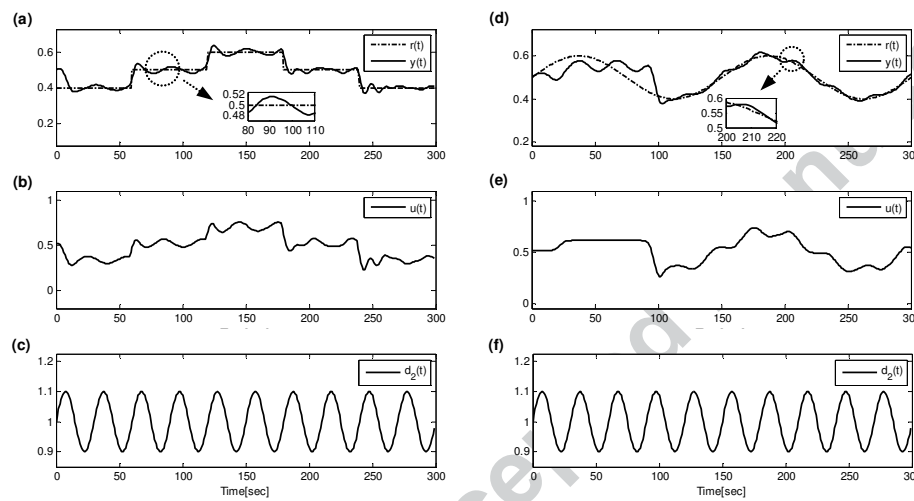


Fig. 13 (a,d) System output (b,e) control input, (c,f) uncertain parameter for variable step input and sinusoidal input for the NARMA-L2 controller in [17] (Parametric uncertainty case).

is applied, the system response has fast settling time, small overshoot and better tracking performance with respect to the controller given in [17]. A comparison of the control input signal in Figure 9 (b) and (d) with Figure 12 (b) and (d) clearly shows how the proposed adaptive NARMA-L2 controller changes abruptly and continuously to compensate for the effect of the parametric uncertainty added to the system. To provide a numerical comparison, a cost function J_{comp} is defined in terms of the tracking error given as:

$$J_{comp} = \mathbf{e}^T \mathbf{e}, \quad \mathbf{e} = \mathbf{r} - \mathbf{y} \quad (40)$$

In Figure 14, J_{comp} is plotted for the proposed adaptive NARMA-L2 controller discussed in this paper, and the NARMA-L2 controller given in [17]. The results are given for both the variable step and sinusoidal reference input signals, and for all

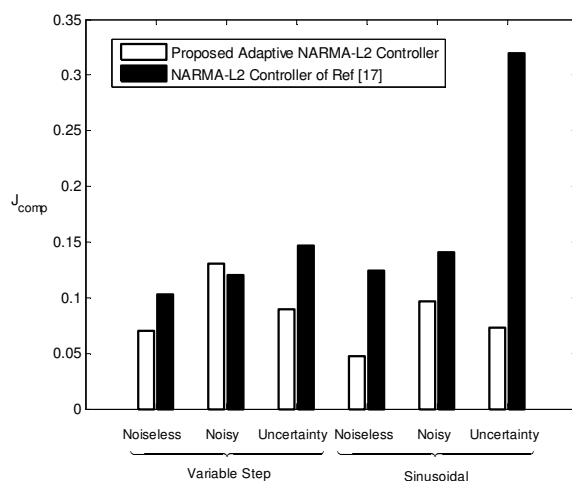


Fig. 14 Comparison of the tracking performance of the proposed adaptive NARMA-L2 controller and the NARMA-L2 controller in [17].

the cases, including the nominal case without noise and parametric uncertainty, for the case when measurement noise is added to the system and also for the case when parametric uncertainty is added to the system. The figure shows that the proposed adaptive NARMA-L2 controller achieves better tracking performance. This is more conspicuous for the tracking problem of the sinusoidal input. To compare the real time applicability of the two controllers, Table 1 and Table 2 present an analysis for the computational load. Table 1 gives a list of the operations for the adaptive NARMA-L2 controller proposed in this paper, the duration of each operation in (ms.) for both variable step and sinusoidal types of reference inputs and for all the cases, the nominal case without noise and parametric uncertainty, for the case when measurement noise is added to the system and for the case when parametric uncertainty is added to the system. Table 2 lists the same type of information for the NARMA-L2 controller discussed in reference [17]. It should be noted that the main operations are different since the basic procedures of the two controllers are different. A comparison can be made based on the total loop time which is basically the summation of all the operations in one iteration and represents time in (ms) for one sampling time in the simulations. As can be observed from Table 1 and Table 2, the total loop time is shorter for the proposed adaptive NARMA-L2 controller for all the cases and for all types of reference signals, in comparison to the NARMA-L2 controller of [17]. This justifies the fact that the computational load of the proposed controller is less than the compared controller of [17]. An analysis of Figures 5, 7, 9, 11, 12, 13 and 14, Table 1 and 2 leads us to the following conclusion: In comparison to the NARMA-L2 controller given in [17], the adaptive NARMA-L2 controller proposed in this study achieves shorter settling time, smaller overshoot, better tracking performance and since its basic methodology is simpler and more straightforward, the computational load is also less. An important advantage of the proposed adaptive NARMA-L2 controller is that it is a direct approach, online SVR is employed to estimate the subdynamical terms $\hat{\mathbf{F}}(\mathbf{X}_n)$ and $\hat{\mathbf{G}}(\mathbf{X}_n)$ in a

Table 1 Computational load of the proposed adaptive NARMA-L2 controller (ms).

Reference Type	Variable Step			Sinusoidal		
	Nominal	Noisy	Uncertainty	Nominal	Noisy	Uncertainty
Training Time	12.5559	9.9567	26.2754	12.5526	13.1747	26.818
Prediction Time	0.61273	0.57875	0.57028	0.68933	0.57202	0.57595
Controller	1.4188	1.4038	1.4863	1.6939	1.3946	1.4589
Miscellaneous	1.8594	1.9293	1.9191	2.232	1.8555	1.8942
Total Loop Time	16.4468	13.8686	30.251	17.1679	16.9968	30.7471

Table 2 Computational load of the NARMA-L2 controller in [17] (ms).

Reference Type	Variable Step			Sinusoidal		
	Nominal	Noisy	Uncertainty	Nominal	Noisy	Uncertainty
Cases/Operations						
NARX/NARMA Conversion	1.1596	1.1245	1.1375	1.2113	1.2875	1.2866
Jacobian Matrix	1.5606	1.6188	3.0701	4.2963	1.5324	5.3603
Parameter Tuning	0.039433	0.038141	0.05237	0.091931	0.043815	0.084297
NARX Model Training	13.2287	17.8152	25.6888	8.5691	13.7942	22.1658
Miscellaneous	4.1028	3.9604	3.886	4.2496	4.4813	4.3989
Total Loop Time	20.0912	24.557	33.8348	18.4183	21.1392	33.2959

single step which are consequently utilized to calculate the NARMA-L2 control input signal, no model conversion or auxiliary variables are needed.

4 Conclusion

In this paper, online SVR is employed to obtain the dynamics of a NARMA-L2 model and consequently this estimated model is adopted to construct a NARMA-L2 controller. Although in technical literature there are various applications of NARMA-L2 controller where the system model is obtained by gradient-based techniques, it is well known that gradient descent has the drawback of getting stuck at the local minimum. On the other hand, SVR is generally preferred for its better generalization property and guarantee of global minimum. Here, the simplicity and effectiveness of NARMA-L2 controllers are combined with the strengths of SVR and a novel method is proposed where the subdynamics of the NARMA-L2 model are directly computed with an online SVR. **The main contribution of the proposed controller is combining online SVR technique and NARMA-L2 methodology in a direct approach where system model is obtained in a single step. Here, the NARMA-L2 model of the system is estimated from input-output data using an online SVR, this is a straightforward implementation as opposed to existing literature where model conversions and auxiliary variables are employed [17].** Tracking performance is evaluated with simulations on the CSTR benchmark problem, for both a variable step signal and a sinusoidal signal. The robustness of the proposed method is assessed under noisy conditions and also when there is parametric uncertainty in the system. **Additionally, the performance of the proposed controller is compared with another SVR based NARMA-L2 controller from the literature and a detailed comparison is presented.** Simulation results have proved that the method achieves good performance for all testing conditions.

Compliance with Ethical Standards

Conflict of Interest

The authors declare that there is no conflict of interests regarding the publication of this paper.

Ethical Approval

This article does not contain any studies with human participants or animals performed by any of the authors.

References

1. Narendra KS, Mukhopadhyay S (1997) Adaptive control using neural networks and approximate models. *IEEE Transactions on Neural Networks* 8(3):475–485. doi: 10.1109/72.572089
2. Majstorovic M, Nikolic I, Radovic J, Kvascev G (2008) Neural network control approach for a two-tank system. In: 9th Symposium on Neural Network Applications in Electrical Engineering (NEUREL 2008). Belgrade, Serbia
3. Pedro JO, Nyandoro OTC, John S (2009) Neural Network Based Feedback Linearisation Slip Control of an Anti-Lock Braking System. In: Asian Control Conference (ASCC 2009). Hong Kong, China
4. De Jesus O, Pukrittayakamee A, Hagan MT (2001) A comparison of neural network control algorithms. In: International Joint Conference on Neural Networks (IJCNN'01). Washington, D.C.
5. Pukrittayakamee A, De Jesus O, Hagan MT (2002) Smoothing the Control Action for NARMA-L2 Controllers. In: 45th Midwest Symposium on Circuits and Systems (MWSCAS 2002). Tulsa, OK
6. Hagan MT, Demuth HB, De Jesus O (2002) An introduction to the use of neural networks in control systems. *International Journal of Robust and Nonlinear Control* 12(11): 959–985. doi: 10.1002/rnc.727
7. Wahyudi, Mokri SS, Shafie AA (2008) Real Time Implementation of NARMA L2 Feedback Linearization and Smoothed NARMA L2 Controls of a Single Link Manipulator. In: International Conference on Computer and Communication Engineering. Kuala Lumpur, Malaysia
8. Akbarimajd A, Kia S (2010) NARMA-L2 Controller for 2-DoF Underactuated Planar Manipulator. In: International Conference on Control, Automation, Robotics and Vision (ICARCV 2010). Singapore
9. Vesselenyi T, Dzitac S, Dzitac I, Manolescu MJ (2007) Fuzzy and Neural Controllers for a Pneumatic Actuator. *International Journal of Computers Communications and Control* 2(4):375–387.
10. Awwad A, Abu-Rub H, Toliyat HA (2008) Nonlinear Autoregressive Moving Average (NARMA-L2) Controller for Advanced AC Motor Control. In: 34th Annual Conference of the IEEE Industrial Electronics Society (IECON 2008). Orlando, FL
11. Pedro J, Ekoru J (2013) NARMA-L2 Control of a Nonlinear Half-Car Servo-Hydraulic Vehicle Suspension System. *Acta Polytechnica Hungarica* 10(4): 5–26.
12. Lutfy OF, Selamat H (2015) Wavelet Neural Network-Based NARMA-L2 Internal Model Control Utilizing Micro-artificial Immune Techniques to Control Nonlinear Systems. *Arabian Journal for Science and Engineering* 40(9): 2813–2828.
13. Paul R, Chokkadi S (2016) Implementation of NARMA-L2 Controller for Shell and Tube Heat Exchanger Temperature Process. *Industrial and Engineering Chemistry Research* 55(19): 5644–5653.
14. Al-Dunainawi Y, Abbod MF, Jizany A (2017) A new MIMO ANFIS-PSO based NARMA-L2 controller for nonlinear dynamic systems. *Engineering Applications of Artificial Intelligence* 62: 265–275.

15. Yang Y, Xiang C, Gao SH, Lee TH (2018) Data-driven identification and control of nonlinear systems using multiple NARMA-L2 models. *International Journal of Robust and Nonlinear Control* 28(12):3806–3833, Special Issue: SI.
16. Vapnik VN (1999) An Overview of Statistical Learning Theory. *IEEE Transactions on Neural Networks* 10(5):988–999. doi: 10.1109/72.788640
17. Uçak K, Günel GÖ (2016) A Novel Adaptive NARMA-L2 Controller Based on Online Support Vector Regression for Nonlinear Systems. *Neural Processing Letters* 44(3): 857–886
18. Şen GD, Günel GÖ (2019) A NARMA-L2 Controller Based on Online LSSVR for Nonlinear Systems. In: 15th European Workshop on Advanced Control and Diagnosis.
19. Wanfeng S, Shengdun Z, Yajing S (2008) Adaptive PID Controller Based on Online LSSVM Identification. In: IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM 2008). Xian, China
20. Zhao J, Li P, Wang Xs (2009) Intelligent PID Controller Design with Adaptive Criterion Adjustment via Least Squares Support Vector Machine. In: 21st Chinese Control and Decision Conference (CCDC 2009). Guilin, China
21. Yuan XF, Wang YN, Wu LH (2008) Composite feedforward-feedback controller for generator excitation system. *Nonlinear Dynamics* 54(4):355–364. doi: 10.1007/s11071-008-9334-6
22. Iplikci S (2010) A comparative study on a novel model-based PID tuning and control mechanism for nonlinear systems. *International Journal of Robust and Nonlinear Control* 20(13):1483–1501.
23. Takao K, Yamamoto T, Hinamoto T (2006) A Design of PID Controllers with a Switching Structure by a Support Vector Machine. In: 2006 IEEE International Joint Conference on Neural Network (IJCNN). Vancouver, Canada
24. Liu X, Yi J, Zhao D (2005) Adaptive Inverse Control System Based on Least Squares Support Vector Machines. In: 2nd International Symposium on Neural Networks(ISNN 2005). Chongqing, China
25. Wang H, Pi DY, Sun YX (2007) Online SVM regression algorithm-based adaptive inverse control. *Neurocomputing* 70(4-6):952–959. doi: 10.1016/j.neucom.2006.10.021
26. Yuan XF, Wang YN, Wu LH (2008) Adaptive Inverse Control of Excitation System with Actuator Uncertainty. *Neural Processing Letters* 27(2):125–136. doi: 10.1007/s11063-007-9064-7
27. Zhao ZC, Liu ZY, Xia ZM, Zhang JG (2012) Internal Model Control Based on LS-SVM for a Class of Nonlinear Process. In: International Conference on Solid State Devices and Materials Science (SSDMS). Macao, China
28. Zhong WM, Pi DY, Sun YX, Xu C, Chu SZ (2006) SVM Based Internal Model Control for Nonlinear Systems. In: 3rd International Symposium on Neural Networks (ISNN 2006). Chengdu, China
29. Sun CY, Song JY (2007) An Adaptive Internal Model Control Based on LS-SVM. In: International Symposium on Neural Networks(ISNN 2007). Nanjing, China
30. Wang YN, Yuan XF (2008) SVM Approximate-based Internal Model Control Strategy. *Acta Automatica Sinica* 34(2):172–179. doi:10.3724/SP.J.1004.2008.00172
31. Iplikci S (2006) Online trained support vector machines-based generalized predictive control of non-linear systems. *International Journal of Adaptive Control and Signal Processing* 20(10):599–621. doi: 10.1002/acs.919
32. Iplikci S (2006) Support vector machines-based generalized predictive control. *International Journal of Robust and Nonlinear Control* 16(17):843–862. doi: 10.1002/rnc.1094
33. Zhiying D, Xianfang W (2008) Nonlinear Generalized Predictive Control Based on Online SVR. In: 2nd International Symposium on Intelligent Information Technology Application. Shanghai, China
34. Shin J, Kim HJ, Park S, Kim Y (2010) Model predictive flight control using adaptive support vector regression. *Neurocomputing* 73(4-6):1031–1037. doi: 10.1016/j.neucom.2009.10.002
35. Wang DS, Shen JJ, Zhu SH, Jiang GP (2020) Model predictive control for chlorine dosing of drinking water treatment based on support vector machine model. *Desalination and Water Treatment* 173:133–141.
36. Pourjafari E, Reformat M (2019) A Support Vector Regression Based Model Predictive Control for Volt-Var Optimization of Distribution Systems. *IEEE Access* 7: 93352–93363.
37. Uçak K, Günel GÖ (2016) An adaptive support vector regressor controller for nonlinear systems. *Soft Computing* 20(7): 2531–2556.
38. Uçak K, Günel GÖ (2017) Generalized self-tuning regulator based on online support vector regression. *Neural Computing and Applications* 28: S775–S801.

39. Uçak K, Günel GÖ (2019) Model free adaptive support vector regressor controller for nonlinear systems. *Engineering Applications of Artificial Intelligence* 81: 47-67.
40. Uçak K, Günel GÖ An adaptive sliding mode controller based on online support vector regression for nonlinear systems. *Soft Computing* 24(6): 4623-4643.
41. Ma J, Theiler J, Perkins S (2003) Accurate Online Support Vector Regression. *Neural Computation* 15(11):2683-2703.
42. Wang X, Du Z, Chen Z, Pan F (2009) Dynamic Modeling of Biotechnical Process Based on Online Support Vector Machine. *Journal of Computers* 4(3): 251-258.
43. Uçak K, Üstoğlu İ, Günel GÖ (2018) Safety-Critical Support Vector Regressor Controller for Nonlinear Systems. *Neural Processing Letters* 48: 419-440.
44. Uçak K (2016) Support Vector Regression based Controller Design Methods for Nonlinear Systems. Dissertation, Istanbul Technical University
45. Mario M (2002) On-Line Support Vector Machine Regression. In: 13th European Conference on Machine Learning (ECML 2002). Helsinki, Finland
46. Kravaris C, Palanki S (1988) Robust nonlinear state feedback under structured uncertainty. *AIChE Journal* 34(7): 1119-1127.
47. Wu W, Chou YS (1999) Adaptive feedforward and feedback control of non-linear time-varying uncertain systems. *International Journal of Control* 72(12): 1127-1138.
48. Levenspiel O (1999) *Chemical Reaction Engineering*. John Wiley and Sons, USA
49. Fogler HS(2006) *Elements of Reaction Engineering*. Pearson Education.
50. Ungar LH (1990) Neural Networks for Control. In: Miller III WT, Sutton RS, Werbos PJ(eds) *A Bioreactor Benchmark for Adaptive Network based Process Control*, MIT Press, USA, pp 387-402