International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING

ISSN:2147-6799 www.ijisae.org Original Research Paper

Runge-Kutta MIMO NARMA-L2 Controller

Kemal Uçak*[1]

Submitted: 04/05/2020 **Accepted : 12/04/2021**

Abstract: This paper introduces a novel control architecture which combines nonlinear autoregressive moving average (NARMA)-L2 control method with Runge Kutta based system modeling. Control law is developed by employing NARMA-L2 model via Taylor expansion. Jacobian information required for the controller is constituted by making use of RK model of plant to be controlled. To evaluate performance, simulations are carried out on the Three Tank System. Runge-Kutta based Generalized Predictive Controller (GPC) is deployed to compare the performance of the introduced controller. Acquired results illustrate that introduced controller has substantially good performance on multiple input multiple output (MIMO) nonlinear systems.

Keywords: *Adaptive Controller, NARMA-L2 Controller, Runge-Kutta Identification, Runge-Kutta NARMA-L2 Controller*

1. Introduction

Controller design which is as simple as possible is an intended goal in terms of implementation and analysis of the control architectures. NARMA-L2, which is an affine linear structure derived from nonlinear dynamics using Taylor series expansion, is one of the most effective techniques to design control law for nonlinear systems. Since control signal is derived via subdynamics of the NARMA-L2 model, any additional control architecture is not needed.

Literature contains numerous implementations of NARMA-L2 based controllers. Majstorovic et al [1] deployed a NARMA-L2 controller derived from offline NARMA model for a two-tank system. Pedro et al. realized NARMA-L2 to control slip in an anti-lock braking system (ABS) [2]. Hagan et al. utilized a NARMA-L2 control method using a neural network for a continuously stirred tank reactor (CSTR) system [3, 4]. To diminish oscillations in NARMA-L2 controller, Pukritttayakamee et al. [5] introduced to employ an additional term that includes a combination of reference and output signals. Wahyudi, Mokri and Shafie [6] utilized a smoothed NARMA-L2 controller to overcome the appearing nonlinearities and parametric uncertainties in a robotic link. The control of underactuated systems is difficult, and static feedback law can be deployed for asymptotically stabilization of the system, therefore Akbarimajd and Kia [7] applied NARMA-L2 control method for a 2-dof underactuated robot. Vesselenyi et al [8] used NARMA-L2 position controller for a pneumatic actuator. The common point of the studies in literature is that NARMA-L2 controller is derived from the offline system model [1-9].

Presented work acquaints an original Runge-Kutta (RK) based NARMA-L2 control technique for nonlinear MIMO systems. Parameters are identified via a RK system model. Then, using the future system outputs, a NARMA-L2 control signal is derived. To assess the performance, three tank system (TTS) dynamics are used and controller has been compelled to work under various conditions vital for control. The performance of the controller has been compared with Runge-Kutta based generalized predictive control (GPC) introduced in [10]. The obtained results indicate that the RK-NARMA-L2 controller can be conveniently applied for nonlinear MIMO systems.

The paper is organised as: NARMA model is overviewed in section 2. The details related to the introduced controller and Runge-Kutta identification technique are summarized in section 3. Results of the control performance assessment for nonlinear three tank system are illustrated in Section 4. The paper is finalized with detailed discussion in conclusion part.

2. NARMA-L2 Model and Controller

Nonlinear dynamical systems can be expressed via NARMA model around the equilibrium state [11]. NARMA model can be given in companion form:

$$y\left[n+d\right] \cong F\left[y\left[n\right]\cdots y\left[n-n_v+1\right], u\left[n\right]\cdots u\left[n-n_u+1\right]\right] \tag{1}$$

Here, d denotes relative degree [12][13]. For $d \geq 2$, if the NARMA model is rewritten as

$$y\left[n+d\right] \cong F\left[y\left[n\right]\cdots y\left[n-n_v+1\right], u\left[n+1\right]\cdots u\left[n-n_u\right]\right] \tag{2}$$

and if “n-1” is substituted in place of “n” time index, the model is obtained as

$$y\left[n+1\right] \cong F\left[y\left[n-1\right]\cdots y\left[n-n_v\right], u\left[n\right]\cdots u\left[n-n_u-1\right]\right] \tag{3}$$

NARMA-L2 model is a very effective system identification structure derived via Taylor expansion. For system model given in (3), NARMA-L2 model where control signal appears linearly [4],[7],[11]-[15] is written as

$$y\left[n+1\right] \cong f + g\,u\left[n\right] \tag{4}$$

where

$$f = F\left[y\left[n-1\right]\cdots y\left[n-n_v\right], 0\cdots u\left[n-n_u-1\right]\right] \cong y\left[n\right] \tag{5}$$

$$g = \left.\frac{\partial f}{\partial u\left[n\right]}\right|_{y\left[n-1\right]\cdots y\left[n-n_y\right], 0\cdots u\left[n-n_u-1\right]} \tag{6}$$

It is possible to express Taylor expansion for MIMO system as:

$$\mathbf{Y}\left[n+1\right] \cong \mathbf{F}\left[n\right] + \mathbf{G}\left[n\right]\mathbf{u}\left[n\right] \tag{7}$$

[1] *Muğla Sıtkı Koçman University, Department of Electrical and Electronics Engineering, Muğla-48000, Turkey,*
ORCID ID: 0000-0001-7005-7940

* *Corresponding Author Email: ucakk@mit.edu, ucak@mu.edu.tr*

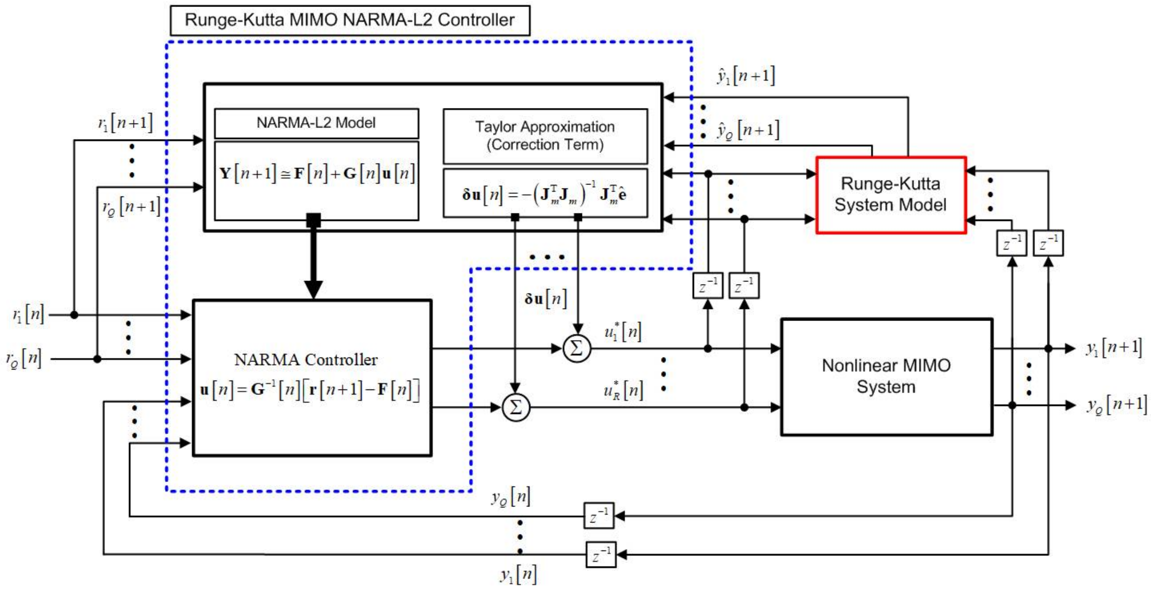
Runge-Kutta MIMO NARMA-L2 Controller
NARMA-L2 Model
$\mathbf{Y}[n+1]\cong\mathbf{F}[n]+\mathbf{G}[n]\mathbf{u}[n]$
Taylor Approximation (Correction Term)
$\boldsymbol{\delta}\mathbf{u}[n]=-(\mathbf{J}_m^T\mathbf{J}_m)^{-1}\mathbf{J}_m^T\hat{\mathbf{e}}$
$\hat{y}_1[n+1]$
$\hat{y}_Q[n+1]$
Runge-Kutta System Model
$r_1[n+1]$
$r_Q[n+1]$
$r_1[n]$
$r_Q[n]$
NARMA Controller
$\mathbf{u}[n]=\mathbf{G}^{-1}[n][\mathbf{r}[n+1]-\mathbf{F}[n]]$
$\boldsymbol{\delta}\mathbf{u}[n]$
$u_1^*[n]$
$u_R^*[n]$
Nonlinear MIMO System
$y_1[n+1]$
$y_Q[n+1]$
z^{-1}
$y_Q[n]$
$y_1[n]$

Fig. 1. A schematic diagram of RK$_{\text{NARMA-L2}}$ control architecture.

In (7),

$$\mathbf{F}[n]=\begin{bmatrix} f_1[n] \\ \vdots \\ f_q[n] \end{bmatrix},\mathbf{G}[n]=\begin{bmatrix} g_{11}[n]\cdots g_{1m}[n] \\ \vdots \quad \ddots \quad \vdots \\ g_{q1}[n]\cdots g_{qm}[n] \end{bmatrix},\ \mathbf{u}[n]=\begin{bmatrix} u_1[n] \\ \vdots \\ u_m[n] \end{bmatrix} \tag{8}$$

q and m denote output and input dimensions and:

$$\begin{aligned} & f_q[n]\cong y_q[n] \\ & g_{qm}=\frac{\partial f_q[n]}{\partial u_m[n]}=\frac{\partial y_q[n]}{\partial u_m[n]} \end{aligned} \tag{9}$$

$\mathbf{R}[n+1]$ reference signal can be substituted in place of $\mathbf{Y}[n+1]$ in order to attain control signal so that system output equals the reference signal. Required control input that forces the controlled outputs to reference signals is computed as follows:

$$\mathbf{u}[n]=\mathbf{G}^{-1}[n][\mathbf{R}[n+1]-\mathbf{F}[n]] \tag{10}$$

As observed in (8) and (9), the entries of the $\mathbf{G}[n]$ matrix are system Jacobian which is obtained using RK system model.

3. Proposed RK NARMA-L2 controller

Figure 1 demonstrates mechanism of RK NARMA-L2 controller where number of the system inputs-outputs are represented by R and Q respectively. The adjustment mechanism is composed of two crucial parts to be examined: RK NARMA-L2 controller to compute required control input and RK system model to estimate system dynamics required to constitute the $\mathbf{G}[n]$ matrix. To enhance the understandability of the proposed control architecture, RK$_{\text{NARMA-L2}}$ is used as an abbreviation to Runge-Kutta NARMA-L2 controller and RK$_{\text{model}}$ is utilized to denote Runge-Kutta system model. The adjustment mechanism given in Figure 1 consists of two main phases which are executed online consecutively: prediction and control phases.

3.1 Prediction Phase (RK$_{\text{model}}$)

The candidate signal ($\mathbf{u}^*[n]$) is calculated as given in (10) by using the system Jacobian derived via RK$_{\text{model}}$. Then, to observe the system response to the acquired $\mathbf{u}^*[n]$, it is applied to the RK$_{\text{model}}$. RK$_{\text{model}}$ detailed in Figure 2 incorporates three sub-blocks to capture the MIMO nonlinear system dynamics: raw Runge-Kutta system model, Runge-Kutta Model based EKF(RK$_{\text{EKF}}$) and Runge-Kutta based Model parameter estimation (RK$_{\text{estimator}}$) sub-blocks[10],[16]–[18].

3.1.1. Runge-Kutta System Model

Let a nonlinear continuous-time MIMO system be represented by:

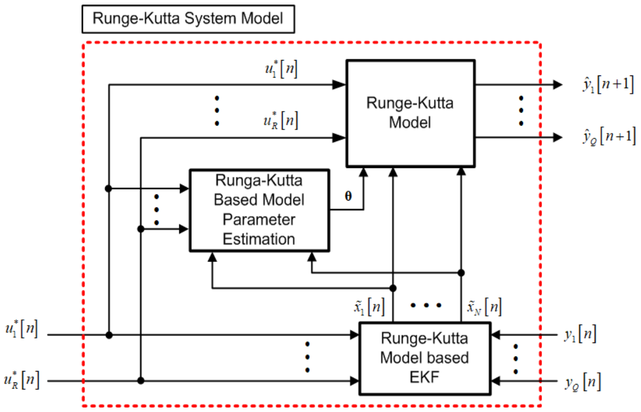

Fig. 2. Runge-Kutta System Model Subblocks [10, 16, 17, 23].

$$\begin{aligned} & \dot{\mathbf{x}}(\mathbf{t})=\mathbf{f}(\mathbf{x}(\mathbf{t}),\mathbf{u}(\mathbf{t}),\boldsymbol{\theta}) \\ & \mathbf{y}(\mathbf{t})=\mathbf{g}(\mathbf{x}(\mathbf{t}),\mathbf{u}(\mathbf{t})) \end{aligned} \tag{12}$$

In (12) $\mathbf{x}=[x_1\cdots x_N]^T\in R^N$ denotes state vector and $\mathbf{u}=[u_1\cdots u_R]^T\in R^R$ stands for input vector, $\mathbf{y}=[y_1\cdots y_Q]^T\in R^Q$ denotes system outputs, $\mathbf{f}$ and $\mathbf{g}$ are continuously differentiable functions with respect to system states and control inputs [10] [18]. The following constraints are valid for **x** and **u** [10, 16–18]

$$\begin{aligned} x_{i_{\min}} &\le x_i \le x_{i_{\max}} \\ u_{i_{\min}} &\le u_i \le u_{i_{\max}} \end{aligned} \tag{13}$$

The continuous nonlinear system given in (12) and (13) can be discretized using 4th order RK integration method, and system states are approximated with:

$$\hat{\mathbf{x}}[n+1] = \hat{\mathbf{x}}[n] + \frac{1}{6}\mathbf{K}_{1\mathrm{X}}[n] + \frac{2}{6}\mathbf{K}_{2\mathrm{X}}[n] + \frac{2}{6}\mathbf{K}_{3\mathrm{X}}[n] + \frac{1}{6}\mathbf{K}_{4\mathrm{X}}[n] \tag{14}$$

and

$$\mathbf{y}[n+1] = \mathbf{g}\left(\hat{\mathbf{x}}[n+1], \mathbf{u}[n]\right) \tag{15}$$

where

$$\begin{aligned} \mathbf{K}_{1\mathrm{X}}[n] &= T_s f\left(\hat{\mathbf{x}}[n], \mathbf{u}[n], \boldsymbol{\theta}\right) \\ \mathbf{K}_{2\mathrm{X}}[n] &= T_s f\left(\hat{\mathbf{x}}[n] + \frac{1}{2}\mathbf{K}_{1\mathrm{X}}[n], \mathbf{u}[n], \boldsymbol{\theta}\right) \\ \mathbf{K}_{3\mathrm{X}}[n] &= T_s f\left(\hat{\mathbf{x}}[n] + \frac{1}{2}\mathbf{K}_{2\mathrm{X}}[n], \mathbf{u}[n], \boldsymbol{\theta}\right) \\ \mathbf{K}_{4\mathrm{X}}[n] &= T_s f\left(\hat{\mathbf{x}}[n] + \mathbf{K}_{3\mathrm{X}}[n], \mathbf{u}[n], \boldsymbol{\theta}\right) \end{aligned} \tag{16}$$

The discrete system model in (14)-(16) can be rewritten as follows:

$$\begin{aligned} \hat{\mathbf{x}}[n+1] &= \hat{\mathbf{f}}\left(\hat{\mathbf{x}}[n], \mathbf{u}[n], \boldsymbol{\theta}\right) \\ \hat{\mathbf{y}}[n+1] &= \mathbf{g}\left(\hat{\mathbf{x}}[n], \mathbf{u}[n]\right) \end{aligned} \tag{17}$$

Using approximated states as entry in (17) repetitively, a predictive model can be derived as:

$$\begin{aligned} \hat{\mathbf{x}}[n+k] &= \hat{\mathbf{f}}\left(\hat{\mathbf{x}}[n+k-1], \mathbf{u}[n], \boldsymbol{\theta}\right) \\ \hat{\mathbf{y}}[n+k] &= \mathbf{g}\left(\hat{\mathbf{x}}[n+k-1], \mathbf{u}[n]\right) \text{ for } k = 1, \cdots, K \end{aligned} \tag{18}$$

where K indicates prediction horizon. The model in (18) necessitates state information ($\left[\tilde{x}_1[n] \cdots \tilde{x}_N[n]\right]$) as well as accurate system parameters ($\boldsymbol{\theta}$) if there is deterioration in parameters. Therefore, RK$_{\mathrm{EKF}}$ is utilized to obtain current system states and RK$_{\mathrm{estimator}}$ is required to identify the uncertain parameters via optimization theory. Using the current system states and model parameters in recursive predictive model in (18), the system behavior can be accurately identified.

3.1.2. An overview of RK$_{\mathrm{EKF}}$

EKF is one of the effective filters to attain system states via observed system input-output. Let a discrete-time nonlinear MIMO system be represented with:

$$\begin{aligned} \mathbf{x}[n+1] &= \mathbf{h}\left(\mathbf{x}[n], \mathbf{u}[n]\right) + \mathbf{w}[n] \\ \mathbf{y}[n+1] &= \mathbf{g}\left(\mathbf{x}[n], \mathbf{u}[n]\right) + \mathbf{v}[n] \end{aligned} \tag{19}$$

Here, $\mathbf{w}$ is system noise vector with covariance matrix $\mathbf{Q}$, $\mathbf{v}$ expresses measurement noise vector with covariance matrix $\mathbf{R}$ [10], [16]. In prediction phase of EKF, states and covariance matrix can be written with $\mathbf{A}[n]$ state transition matrix as [10], [16], [19]:

$$\begin{aligned} \tilde{\mathbf{x}}^{-}[n] &= \mathbf{h}\left(\tilde{\mathbf{x}}[n-1], \mathbf{u}[n-1]\right) \\ \mathbf{P}^{-}[n] &= \mathbf{A}[n]\mathbf{P}[n-1]\mathbf{A}^{\mathrm{T}}[n] + \mathbf{Q} \end{aligned} \tag{20}$$

Then, Kalman gain matrix $\mathbf{K}[n]$ is derived, $\tilde{\mathbf{x}}^{-}[n]$ and $\mathbf{P}^{-}[n]$ in prediction step can be straightened as [10], [16]:

$$\begin{aligned} \mathbf{K}[n] &= \mathbf{P}^{-}[n]\mathbf{H}^{\mathrm{T}}[n]\left(\mathbf{H}[n]\mathbf{P}^{-}[n]\mathbf{H}^{\mathrm{T}}[n] + \mathbf{R}\right)^{-1} \\ \tilde{\mathbf{x}}[n] &= \tilde{\mathbf{x}}^{-}[n] + \mathbf{K}[n]\left(\mathbf{y}[n] - \mathbf{g}\left(\tilde{\mathbf{x}}^{-}[n], \mathbf{u}[n-1]\right)\right) \\ \mathbf{P}[n] &= \left(\mathbf{I} - \mathbf{K}[n]\mathbf{H}[n]\right)\mathbf{P}^{-}[n] \end{aligned} \tag{21}$$

$\mathbf{A}[n]$ and $\mathbf{H}[n]$ in (21) are derived as [10], [16]:

$$\mathbf{A}[n] = \frac{\partial \mathbf{h}}{\partial \mathbf{x}}\bigg|_{\substack{\mathbf{x}=\tilde{\mathbf{x}}[n-1] \\ \mathbf{u}=\mathbf{u}[n-1]}} = \frac{\partial \hat{\mathbf{f}}}{\partial \mathbf{x}}\bigg|_{\substack{\mathbf{x}=\tilde{\mathbf{x}}[n-1] \\ \mathbf{u}=\mathbf{u}[n-1]}}, \quad \mathbf{H}[n] = \frac{\partial \mathbf{g}}{\partial \mathbf{x}}\bigg|_{\substack{\mathbf{x}=\tilde{\mathbf{x}}[n-1] \\ \mathbf{u}=\mathbf{u}[n-1]}} \tag{22}$$

Here,

$$\frac{\partial \hat{\mathbf{f}}}{\partial \mathbf{x}}\bigg|_{\substack{\mathbf{x}=\tilde{\mathbf{x}}[n-1] \\ \mathbf{u}=\mathbf{u}[n-1]}} = \left[\frac{\partial f_i\left(\tilde{\mathbf{x}}[n-1], \mathbf{u}[n-1]\right)}{\partial \tilde{x}_j[n-1]}\right] = \left[\frac{\partial \tilde{x}_i[n]}{\partial \tilde{x}_j[n-1]}\right] \tag{23}$$

for i=1,..,N and j=1,…,N.

$$\begin{aligned} \frac{\partial \tilde{x}_i[n]}{\partial \tilde{x}_j[n-1]} = \delta_{i,j} &+ \frac{1}{6}\frac{\partial \mathrm{K}_{1\mathrm{X}_i}[n-1]}{\partial \tilde{x}_j[n-1]} + \frac{2}{6}\frac{\partial \mathrm{K}_{2\mathrm{X}_i}[n-1]}{\partial \tilde{x}_j[n-1]} \\ &+ \frac{2}{6}\frac{\partial \mathrm{K}_{3\mathrm{X}_i}[n-1]}{\partial \tilde{x}_j[n-1]} + \frac{1}{6}\frac{\partial \mathrm{K}_{4\mathrm{X}_i}[n-1]}{\partial \tilde{x}_j[n-1]} \end{aligned} \tag{24}$$

$$\frac{\partial \mathrm{K}_{1\mathrm{X}_i}[n-1]}{\partial \tilde{x}_j[n-1]} = T_s\left[\frac{\partial f_i}{\partial x_j}\right]_{\substack{\mathbf{x}=\tilde{\mathbf{x}}[n-1] \\ \mathbf{u}=\mathbf{u}[n-1]}} \tag{25}$$

$$\frac{\partial \mathrm{K}_{2\mathrm{X}_i}[n-1]}{\partial \tilde{x}_j[n-1]} = T_s\left[\sum_{p=1}^{N}\frac{\partial f_i}{\partial x_p}\left(\delta_{p,j} + \frac{1}{2}\frac{\partial \mathrm{K}_{1\mathrm{X}_p}[n-1]}{\partial \tilde{x}_j[n-1]}\right)\right]_{\substack{\mathbf{x}=\tilde{\mathbf{x}}[n-1]+\frac{1}{2}\mathbf{K}_{1\mathrm{X}}[n-1] \\ \mathbf{u}=\mathbf{u}[n-1]}} \tag{26}$$

$$\frac{\partial \mathrm{K}_{3\mathrm{X}_i}[n-1]}{\partial \tilde{x}_j[n-1]} = T_s\left[\sum_{p=1}^{N}\frac{\partial f_i}{\partial x_p}\left(\delta_{p,j} + \frac{1}{2}\frac{\partial \mathrm{K}_{2\mathrm{X}_p}[n-1]}{\partial \tilde{x}_j[n-1]}\right)\right]_{\substack{\mathbf{x}=\tilde{\mathbf{x}}[n-1]+\frac{1}{2}\mathbf{K}_{2\mathrm{X}}[n-1] \\ \mathbf{u}=\mathbf{u}[n-1]}} \tag{27}$$

$$\frac{\partial \mathrm{K}_{4\mathrm{X}_i}[n-1]}{\partial \tilde{x}_j[n-1]} = T_s\left[\sum_{p=1}^{N}\frac{\partial f_i}{\partial x_p}\left(\delta_{p,j} + \frac{\partial \mathrm{K}_{3\mathrm{X}_p}[n-1]}{\partial \tilde{x}_j[n-1]}\right)\right]_{\substack{\mathbf{x}=\tilde{\mathbf{x}}[n-1]+\mathbf{K}_{3\mathrm{X}}[n-1] \\ \mathbf{u}=\mathbf{u}[n-1]}} \tag{28}$$

3.1.3. An overview of RK$_{\mathrm{estimator}}$

The identification performance of RK$_{\mathrm{model}}$ is directly dependent on parametric accuracy of the model($\boldsymbol{\theta}$). If system model parameters diverge from their actual values or system model parameters cannot be measured accurately, success of the identification carried out by RK$_{\mathrm{model}}$ gets worse[16]. Thus, in order to improve the identification performance of RK$_{\mathrm{model}}$, it is required to utilize online adjustment mechanism [16]. For this purpose, the following adjustment rule is deployed to optimize parameter vector of RK$_{\mathrm{model}}$($\boldsymbol{\theta}$)[10]:

$$\boldsymbol{\theta}[n+1] = \boldsymbol{\theta}[n] - \frac{\mathbf{J}_{\boldsymbol{\theta}}^{\mathrm{T}}\mathbf{e}}{\mathbf{J}_{\boldsymbol{\theta}}^{\mathrm{T}}\mathbf{J}_{\boldsymbol{\theta}}} \tag{29}$$

where

$$\mathbf{J}_{\boldsymbol{\theta}} = \left[\frac{\partial e_1[n+1]}{\partial \boldsymbol{\theta}[n]} \cdots \frac{\partial e_N[n+1]}{\partial \boldsymbol{\theta}[n]}\right]^{\mathrm{T}} = -\left[\frac{\partial \hat{x}_1[n+1]}{\partial \boldsymbol{\theta}[n]} \cdots \frac{\partial \hat{x}_N[n+1]}{\partial \boldsymbol{\theta}[n]}\right]^{\mathrm{T}} \tag{30}$$

and

$$\mathbf{e} = \begin{bmatrix} e_1[n+1] \\ \vdots \\ e_N[n+1] \end{bmatrix} = \begin{bmatrix} x_1[n+1] - \hat{x}_1[n+1] \\ \vdots \\ x_N[n+1] - \hat{x}_N[n+1] \end{bmatrix} \tag{31}$$

For the computation of Jacobian matrix in (30), $\dfrac{\partial \hat{x}_i[n+1]}{\partial \boldsymbol{\theta}}$ is required. The $\dfrac{\partial \hat{x}_i[n+1]}{\partial \boldsymbol{\theta}}$ term is derived as[10]:

$$\begin{aligned} \frac{\partial \hat{x}_i[n+1]}{\partial \boldsymbol{\theta}[n]} = \frac{\partial \hat{x}_i[n]}{\partial \boldsymbol{\theta}[n]} &+ \frac{1}{6}\frac{\partial \mathrm{K}_{1\mathrm{X}_i}[n]}{\partial \boldsymbol{\theta}[n]} + \frac{2}{6}\frac{\partial \mathrm{K}_{2\mathrm{X}_i}[n]}{\partial \boldsymbol{\theta}[n]} \\ &+ \frac{2}{6}\frac{\partial \mathrm{K}_{3\mathrm{X}_i}[n]}{\partial \boldsymbol{\theta}[n]} + \frac{1}{6}\frac{\partial \mathrm{K}_{4\mathrm{X}_i}[n]}{\partial \boldsymbol{\theta}[n]} \end{aligned} \tag{32}$$

$$
\begin{aligned}
\frac{\partial \mathrm{K}_{1\mathrm{X}_i}[n]}{\partial \boldsymbol{\theta}[n]} &= T_s\left[\frac{\partial f_i}{\partial \boldsymbol{\theta}}\right]_{\substack{\mathbf{x}=\hat{\mathbf{x}}[n]\\ \mathbf{u}=\mathbf{u}[n]\\ \boldsymbol{\theta}=\boldsymbol{\theta}[n]}}\\
\frac{\partial \mathrm{K}_{2\mathrm{X}_i}[n]}{\partial \boldsymbol{\theta}[n]} &= T_s\left[\frac{\partial f_i}{\partial \boldsymbol{\theta}}+\frac{1}{2}\sum_{j=1}^{N}\frac{\partial f_i}{\partial x_j}\frac{\partial \mathrm{K}_{1\mathrm{X}_j}[n]}{\partial \boldsymbol{\theta}}\right]_{\substack{\mathbf{x}=\hat{\mathbf{x}}[n]+\frac{1}{2}\mathbf{K}_{1\mathbf{X}}[n]\\ \mathbf{u}=\mathbf{u}[n]\\ \boldsymbol{\theta}=\boldsymbol{\theta}[n]}}\\
\frac{\partial \mathrm{K}_{3\mathrm{X}_i}[n]}{\partial \boldsymbol{\theta}[n]} &= T_s\left[\frac{\partial f_i}{\partial \boldsymbol{\theta}}+\frac{1}{2}\sum_{j=1}^{N}\frac{\partial f_i}{\partial x_j}\frac{\partial \mathrm{K}_{2\mathrm{X}_j}[n]}{\partial \boldsymbol{\theta}}\right]_{\substack{\mathbf{x}=\hat{\mathbf{x}}[n]+\frac{1}{2}\mathbf{K}_{2\mathbf{X}}[n]\\ \mathbf{u}=\mathbf{u}[n]\\ \boldsymbol{\theta}=\boldsymbol{\theta}[n]}}\\
\frac{\partial \mathrm{K}_{4\mathrm{X}_i}[n]}{\partial \boldsymbol{\theta}[n]} &= T_s\left[\frac{\partial f_i}{\partial \boldsymbol{\theta}}+\sum_{j=1}^{N}\frac{\partial f_i}{\partial x_j}\frac{\partial \mathrm{K}_{3\mathrm{X}_j}[n]}{\partial \boldsymbol{\theta}}\right]_{\substack{\mathbf{x}=\hat{\mathbf{x}}[n]+\mathbf{K}_{3\mathbf{X}}[n]\\ \mathbf{u}=\mathbf{u}[n]\\ \boldsymbol{\theta}=\boldsymbol{\theta}[n]}}
\end{aligned}
\tag{33}
$$

3.2. Control Phase (RK$_{\text{NARMA-L2}}$)

In order to constitute control law in (10), it is required to compute $\mathbf{G}[n]$ matrix. The Jacobian information can be approximated via RK$_{\text{model}}$ as follows [10]:

$$
\begin{aligned}
\frac{\partial \hat{y}_q[n+k]}{\partial u_m[n]} &= \frac{\partial g_q\left(\hat{\mathbf{x}}[n+k],\mathbf{u}[n]\right)}{\partial u_m[n]}\\
\frac{\partial \hat{y}_q[n+k]}{\partial u_m[n]} &= \left[\frac{\partial g_q}{\partial u_m}+\sum_{i=1}^{N}\frac{\partial g_q}{\partial x_i}\frac{\partial \hat{x}_i[n+k]}{\partial u_m[n]}\right]_{\substack{\mathbf{x}=\hat{\mathbf{x}}[n+k]\\ g_q=g_q(\hat{\mathbf{x}}[n+k],,\mathbf{u}[n],)\\ u_r=u_r[n]}}
\end{aligned}
\tag{34}
$$

$\dfrac{\partial \hat{x}_i[n+k]}{\partial u_r[n]}$ term is calculated as [10]:

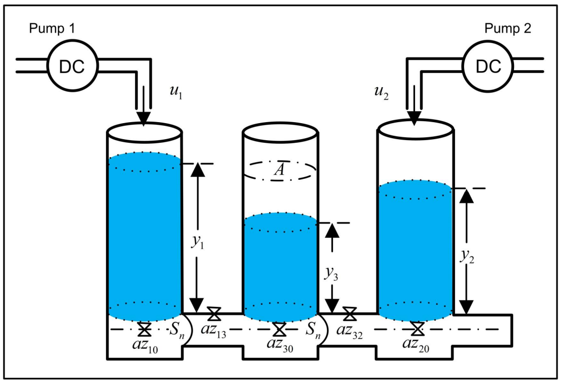

Fig. 3. Three Tank System [10, 16, 17, 22, 23].

$$
\begin{aligned}
\frac{\partial \hat{x}_i[n+k]}{\partial u_r[n]} &= \frac{\partial \hat{x}_i[n+k-1]}{\partial u_r[n]}+\frac{1}{6}\frac{\partial \mathrm{K}_{1\mathrm{X}_i}[n+k-1]}{\partial u_r[n]}+\frac{2}{6}\frac{\partial \mathrm{K}_{2\mathrm{X}_i}[n+k-1]}{\partial u_r[n]}\\
&\quad+\frac{2}{6}\frac{\partial \mathrm{K}_{3\mathrm{X}_i}[n+k-1]}{\partial u_r[n]}+\frac{1}{6}\frac{\partial \mathrm{K}_{4\mathrm{X}_i}[n+k-1]}{\partial u_r[n]}
\end{aligned}
\tag{35}
$$

$$
\begin{aligned}
\frac{\partial \mathrm{K}_{1\mathrm{X}_i}[n+k-1]}{\partial u_r[n]} &= T_s\left[\frac{\partial f_i}{\partial u_r}+\sum_{j=1}^{N}\frac{\partial f_i}{\partial x_j}\frac{\partial \hat{x}_j[n+k-1]}{\partial u_r[n]}\right]_{\mathbf{x}=\hat{\mathbf{x}}[n+k-1]}\\
\frac{\partial \mathrm{K}_{2\mathrm{X}_i}[n+k-1]}{\partial u_r[n]} &= T_s\left[\frac{\partial f_i}{\partial u_r}+\sum_{j=1}^{N}\frac{\partial f_i}{\partial x_j}\left(\frac{\partial \hat{x}_j[n+k-1]}{\partial u_r[n]}+\frac{1}{2}\frac{\partial \mathrm{K}_{1\mathrm{X}_j}[n+k-1]}{\partial u_r[n]}\right)\right]_{\mathbf{x}=\hat{\mathbf{x}}[n+k-1]+\frac{1}{2}\mathbf{K}_{1\mathbf{X}}[n+k-1]}\\
\frac{\partial \mathrm{K}_{3\mathrm{X}_i}[n+k-1]}{\partial u_r[n]} &= T_s\left[\frac{\partial f_i}{\partial u_r}+\sum_{j=1}^{N}\frac{\partial f_i}{\partial x_j}\left(\frac{\partial \hat{x}_j[n+k-1]}{\partial u_r[n]}+\frac{1}{2}\frac{\partial \mathrm{K}_{2\mathrm{X}_j}[n+k-1]}{\partial u_r[n]}\right)\right]_{\mathbf{x}=\hat{\mathbf{x}}[n+k-1]+\frac{1}{2}\mathbf{K}_{2\mathbf{X}}[n+k-1]}\\
\frac{\partial \mathrm{K}_{4\mathrm{X}_i}[n+k-1]}{\partial u_r[n]} &= T_s\left[\frac{\partial f_i}{\partial u_r}+\sum_{j=1}^{N}\frac{\partial f_i}{\partial x_j}\left(\frac{\partial \hat{x}_j[n+k-1]}{\partial u_r[n]}+\frac{\partial \mathrm{K}_{3\mathrm{X}_j}[n+k-1]}{\partial u_r[n]}\right)\right]_{\mathbf{x}=\hat{\mathbf{x}}[n+k-1]+\mathbf{K}_{3\mathbf{X}}[n+k-1]}
\end{aligned}
\tag{36}
$$

Thus, all derivations required to attain $\mathbf{G}[n]$ matrix for the computation of $\mathbf{u}[n]$ by RK$_{\text{NARMA-L2}}$ controller are given in (7)-(10) and (34)-(36). When NARMA controller is derived, some high order terms are neglected in Taylor expansion. Therefore, in order to minimize the tracking error resulting from linearization, a correction term for the control input ($\boldsymbol{\delta}\mathbf{u}[n]$) is summed up with RK$_{\text{NARMA-L2}}$ controller output. To compute $\boldsymbol{\delta}\mathbf{u}[n]$, objective function given in (37) is minimized.

$$
F\left(\mathbf{u}[n],\mathbf{e}_q\right)=\sum_{q=1}^{Q}\sum_{k=1}^{K}\left[e_q[n+k]\right]^2+\sum_{r=1}^{R}\lambda_r\left[u_r[n]-u_r[n-1]\right]^2 \tag{37}
$$

Here, $e_q[n+k]=r_q[n+k]-\hat{y}_q[n+k]$. By considering a small deviation as $\boldsymbol{\delta}\mathbf{u}[n]$ from control signal, the objective function in (37) can be expanded via Taylor expansion:

$$
F\left(\mathbf{u}[n]+\delta\mathbf{u}[n]\right)\cong F\left(\mathbf{u}[n]\right)+\frac{\partial F\left(\mathbf{u}[n]\right)}{\partial \mathbf{u}[n]}\delta\mathbf{u}[n]+\frac{1}{2}\frac{\partial^2 F\left(\mathbf{u}[n]\right)}{\partial^2 \mathbf{u}[n]}\left(\delta\mathbf{u}[n]\right)^2 \tag{38}
$$

Using first order optimality conditions, it is possible to acquire correction term as follows:

$$
\begin{aligned}
\frac{\partial F\left(\mathbf{u}[n]+\delta\mathbf{u}[n]\right)}{\partial \delta\mathbf{u}[n]} &\cong \frac{\partial F\left(\mathbf{u}[n]\right)}{\partial \mathbf{u}[n]}+\frac{\partial^2 F\left(\mathbf{u}[n]\right)}{\partial^2 \mathbf{u}[n]}\delta\mathbf{u}[n]=0\\
\delta\mathbf{u}[n] &= \frac{-\dfrac{\partial F\left(\mathbf{u}[n]\right)}{\partial \mathbf{u}[n]}}{\dfrac{\partial^2 F\left(\mathbf{u}[n]\right)}{\partial^2 \mathbf{u}[n]}}
\end{aligned}
\tag{39}
$$

By approximating the Hessian as [10]

$$
\frac{\partial F\left(\mathbf{u}[n]\right)}{\partial \mathbf{u}[n]}=2\mathbf{J}_m^{\mathrm{T}}\hat{\mathbf{e}},\ \frac{\partial^2 F\left(\mathbf{u}[n]\right)}{\partial^2 \mathbf{u}[n]}=2\mathbf{J}_m^{\mathrm{T}}\mathbf{J}_m \tag{40}
$$

$\boldsymbol{\delta}\mathbf{u}[n]$ is calculated as [10]

$$
\boldsymbol{\delta}\mathbf{u}[n]=-\left[\mathbf{J}_m^{\mathrm{T}}\mathbf{J}_m\right]^{-1}\mathbf{J}_m^{\mathrm{T}}\hat{\mathbf{e}} \tag{41}
$$

where $\hat{\mathbf{e}}$ is

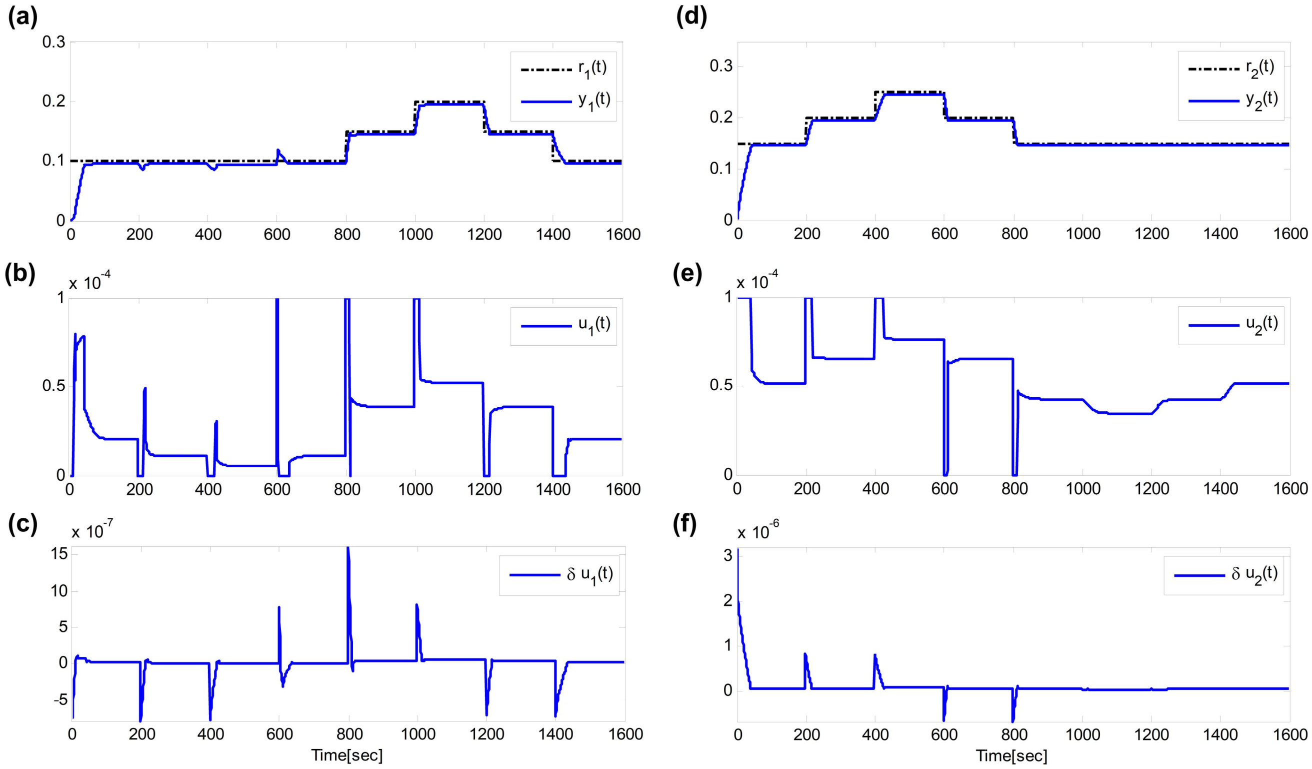

Fig. 4. (**a**,**d**) Outputs of the system, (**b**,**e**) control inputs and (**c**,**f**) correction terms for the nominal case.

$$\hat{\mathbf{e}}=\begin{bmatrix}\hat{e}_1[n+1]\\ \vdots\\ \hat{e}_Q[n+K]\\ \sqrt{\lambda_1}\ \Delta u_1[n]\\ \vdots\\ \sqrt{\lambda_R}\ \Delta u_R[n]\end{bmatrix}=\begin{bmatrix}r_1[n+1]-\hat{y}_1[n+1]\\ \vdots\\ r_Q[n+K]-\hat{y}_Q[n+K]\\ \sqrt{\lambda_1}\ [u_1[n]-u_1[n-1]]\\ \vdots\\ \sqrt{\lambda_R}\ [u_R[n]-u_R[n-1]]\end{bmatrix} \quad (42)$$

In a nutshell, the suboptimal correction term ($\boldsymbol{\delta}\mathbf{u}[n]$) is summed up to the control signal computed with RK$_{\text{NARMA-L2}}$ controller, optimal control vector ($\mathbf{u}^*[n]=\mathbf{u}[n]+\delta\mathbf{u}[n]$) is attained as $\mathbf{u}^*[n]=\mathbf{u}[n]+\delta\mathbf{u}[n]$ [10, 16].

4. Simulation Results

Proposed RK$_{\text{NARMA-L2}}$ controller has been assessed by simulations performed on TTS. Figure 3 illustrates a schematic diagram of TTS which is a benchmark system formed by interconnecting three ideal cylindrical tanks serially [10, 16, 17, 20-23].

Table 1. Parameters for TTS [10, 16, 17, 22, 23].

Symbol	Parameter description	Value
az_{13}	coefficient of outflow (among tank 1 and tank 3)	0.52
az_{32}	coefficient of outflow (among tank 3 and tank 2)	0.55
az_{10}	coefficient of outflow (from tank 1 to the reservoir)	0.26
az_{20}	coefficient outflow (from tank 2 to the reservoir)	0.28
az_{30}	coefficient of outflow (from tank 3 to the reservoir)	0.45
A	cylinderal cross section	0.0154[m^2]
S_n	Cross section of connection pipe n	$5x10^{-5}[m^2]$
g	gravitational constant	$9.81\,[m/s^2]$

Two pumps above the tanks are deployed to pump the water, and valves restrict the flow between tanks [10, 16, 17, 20, 21]. The differential equations given in (43) represent the dynamics of TTS:

$$\begin{aligned}\dot{y}_1(t)&=\frac{1}{A}[u_1(t)-Q_{13}(t)-Q_{10}(t)]\\ \dot{y}_2(t)&=\frac{1}{A}[u_2(t)+Q_{32}(t)-Q_{20}(t)]\\ \dot{y}_3(t)&=\frac{1}{A}[Q_{13}(t)-Q_{32}(t)-Q_{30}(t)]\end{aligned} \quad (43)$$

where;

$$\begin{aligned}Q_{13}(t)&=az_{13}S_n\,\text{sgn}(y_1(t)-y_3(t))\sqrt{2g|y_1(t)-y_3(t)|}\\ Q_{32}(t)&=az_{32}S_n\,\text{sgn}(y_3(t)-y_2(t))\sqrt{2g|y_3(t)-y_2(t)|}\\ Q_{10}(t)&=az_{10}S_n\sqrt{2g|y_1(t)|}\\ Q_{20}(t)&=az_{20}S_n\sqrt{2g|y_2(t)|}\\ Q_{30}(t)&=az_{30}S_n\sqrt{2g|y_3(t)|}\end{aligned} \quad (44)$$

$y_i(t)$ shows height of the liquid, $u_i(t)$ indicates the control signal, $Q_{ji}(t)$ is rate of flow between tanks [10][16],[17],[22]. The description of the symbols in (43),(44) and Figure 2 are given in Table 1. In this context, $y_1(t)$ and $y_2(t)$ represent outputs to be controlled, $u_1(t)$ and $u_2(t)$ are control input signals. $y_3(t)$ is an uncontrollable system dynamic [21].

The control signal limitations of the TTS are given as $u_{1_{\min}}=u_{2_{\min}}=0\,m^3/s$ and $u_{1_{\max}}=u_{2_{\max}}=10^{-4}\,m^3/s$ and Ts=1s in the simulations [10],[16][23]. The RK$_{\text{NARMA-L2}}$ controller performance is compared with RK$_{\text{GPC}}$ for nominal, measurement and parametric uncertainty cases.

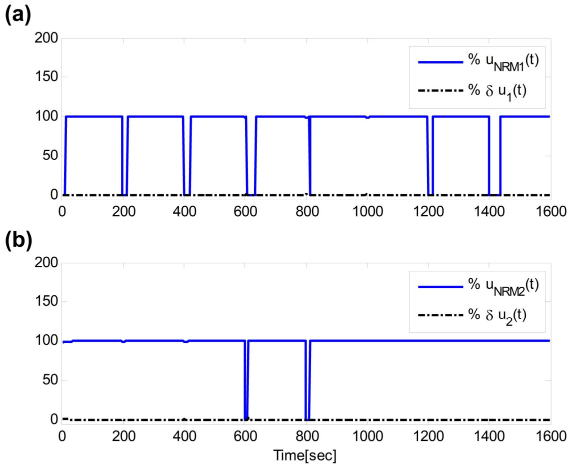

Fig. 5. Percentage of control task share between (**a**) RK$_{\text{NARMA-L2}}$ controller and (**b**) correction terms($\boldsymbol{\delta u}$) for the nominal case.

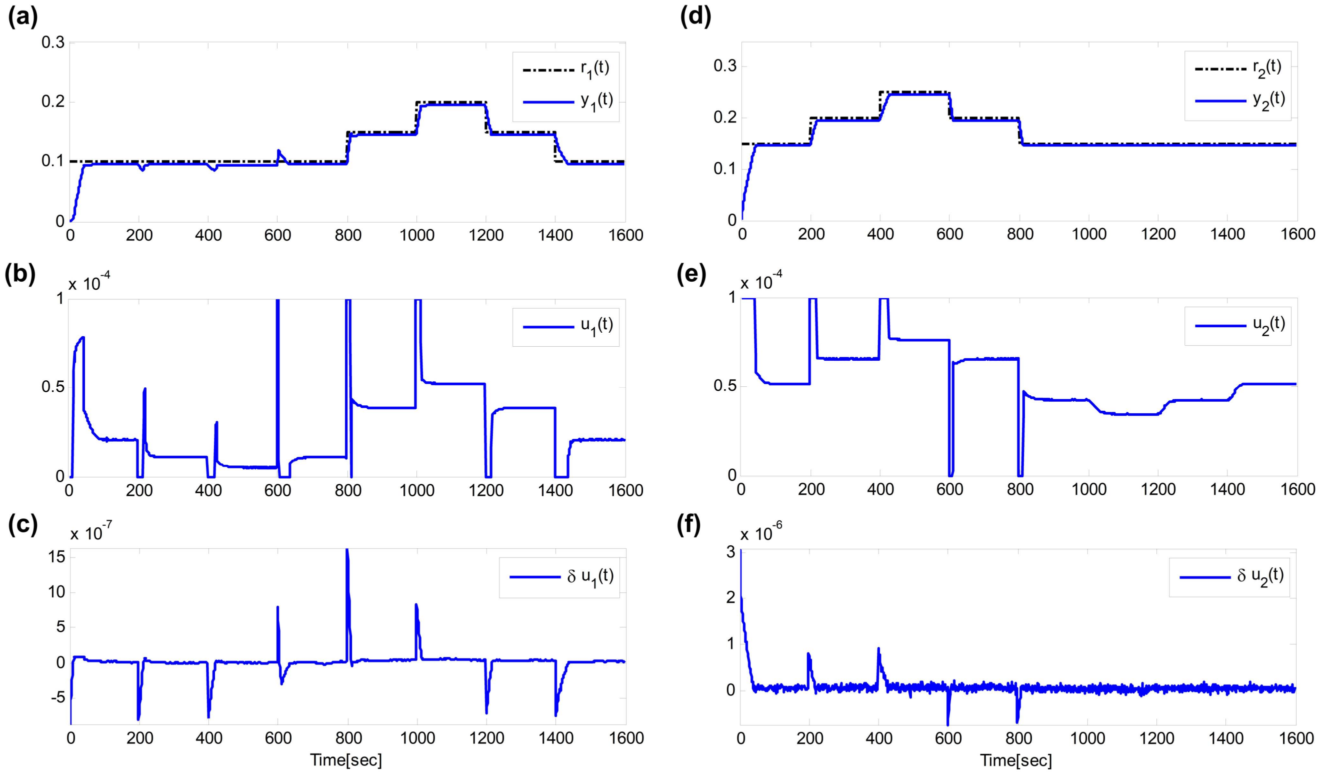

Fig. 6. (**a**,**d**) Outputs of the system, (**b**,**e**) control inputs and (**c**,**f**) correction terms for measurement noise case.

4.1. Nominal Case

The reference tracking performance of RK$_{\text{NARMA-L2}}$ is given in Figure 4 (a,d). Control inputs and correction terms computed by RK$_{\text{NARMA-L2}}$ are also depicted in Figure 4 (b,c,e,f). It is clearly seen from Figure 4 that RK$_{\text{NARMA-L2}}$ controller provides small tracking errors for controlled outputs. The impact percentage of the RK$_{\text{NARMA-L2}}$ controller and correction terms on construction of control signal is illustrated in Figure 5. It can be seen that almost all control task is carried out by RK$_{\text{NARMA-L2}}$ controller.

4.2. Measurement Noise Case

The RK$_{\text{NARMA-L2}}$ controller performance has been assessed for the case where measurement devices are contaminated by measurement noise. Therefore, $y_1(t), y_2(t)$ are incurred measurement noise having standard deviation values $\sigma_{y1}(t) = \sigma_{y2}(t) = 0.03$. Closed –loop performance is depictured in Figure 6. Also, Figure 7 illustrates percentage of control task.

4.3. Parametric Uncertainty

By considering the outflow parameter as the parameter with uncertainty given as $az_{13}(t) = 0.52 + 0.28\sin(0.0133\,\pi t)$, prediction achievement of RK$_{\text{estimator}}$ subblock in RK$_{\text{model}}$,and tracking ability of RK$_{\text{NARMA-L2}}$ controller can be examined as given in Figure 8. The convergence of the uncertain parameter to its actual value is illustrated in Figure 8(g). The control task share is given in Figure 9.

4.4. Computation Time

In this section, the computational load of the algorithms has been examined. For this purpose, the total computation times of the algorithm have been illustrated for the mentioned three different cases in Figure 10. In Figure 10 (a,b,c), the computation times for

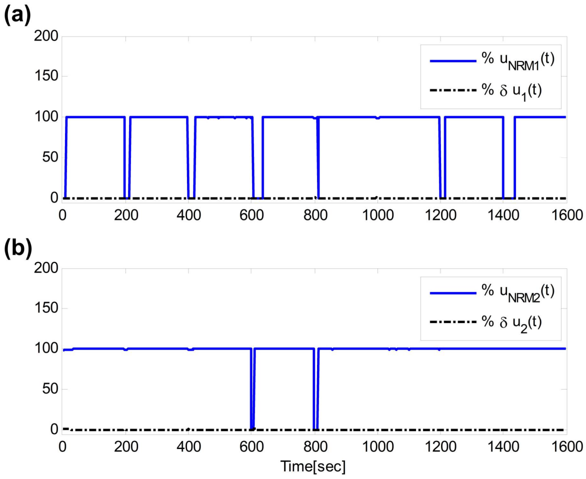

Fig. 7. Percentage of control task share between (**a**)RK$_{\text{NARMA-L2}}$ controller and (**b**)correction terms ($\boldsymbol{\delta u}$) for measurement noise case.

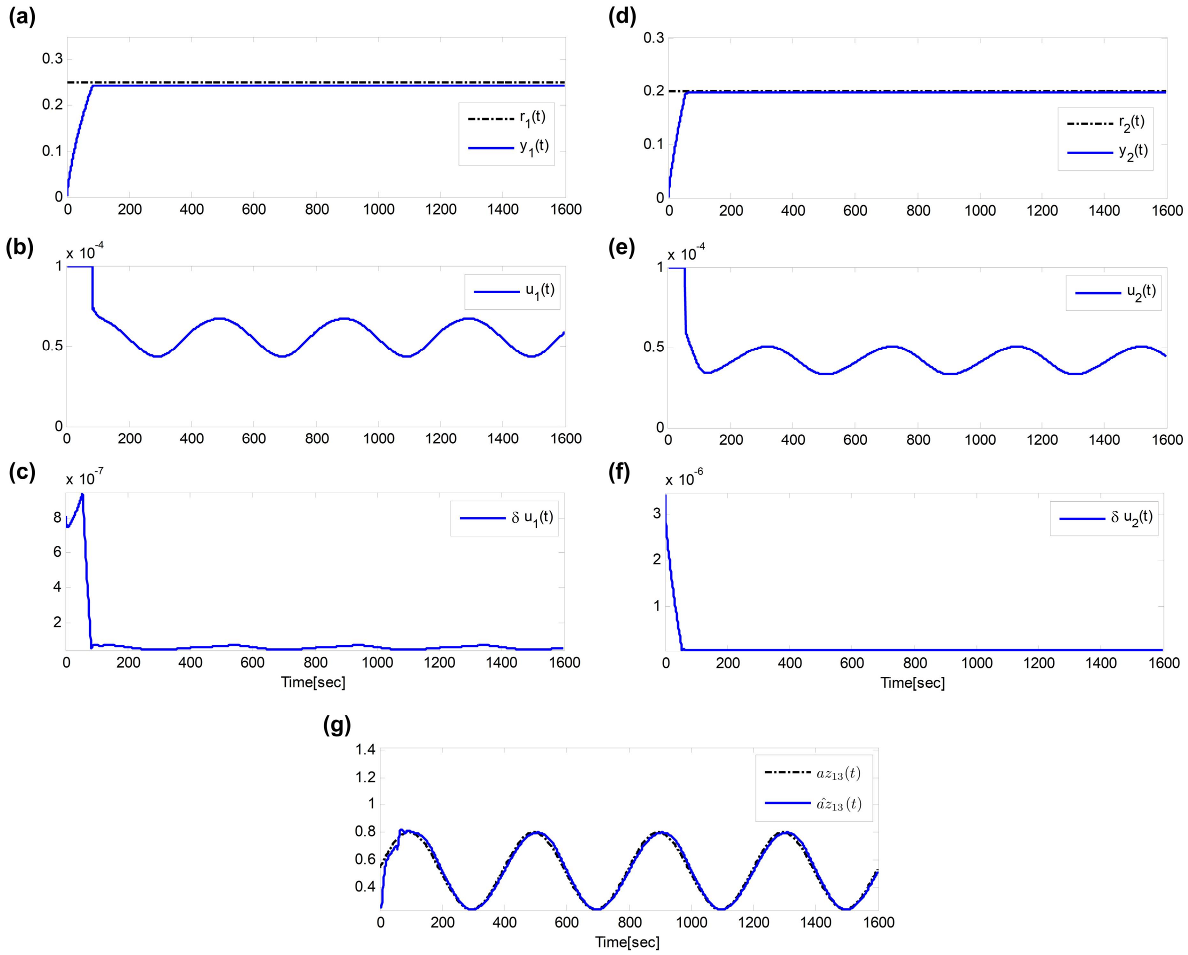

Fig. 8. (**a**,**d**) Outputs of the system, (**b**,**e**) control inputs, (**c**,**f**) correction terms ,(**g**) uncertain parameter $az_{13(t)}$ and its estimation for parametric uncertainty case.

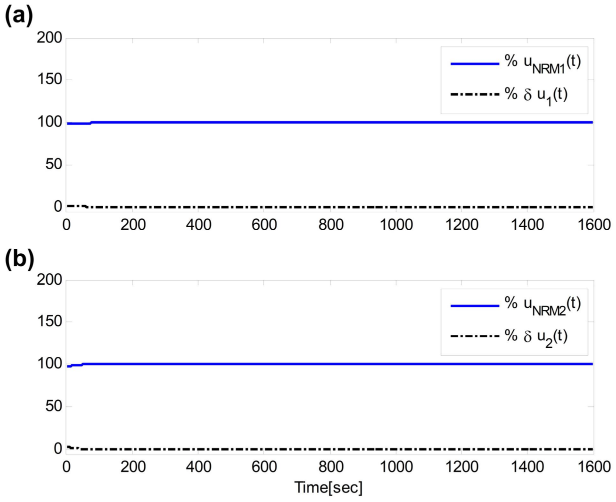

Fig. 9. Percentage of control task share between (**a**)RK$_{\text{NARMA-L2}}$ controller and (**b**)correction terms ($\mathbf{\delta u}$) for parametric uncertainty

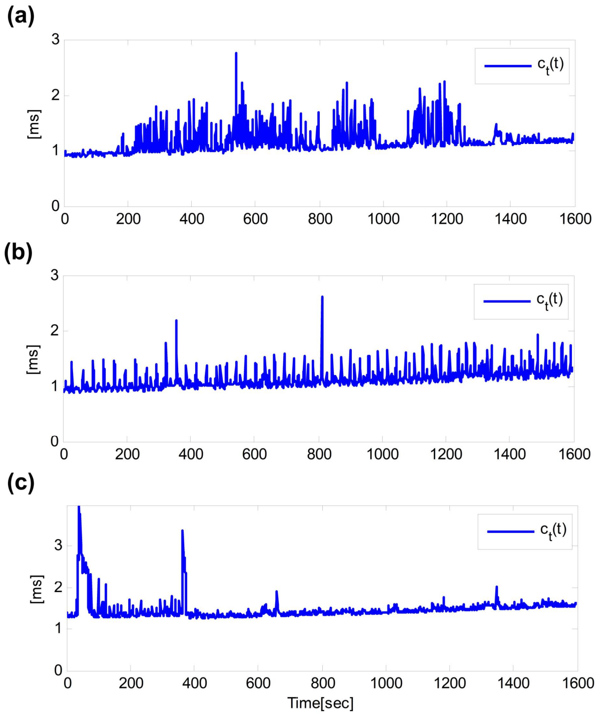

Fig. 10. Computation times[ms] for (**a**) nominal, (**b**) measurement noise and (**c**) parametric uncertainty cases.

nominal case, and for the cases where measurement noise and parametric uncertainty are implemented are illustrated respectively. Maximum computational time is less than 5 ms. Thus, controller is convenient for real time applications.

4.5. Comparison with Runge-Kutta based GPC

The performance of the controller has been compared with Runge-Kutta GPC(RK$_{\text{GPC}}$) proposed in [10] to provide a basis for evaluation. Since RK$_{\text{NARMA-L2}}$ controller is constructed via one-step ahead model behavior, and to carry out a fair evaluation, the prediction horizon for predictive mechanism is chosen as “1”.

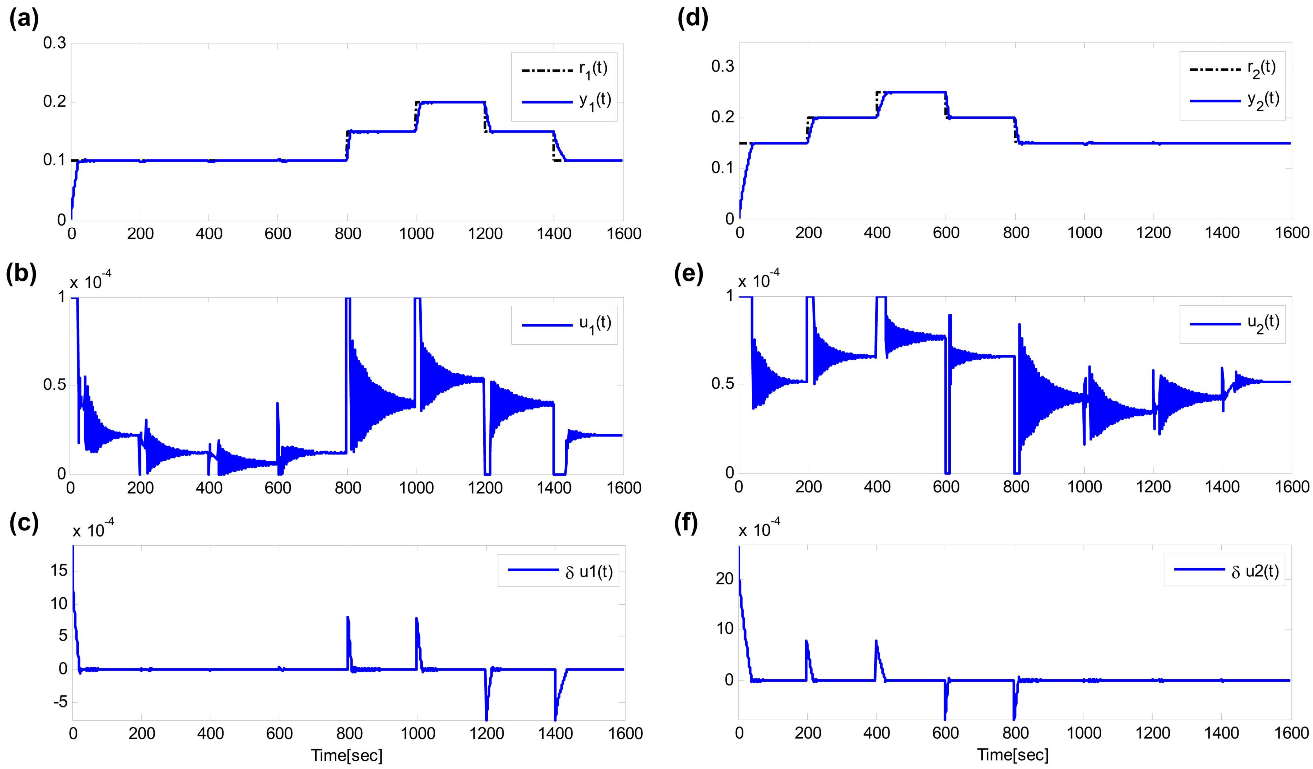

Fig. 11. (**a**,**d**) Outputs of the system, (**b**,**e**) control inputs and (**c**,**f**) correction terms for nominal case(RK$_{\mathrm{GPC}}$).

The tracking performance of RK$_{\mathrm{GPC}}$ is depicted in Figure 11 for nominal condition. The performance of the RK$_{\mathrm{GPC}}$ for measurement noise and parametric uncertainty cases are illustrated in Figure 12 and 13, respectively. The magnitude of chattering control signals is larger in RK$_{\mathrm{GPC}}$ compared to RK$_{\mathrm{NARMA\text{-}L2}}$ This chattering can be suppressed by increasing the prediction horizon. The following performance index function is deployed to constitute Table 2 for performance comparison.

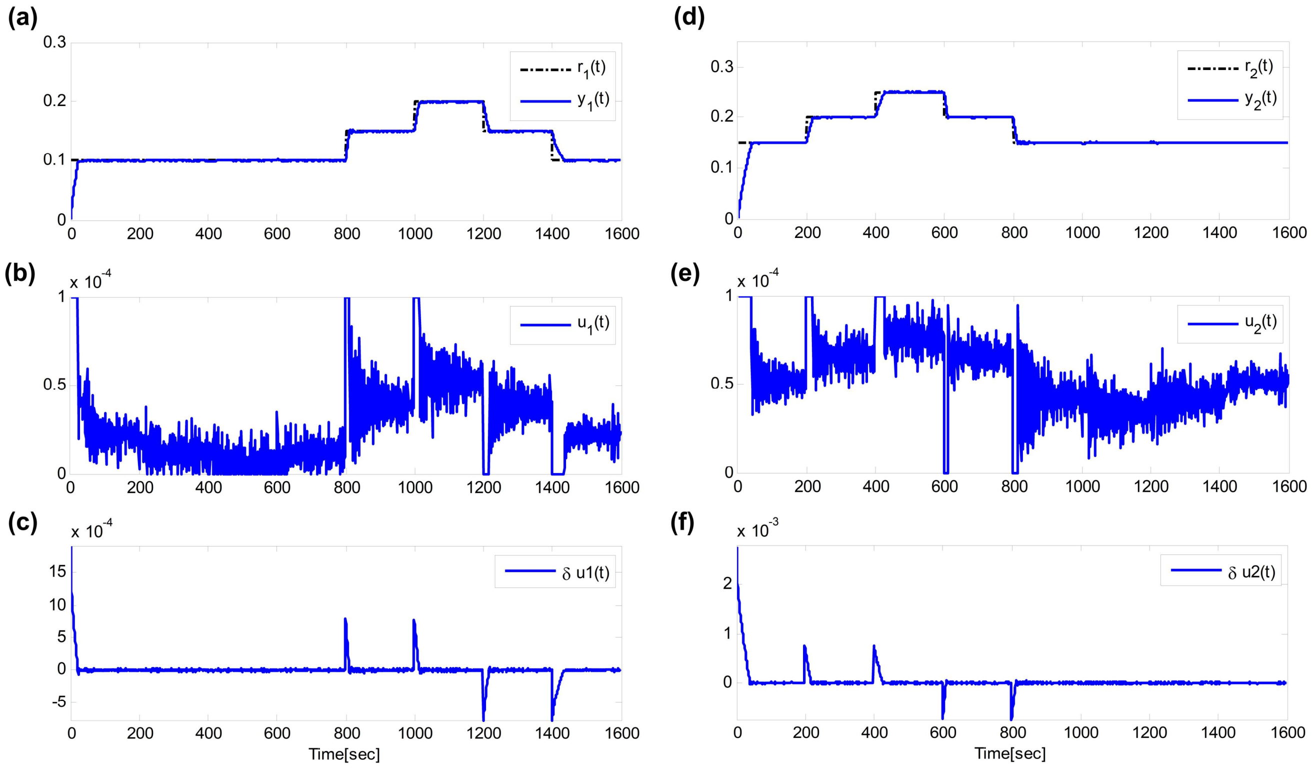

Fig. 12. (**a**,**d**) Outputs of the system, (**b**,**e**) control inputs and (**c**,**f**) correction terms for measurement noise case(RK$_{\mathrm{GPC}}$).

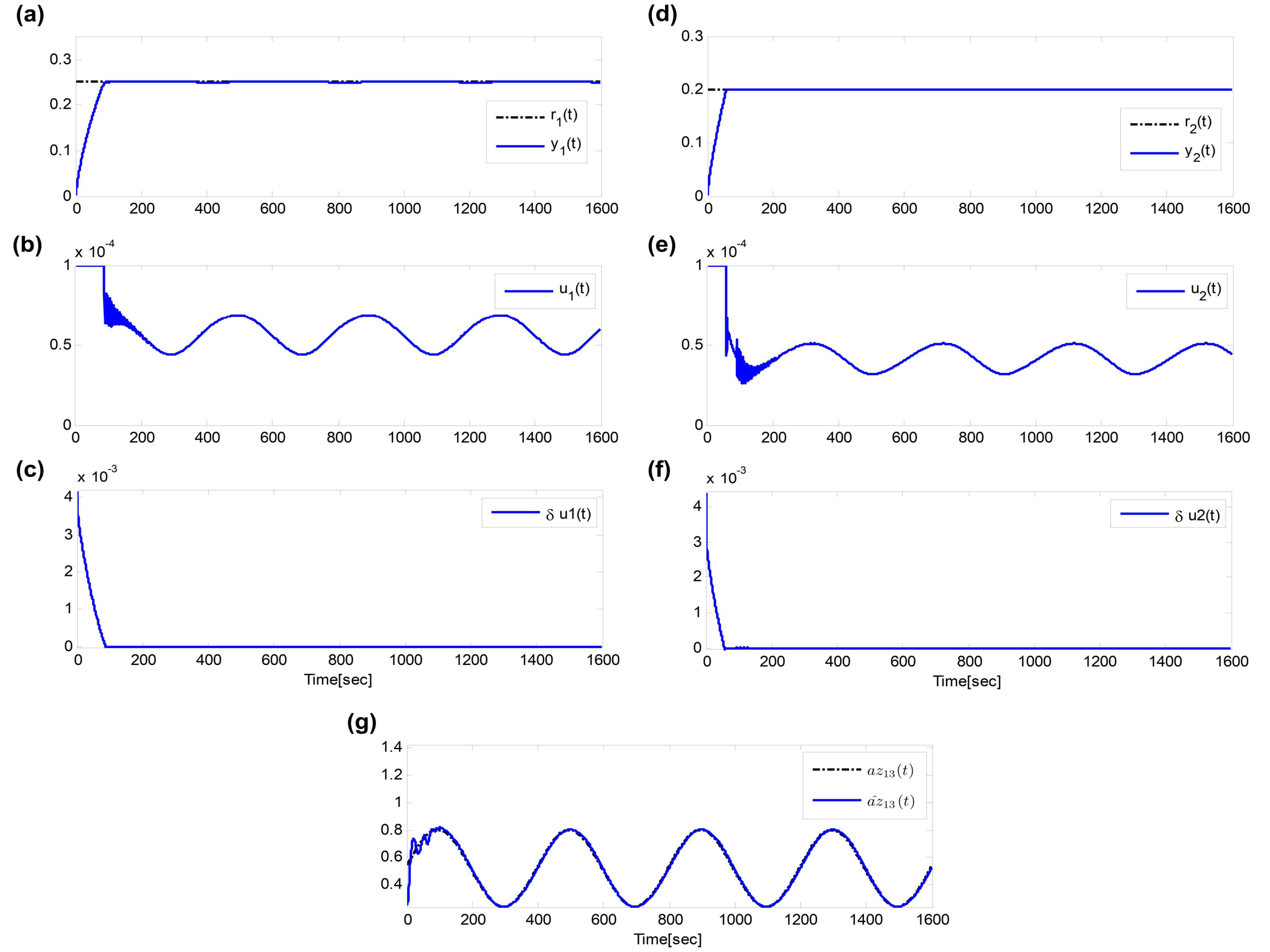

Fig. 13. (**a**,**d**) Outputs of the system, (**b**,**e**) control inputs, (**c**,**f**) correction terms ,(**e**) uncertain parameter $az_{13(t)}$ and its estimation for parametric uncertainty case(RK$_{GPC}$).

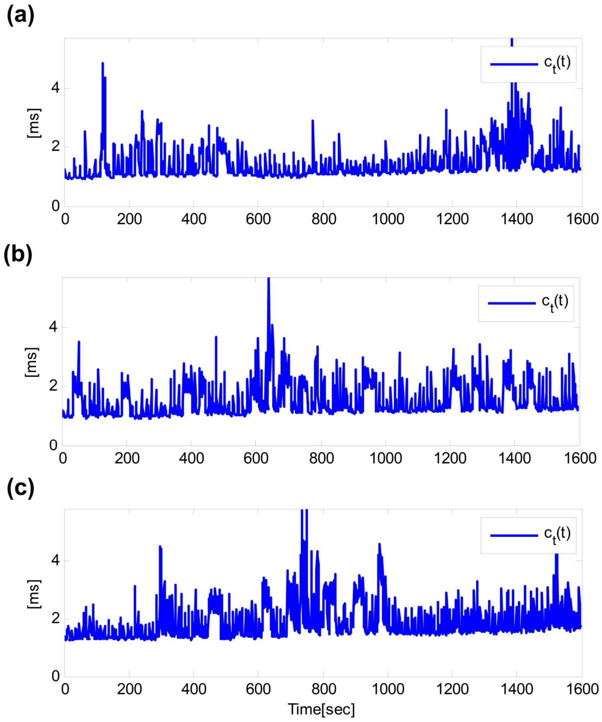

Fig. 14. Computation times[ms] for (**a**) nominal, (**b**) measurement noise and (**c**) parametric uncertainty cases (RK$_{GPC}$).

$$\mathbf{J}=\sum_{n=2}^{\infty}\left(\frac{\mathbf{r}[n]-\mathbf{y}[n]}{\mathbf{r}[n]}\right)^2+\left(\frac{\mathbf{u}[n]-\mathbf{u}[n-1]}{\mathbf{u}_{\max}}\right)^2+\left(\frac{az_{13}[n]-a\hat{z}_{13}[n]}{az_{13}[n]}\right)^2 \quad (45)$$

Table 2. Performance Comparison via (45).

Cases\Controller	RK$_{NARMA-L2}$	RK$_{GPC}$
Nominal	216.9048	220.1724
Measurement Noise	216.8309	244.0223
Parametric Uncertainty	609.0902	627.7495

As can be seen from Table 2, RK$_{NARMA-L2}$ has better performance than RK$_{GPC}$. It is observed that RK$_{NARMA-L2}$ is 1.4841 %, 11.1430 % and 2.9724 % better than RK$_{GPC}$ for nominal, measurement noise and parametric uncertainty cases, respectively. In order to evaluate the applicability of the control algorithms, the computation times are illustrated in Figure 10 and Figure 14 for RK$_{NARMA-L2}$ and RK$_{GPC}$ respectively. Table 3 is obtained by considering the maximum computation times of RK$_{NARMA-L2}$ and RK$_{GPC}$ for each case.

As can be seen from Figures 4,6,8, RK$_{NARMA-L2}$ controller has small steady state error owing to the negligence of high order terms in Taylor expansion. RK$_{NARMA-L2}$ controller has no chattering in comparison with RK$_{GPC}$. Chattering in RK$_{GPC}$ can be enhanced by increasing prediction horizon. However, the prediction horizon was chosen as “1” to avoid injustice to $RK_{NARMA-L2}$ controller and to provide a fair comparison.

Table 3. Computation times [ms] for $RK_{NARMA-L2}$ and RK_{GPC}.

Cases\Controller	$RK_{NARMA-L2}$	RK_{GPC}
Nominal	2.7561	5.7146
Measurement Noise	2.5959	5.7053
Parametric Uncertainty	3.9749	5.8008

5. Conclusion

In this study, a new control architecture with a $RK_{NARMA-L2}$ controller is proposed for MIMO and nonlinear systems. Control performance is evaluated for three different operating conditions on TTS. RK_{GPC} is utilized to evaluate the control performance of the introduced controller. The computational loads of the control algorithms have also been evaluated to justify the real time applicability of the methods. The attained results and computational loads indicate that it is possible to employ the proposed $RK_{NARMA-L2}$ control architecture successfully.

It is intended to work on stable and adaptive control methods derived via Runge-Kutta identification method in future studies.

References

[1] M. Majstorovic, I. Nikolic, J. Radovic, and G. Kvascev, “Neural network control approach for a two-tank system,” *2008 9th Symp. Neural Netw. Appl. Electr. Eng.*, vol. 2, no. 1, pp. 2–5, 2008.

[2] J. O. Pedro, O. T. C. Nyandoro, and S. John, “Neural network based feedback linearisation slip control of an anti-lock braking system,” *Proc. 2009 7th Asian Control Conf. ASCC 2009*, no. Lmi, pp. 1251–1257, 2009.

[3] O. De Jesus, A. Pukrittayakamee, and M. Hagan, “A comparison of neural network control algorithms,” in *International Joint Conference on Neural Networks*, 2001, vol. 1, pp. 521–526.

[4] M. T. Hagan, H. B. Demuth, and O. De Jesus, “An introduction to the use of neural networks in control systems,” *Int. J. Robust Nonlinear Control*, vol. 12, no. 11, pp. 959–985, Sep. 2002.

[5] A. Pukrittayakamee, O. De Jesus, and M. T. Hagan, “Smoothing the Control Action for NARMA-L2 Controllers,” in *45th Midwest Symposium on Circuits and Systems*, 2002, vol. 3, pp. 37–40.

[6] Wahyudi, S. S. Mokri, and A. A. Shafie, “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*, 2008, pp. 691–697.

[7] A. Akbarimajd and S. Kia, “NARMA-L2 Controller for 2-DoF Underactuated Planar Manipulator,” in *International Conference on Control, Automation, Robotics and Vision (ICARCV 2010)*, 2010, no. December, pp. 195–200.

[8] T. Vesselenyi, S. Dzițac, I. Dzițac, and M.-J. Manolescu, “Fuzzy and Neural Controllers for a Pneumatic Actuator,” *Int. J. Comput. Commun. Control*, vol. 2, no. 4, p. 375, 2007.

[9] K. S. Narendra and S. Mukhopadhyay, “Adaptive control using neural networks and approximate models,” *IEEE Trans. Neural Networks*, vol. 8, no. 3, pp. 475–85, Jan. 1997.

[10] S. Iplikci, “Runge-Kutta model-based adaptive predictive control mechanism for non-linear processes,” *Trans. Inst. Meas. Control*, vol. 35, no. 2, pp. 166–180, 2013.

[11] K. Srakaew, J. Kananai, and R. Chancharoen, “Trajectory Control of a Nonlinear Dynamical System using NARMA L2 Neurocontroller,” *J. Comput. Inf. Technol. Comput. Inf. Technol.*

[12] K. Uçak and G. Ö. Günel, “A Novel Adaptive NARMA-L2 Controller Based on Online Support Vector Regression for Nonlinear Systems,” *Neural Process. Lett.*, vol. 44, no. 3, pp. 857–886, 2016.

[13] K. Uçak, and G. Ö. Günel, “Online Support Vector Regression Based Adaptive NARMA-L2 Controller for Nonlinear Systems,” *Neural Processing Letters,* vol. 53, pp. 405–428, 2021.

[14] X. Yuan, Y. Wang, and L. Wu, “Composite feedforward-feedback controller for generator excitation system,” *Nonlinear Dyn.*, vol. 54, no. 4, pp. 355–364, Jan. 2008.

[15] M. Majstorovic, I. Nikolic, J. Radovic, and G. Kvascev, “Neural network control approach for a two-tank system,” in *Symposium on Neural Network Applications in Electrical Engineering*, 2008, vol. 2, no. 1, pp. 203–206.

[16] K. Uçak, “A Runge – Kutta neural network-based control method for nonlinear MIMO systems,” *Soft Comput.*, vol. 23, no. 17, pp. 7769–7803, 2019.

[17] M. Cetin and S. Iplikci, “A novel auto-tuning PID control mechanism for nonlinear systems,” *ISA Trans.*, vol. 58, pp. 292–308, 2015.

[18] S. Beyhan, “Runge-Kutta model-based nonlinear observer for synchronization and control of chaotic systems,” *ISA Trans.*, vol. 52, no. 4, pp. 501–509, 2013.

[19] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. The MIT Press, 2005.

[20] S. Iplikci, “A support vector machine based control application to the experimental three-tank system,” *ISA Trans.*, vol. 49, no. 3, pp. 376–386, 2010.

[21] D. Theilliol, H. Noura, and J.-C. Ponsart, “Fault diagnosis and accommodation of a three-tank system based on analytical redundancy,” *ISA Trans.*, vol. 41, no. 3, pp. 365–382, 2002.

[22] A. Gmbh, *DTS200 Laboratory Setup Three - Tank - System*. amira GmbH, 2000.

[23] K. Uçak, “A Novel Model Predictive Runge–Kutta Neural Network Controller for Nonlinear MIMO Systems,” *Neural Processing Letters,* vol. 51, pp. 1789–1833, 2020.