A novel adaptive Runge–Kutta controller for nonlinear dynamical systems

A novel adaptive Runge–Kutta controller for nonlinear dynamical systems

Cite this article as: Kemal Uçak, A novel adaptive Runge–Kutta controller for nonlinear dynamical systems, Soft Computing https://doi.org/10.1007/s00500-021-05792-4

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

A novel adaptive Runge-Kutta controller for nonlinear dynamical systems

Kemal UÇAK

Received: / Accepted:

Abstract This paper introduces a new Runge-Kutta (RK) Integration based adaptive controller by considering control law as an ODE for nonlinear MIMO systems. It is aimed to derive a novel adaptive controller by regarding the control law as an ODE with limited information about control law structure. Adaptive parameters are adjusted via an RK predictive system model where Levenberg-Marquardt (LM) technique is deployed. The adjustment mechanism enables to utilize RK both in adaptive controller and system model. The performance evaluation has been delved into on Van de Vusse(VdV) system for diverse situations, and reasonable results have been acquired for introduced adaptation mechanism.

Keywords Adaptive Runge-Kutta controller · Model predictive control · Model predictive Runge-Kutta controller · Runge-Kutta integration

1 Introduction

The quote that "Mathematics is the language in which God has written the universe" atributted to Galileo Galilei is possibly the best apothegm that describes the importance of Mathematics in our life to date. Dynamics expressing events can be defined by differentiation and integration in calculus. While differentiation is utilized to examine how one dynamic alters with respect to another dynamic, integration is appealed to evaluate the cumulative impact of small parts/elements on the whole. Therefore, integration is one of the main branches of calculus (Bittinger et al. 2001). Integration has many real life applications from calculation of Greek quadrature of the circle to analysis of complex nonlinear control systems. In most cases, it is difficult to integrate complicated nonlinear functions analytically. Therefore, over the centuries, in order to approximate and find the numerical value of an integral, many numerical integration techniques have been contrived, dating back to antiquity particularly since the sixteenth century (Davis and Rabinowitz 1984).

Among numerical integration methods, Runge-Kutta(RK) techniques, the name of which comes from Carl David Tolmé Runge (1856-1927) and Martin Wilhelm Kutta (1867-1944) who first studied the technique around 1900(Fasshauer 2020; Roberts 2010), are the most prominent ordinary differential equation(ODE) solver. In spite of being a century old method, it is still frequently deployed to estimate the future behaviour of the complex nonlinear systems numerically.

In a control system, especially in a nonlinear multi input multi output (MIMO) control system, the nonlinear behaviour characteristic and also interaction among dynamics obstruct the approximation and control of the system. Therefore, it is required to employ a controller which can attune the excited unpredictable dynamics with adaptation ability. This circumstance necessitates to deploy adaptive nonlinear MIMO controller architecures in order to ingender the nonlinear dynamics as desired despite nonlinearity and interactions.

There exist a great variety of intelligent adaptation methodologies such as ANN(Saerens and Soquet 1991; Zhang et al. 1995; Tanomaru and Omatu 1992; Psaltis et al. 1988; Efe 2011; Hagan et al. 2002), Fuzzy Logic (Pham and Karaboga 1999; Sharkawy 2010; Bouallègue et al. 2012), ANFIS(Bishr et al.

Kemal UÇAK (✉)
Muğla Sıtkı Koçman University Kötekli 48000, Muğla, Turkey
[1]E-mail: ucak@mu.edu.tr
[2]E-mail: ucakk@mit.edu

2000; Denai et al. 2004) and SVR (Uçak and Günel 2016; Uçak and Günel 2017; Iplikci 2010a; Iplikci 2006). Occasionally, the heavy computational load of the mentioned methods may restrict their ability to be deployed in real-time control architectures. Since precision and computational complexity of modeling techniques are two crucial quiddity in execution of the introduced adaptive architecture, adjustment mechanisms that possess lower computational complexity and advanced precision are more feasible and preferable to realise. Therefore, the RK based identification with low computational load in comparison with soft computing methods, given in (Iplikci 2013; Uçak 2019) is employed to identify nonlinear system dynamics.

A variety of controller architectures based on RK-system model have been introduced. Iplikci (Iplikci 2013) has introduced a model predictive controller(MPC) based on RK model to identify the dynamics of controlled nonlinear MIMO systems. In MPC problem, in order to adjust the control signal vector that compels the system dynamics to follow the reference, it is essential to forecast the possible emerging system behaviour against adjustment to be realized in control vector. The learning rules to update control signal vector are acquired by means of the Taylor series expansion of the objective function. Thus, in order to utilize the derived adjustment laws effectively, it is required to estimate the system Jacobians using system model. Çetin et al (Cetin and Iplikci 2015) deployed the predictive RK system model to derive adjustment rules for an adaptive MIMO PID controller. The proposed auto-tuning mechanism for MIMO PID combines the robustness and fast convergence features of PID and MPC(Cetin and Iplikci 2015). Beyhan (Beyhan 2013) introduced a nonlinear observer which aims to update the system states by using predictive RK model introduced in (Iplikci 2013).

This study introduces a novel adaptive controller where RK integration is directly utilized to construct an adaptive control law. To the best of the author's knowledge such direct implementation of RK integration as a direct control method is not presented in technical literature. By regarding the control signal as an ODE set, firstly, the Runge-Kutta control law is derived, and then the essential information so as to annex adaptability to control law is examined via Levenberg-Marquardt optimization law. As a result of examination, it has been observed that only knowing the correlation degree of $u(t)$ with itself is enough to evolve adaptive control law. Thus, a novel adaptive RK controller that can be adapted even with such limited information has been proposed.

The most important feature that radically distinguishes this study from previous (Uçak 2019; Uçak 2020) and all other Runge-Kutta based studies given in (Iplikci 2013; Cetin and Iplikci 2015; Beyhan 2013; Efe and Kaynak 2000; Efe and Kaynak 1999; Wang and Lin 1998) is that the Runge-Kutta integration is presented as a Runge-Kutta controller in the controller block for the first time without using any machine learning architecture such as neural network etc so as to store statistical information of control signal or controller parameters.

The adjustment mechanism contains an architecture in which RK integration is deployed as control law and system model. Therefore, the adjustment mechanism is composed of a 4th order RK model in order to observe and estimate the future emerging impact of the obtained control signal on system behaviour, RK controller to form the closed-loop system dynamics as desired and adjustment law utilized to tune controller parameters. The high accuracy and low computational load of RK integration techniques evoked the idea that RK integration technique could be used not only to estimate the system model but also derive adaptive control law. The nonlinear system dynamics are approximated via the RK based modeling technique introduced by Iplikci (Iplikci 2013) due to its precision and low execution time. In this study, the main contribution is to introduce a nonlinear RK MIMO controller which indicates the utility of the ordinary differential equation(ODE) solvers as adaptation mechanism for adaptive control theory. The performance evaluation has been performed on nonlinear VdV systems for diverse situations. The results verify that prosperous closed-loop control and identification performances have been attained for the introduced adjustment mechanism and RK model introduced by Iplikci (Iplikci 2013), respectively.

In section 2, adaptive mechanism for RK controller is presented. The adjustment rules and control algorithm for proposed RK controller are detailed in Section 3. The evaluation of the introduced adaptive RK controller is scrutinised on a nonlinear VdV system in Section 4. The controller performance has been compared with Runge-Kutta model based adaptive MIMO PID controller presented in (Cetin and Iplikci 2015) with respect to tracking performances and computational loads of control algorithms for nominal case and when measurement noise and parametric uncertainty are added. A brief conclusion and future works are given in section 5.

2 Adaptive Runge-Kutta controller

In control systems, approximating the dynamic behavior of the system to be controlled is of great significance for adaptive controller architectures. So as to reform the closed-loop system dynamics as desired, it is essential to inset predictive structure based adaptation ability to the controller parameters so as to attune emerging new circumstances. An effective adaptive control mechanism incorporates an accurate system model and convenient controller parameter adjustment laws derived via optimization theory.

The basic model based(MB) adaptive architecture subsumes control law, system model and adjustment law blocks. The possible system behaviours against adjustment in control parameters are approximated via system model block by applying the control signal firstly to the model. Then the required adaptation rules forcing the system output to the desired reference point are acquired and optimal control law is computed. Accurate representation of system dynamics in system model is crucial for good performance of the adaptive controller in MB adaptive architectures. A great number of adaptive controller architectures can be suggested by incorporating different system models, controller structures and adaptation laws for nonlinear MIMO systems(Aström et al. 1977). Any controller including adjustable parameters can be deployed in the MB adaptive mechanism (Uçak and Günel 2017). RK integration method is directly employed as a controller in this paper. As nonlinear system model, several system identification methods based on artificial intelligence like ANN (Efe 2011; Hagan et al. 2002; Efe and Kaynak 2000; Efe and Kaynak 1999), ANFIS (Denai et al. 2004; Jang 1993), SVR (Iplikci 2010a; Iplikci 2006; Iplikci 2010b) etc have been introduced to learn system dynamics. In the introduced mechanism, the nonlinear system dynamics are identified via RK system model given in (Iplikci 2013) so as to ameliorate model/approximation precision and decrease control signal execution time.

The proposed adaptive RK controller architecture is shown in Figure 1 where R expreses the system input dimension and Q stands for the number of the system outputs to be controlled. The model based architecture incorporates two crucial structures to be scrupulously scrutinised: RK controller to express the controller dynamics and RK model so as to forecast P_H–step ahead system outputs. By considering that the abbreviations of these two main blocks simplify the intelligibility of adjustment mechanism, Runge-Kutta controller is abridged as $RK_{controller}$ and system model is RK_{model} throughout the article. The proposed adaptive control mechanism is composed of three main phases consecutively performed in an online manner: prediction, training and control phases.

2.1 Prediction Phase

The $RK_{controller}$ produces a candidate ($\mathbf{u}^{\star}[n]$) signal as

$$\begin{aligned}\mathbf{u}^{\star}[n] =& \mathbf{u}\big[\mathbf{n-1}\big] + \frac{1}{6}\big[\mathbf{K_{1u}}\big[n-1\big] + 2\mathbf{K_{2u}}\big[n-1\big] \\ &+ 2\mathbf{K_{3u}}\big[n-1\big] + \mathbf{K_{4u}}\big[n-1\big]\big]\end{aligned} \tag{1}$$

in which $\mathbf{K_{1u}}\big[n-1\big], \mathbf{K_{2u}}\big[n-1\big], \mathbf{K_{3u}}\big[n-1\big], \mathbf{K_{4u}}\big[n-1\big]$ denote the slopes of control signal and all slopes are adjustable parameters of the $RK_{controller}$. The parameter vector to be optimized in adjustment mechanism is given as

$$\mathbf{K} = \big[\mathbf{K_{1u}}\ \mathbf{K_{2u}}\ \mathbf{K_{3u}}\ \mathbf{K_{4u}}\big] \tag{2}$$

Then, by sequentially applying the obtained $\mathbf{u}^{\star}[n]$ to RK_{model}, the system behaviour and system Jacobian required to adjust controller parameters($\mathbf{K}$) can be acquired. RK_{model} block contains three main subblocks to predict the nonlinear system dynamics: raw RK system model, RK model based EKF(RK_{EKF}) and RK based model parameter estimator ($RK_{estimator}$) subblocks. In order to deploy raw RK system model effectively and predict system dynamics, the current states of the controlled system and actual values of the deviated system parameters($\boldsymbol{\theta}$) are required. Using the available input–output samples obtained from controlled system, the system states can be attained via RK_{EKF}. Because of the lack of conventional modeling techniques or deviation in system parameters ($\boldsymbol{\theta}$), system parameters ($\boldsymbol{\theta}$) may not be determined accurately and the system identification performance and accuracy of the system model may aggravate. Therefore, $RK_{estimator}$ is utilized to predict the actual values of the unmeasured, uncomputed or deviated system parameters ($\boldsymbol{\theta}$). By using raw RK model, RK_{EKF}, and $RK_{estimator}$, RK_{model} can be constituted to forecast P_H–step future system action with high accuracy. The detailed information about subblock of RK_{model} are given in (Iplikci 2013; Uçak 2019; Uçak 2020).

2.2 Training Phase

The adjustment laws to attain the feasible $RK_{controller}$ parameters can be derived via objective function in (3)

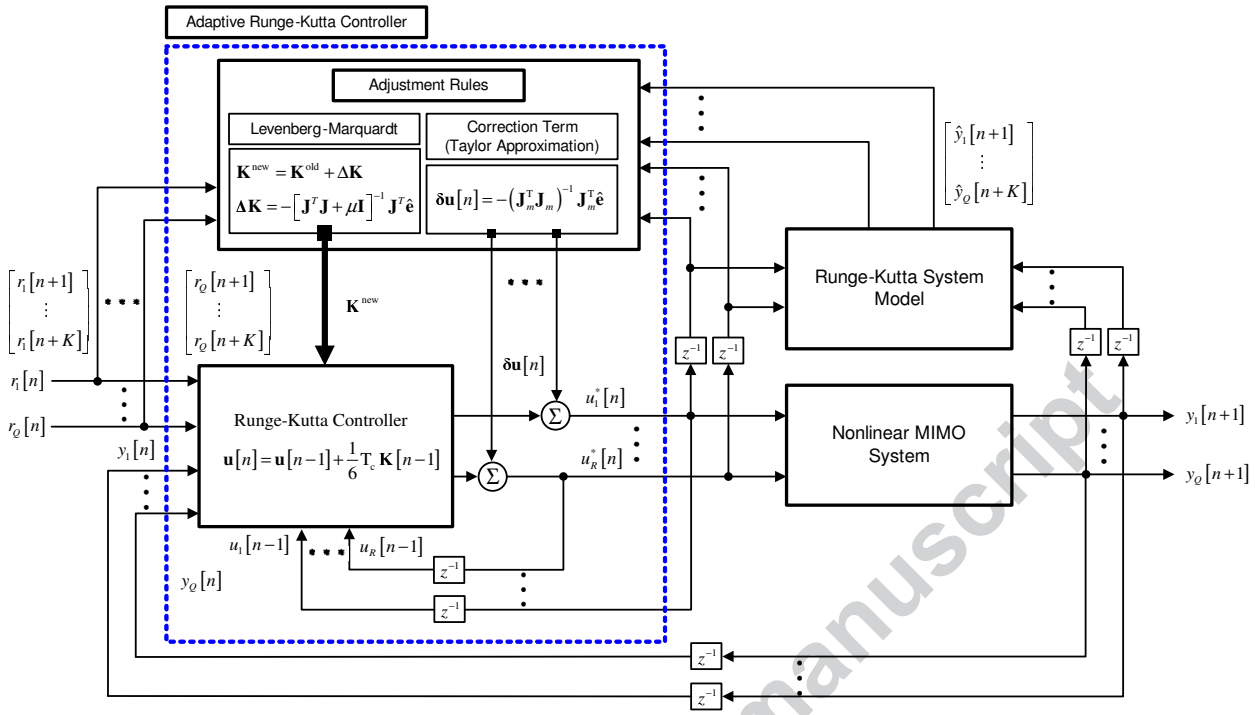

Fig. 1 MP $\mathrm{RK}_{\mathrm{controller}}$ mechanism.

$$F(\mathbf{u}[\mathbf{n}],\hat{\mathbf{e}}_q)=\sum_{q=1}^{Q}\sum_{p=1}^{P_H}\beta_q\left[\hat{e}_q[n+p]\right]^2+\sum_{r=1}^{R}\lambda_r\left[u_r[n]-u_r[n-1]\right]^2 \quad (3)$$

where $\hat{e}_q[n+p]=r_q[n+p]-\hat{y}_q[n+p]$, P_H indicates the prediction horizon, β_q and λ's denote penalty coeficients to hamper chattering in control signals. LM optimization rule can be deployed to optimize the $\mathrm{RK}_{\mathrm{controller}}$ parameters($\mathbf{K}$) as follows:

$$\mathbf{K}^{new}=\mathbf{K}^{old}+\Delta\mathbf{K}\ ,\ \Delta\mathbf{K}=-\left[\mathbf{J}^T\mathbf{J}+\mu\mathbf{I}\right]^{-1}\mathbf{J}^T\hat{\mathbf{e}} \quad (4)$$

where $\mathbf{J}$ is given as

$$\mathbf{J}=\begin{bmatrix}\frac{\partial\hat{e}_1[n+1]}{\partial\mathbf{K_{1u}}[n-1]} & \cdots & \frac{\partial\hat{e}_1[n+1]}{\partial\mathbf{K_{4u}}[n-1]}\\ \vdots & \ddots & \vdots\\ \frac{\partial\hat{e}_Q[n+K]}{\partial\mathbf{K_{1u}}[n-1]} & \cdots & \frac{\partial\hat{e}_Q[n+K]}{\partial\mathbf{K_{4u}}[n-1]}\\ \sqrt{\lambda_1}\frac{\partial\Delta u_1[n]}{\partial\mathbf{K_{1u}}[n-1]} & \cdots & \sqrt{\lambda_1}\frac{\partial\Delta u_1[n]}{\partial\mathbf{K_{4u}}[n-1]}\\ \vdots & \ddots & \vdots\\ \sqrt{\lambda_R}\frac{\partial\Delta u_R[n]}{\partial\mathbf{K_{1u}}[n-1]} & \cdots & \sqrt{\lambda_R}\frac{\partial\Delta u_R[n]}{\partial\mathbf{K_{4u}}[n-1]}\end{bmatrix} \quad (5)$$

and $\hat{\mathbf{e}}$ is error vector

$$\hat{\mathbf{e}}=\begin{bmatrix}\beta_1\hat{e}_1[n+1]\\ \vdots\\ \beta_Q\hat{e}_Q[n+P_H]\\ \sqrt{\lambda_1}\Delta u_1[n]\\ \vdots\\ \sqrt{\lambda_R}\Delta u_R[n]\end{bmatrix}=\begin{bmatrix}\beta_1[r_1[n+1]-\hat{y}_1[n+1]]\\ \vdots\\ \beta_Q[r_Q[n+P_H]-\hat{y}_Q[n+P_H]]\\ \sqrt{\lambda_1}\left[u_1[n]-u_1[n-1]\right]\\ \vdots\\ \sqrt{\lambda_R}\left[u_R[n]-u_R[n-1]\right]\end{bmatrix} \quad (6)$$

The Jacobian matrix ($\mathbf{J}$) can be decomposed into two parts representing the sensitivty of the system($\mathbf{J}_m$) and controller ($\mathbf{J}_c$) depending on their adjustable inputs as in (7).

$$\mathbf{J}_m = \begin{bmatrix} -\frac{\partial \hat{y}_1[n+1]}{\partial \mathbf{u}[n]} \\ \vdots \\ -\frac{\partial \hat{y}_Q[n+P_H]}{\partial \mathbf{u}[n]} \\ \sqrt{\lambda_1}\frac{\partial [u_1[n]-u_1[n-1]]}{\partial \mathbf{u}[n]} \\ \vdots \\ \sqrt{\lambda_R}\frac{\partial [u_R[n]-u_R[n-1]]}{\partial \mathbf{u}[n]} \end{bmatrix} \tag{7}$$

$$\mathbf{J}_c = \begin{bmatrix} \frac{\partial \mathbf{u}[n]}{\partial \mathbf{K}_{1_\mathbf{u}}[n-1]} \cdots \frac{\partial \mathbf{u}[n]}{\partial \mathbf{K}_{4_\mathbf{u}}[n-1]} \end{bmatrix}$$

As can be seen from (7), the $\mathbf{J}_m$ part of the system Jacobian matrix depends on system dynamics estimated via $\mathrm{RK}_{\mathrm{model}}$. $\mathrm{RK}_{\mathrm{model}}$ can be successfully deployed to accomplish P_H-step ahead unknown $\frac{\partial y_Q[n+P_H]}{\partial u_r[n]}$ term. A suboptimal correction term($\delta\mathbf{u}[n]$), utilized to eliminate the non-optimality effects of controller parameters added to control signal, can be obtained via Taylor approximation of the $F(\mathbf{u}[\mathbf{n}], \mathbf{e}_q)$ given in (3)(Iplikci 2010a; Iplikci 2013):

$$F(\mathbf{u}[\mathbf{n}]+\delta\mathbf{u}[\mathbf{n}]) \cong F(\mathbf{u}[\mathbf{n}]) + \frac{\partial F(\mathbf{u}[\mathbf{n}])}{\partial \mathbf{u}[\mathbf{n}]}\delta\mathbf{u}[\mathbf{n}] + \frac{1}{2}\frac{\partial^2 F(\mathbf{u}[\mathbf{n}])}{\partial^2 \mathbf{u}[\mathbf{n}]}(\delta\mathbf{u}[\mathbf{n}])^2 \tag{8}$$

For optimality of $\delta\mathbf{u}[n]$ (3)(Iplikci 2010a; Iplikci 2013)

$$\frac{\partial F(\mathbf{u}[\mathbf{n}]+\delta\mathbf{u}[\mathbf{n}])}{\partial \delta\mathbf{u}[\mathbf{n}]} \cong \frac{\partial F(\mathbf{u}[\mathbf{n}])}{\partial \mathbf{u}[\mathbf{n}]} + \frac{\partial^2 F(\mathbf{u}[\mathbf{n}])}{\partial^2 \mathbf{u}[\mathbf{n}]}\delta\mathbf{u}[\mathbf{n}] = 0 \tag{9}$$

Thus, $\delta\mathbf{u}[n]$ term is concluded as (3)(Iplikci 2010a; Iplikci 2013)

$$\delta\mathbf{u}[\mathbf{n}] = -\frac{\frac{\partial F(\mathbf{u}[\mathbf{n}])}{\partial \mathbf{u}[\mathbf{n}]}}{\frac{\partial^2 F(\mathbf{u}[\mathbf{n}])}{\partial^2 \mathbf{u}[\mathbf{n}]}} \tag{10}$$

As given in (10), computation of $\delta\mathbf{u}[n]$ is subject to $\frac{\partial F(\mathbf{u}[\mathbf{n}])}{\partial \mathbf{u}[\mathbf{n}]}$ and $\frac{\partial^2 F(\mathbf{u}[\mathbf{n}])}{\partial^2 \mathbf{u}[\mathbf{n}]}$ terms. The $\frac{\partial^2 F(\mathbf{u}[\mathbf{n}])}{\partial^2 \mathbf{u}[\mathbf{n}]}$ term can be formed via (3) as

$$\frac{\partial F(\mathbf{u}[\mathbf{n}])}{\partial \mathbf{u}[\mathbf{n}]} = 2\mathbf{J}_m^T\hat{\mathbf{e}} \tag{11}$$

The complexity of Hessian term ($\frac{\partial^2 F(\mathbf{u}[\mathbf{n}])}{\partial^2 \mathbf{u}[\mathbf{n}]}$) resulting from 2nd order derivatives can be diminished using approximation of Hessian term as follows:

$$\frac{\partial^2 F(\mathbf{u}[\mathbf{n}])}{\partial^2 \mathbf{u}[\mathbf{n}]} = 2\mathbf{J}_m^T\mathbf{J}_m \tag{12}$$

Thus, (10) can be rexpressed as

$$\delta\mathbf{u}[\mathbf{n}] = -[\mathbf{J}_m^T\mathbf{J}_m]^{-1}\mathbf{J}_m^T\hat{\mathbf{e}} \tag{13}$$

2.3 Control Phase

Then, employing the trained $\mathrm{RK}_{\mathrm{controller}}$ parameters ($\mathbf{K}^{new}$) obtained in (4) and suboptimal correction term ($\delta\mathbf{u}[n]$), the updated new control action ($\mathbf{u}^\star[n] = \mathbf{u}[n] + \delta\mathbf{u}[n]$) can be acquired via (1,13) so as to adaptively form closed-loop dynamics as desired. To this point, the essentials of proposed adjustment mechanism have been outlined. The derivation of the update rules for $\mathrm{RK}_{\mathrm{controller}}$ are detailed in next section.

3 Adaptive $\mathrm{RK}_{\mathrm{controller}}$

3.1 An overview of $\mathrm{RK}_{\mathrm{controller}}$

Assume that the dynamics of the controller are expressed via the ODE in (14)

$$\dot{\mathbf{u}}(t) = f(\mathbf{u}(t), \mathbf{\Omega}(t)) \tag{14}$$

with the initial condition $\mathbf{u}(0) = \mathbf{u}_0$ (Uçak 2019). If it is assumed that assume that $\mathbf{f}_c$ is known, one-step ahead control signal vector can be computed via 4th order RK ODE solver given in (15):

$$\mathbf{u}[\mathbf{n}] = \mathbf{u}[\mathbf{n}-\mathbf{1}] + \frac{1}{6}[\mathbf{K}_{1\mathbf{u}}[n-1] + 2\mathbf{K}_{2\mathbf{u}}[n-1] + 2\mathbf{K}_{3\mathbf{u}}[n-1] + \mathbf{K}_{4\mathbf{u}}[n-1]] \tag{15}$$

in which $\mathbf{K}_{1\mathbf{u}}[n-1]$, $\mathbf{K}_{2\mathbf{u}}[n-1]$, $\mathbf{K}_{3\mathbf{u}}[n-1]$ and $\mathbf{K}_{4\mathbf{u}}[n-1]$ express the changing rates of the MIMO controller states (Efe and Kaynak 1999). These changing rates can be attained as (Iplikci 2013; Efe and Kaynak 1999; Wang and Lin 1998):

$$
\begin{aligned}
\mathbf{K_{1u}}[n-1] &= \mathrm{T}_c \mathbf{f}_c\big(\mathbf{x}_1[n-1], \mathbf{\Omega}[n]\big)\Big|_{\mathbf{x}_1[n-1]=\mathbf{u}[n-1]} \\
\mathbf{K_{2u}}[n-1] &= \mathrm{T}_c \mathbf{f}_c\big(\mathbf{x}_2[n-1], \mathbf{\Omega}[n]\big)\Big|_{\mathbf{x}_2[n-1]=\mathbf{u}[n-1]+\frac{1}{2}\mathbf{K_{1u}}[n-1]} \\
\mathbf{K_{3u}}[n-1] &= \mathrm{T}_c \mathbf{f}_c\big(\mathbf{x}_3[n-1], \mathbf{\Omega}[n]\big)\Big|_{\mathbf{x}_3[n-1]=\mathbf{u}[n-1]+\frac{1}{2}\mathbf{K_{2u}}[n-1]} \\
\mathbf{K_{4u}}[n-1] &= \mathrm{T}_c \mathbf{f}_c\big(\mathbf{x}_4[n-1], \mathbf{\Omega}[n]\big)\Big|_{\mathbf{x}_4[n-1]=\mathbf{u}[n-1]+\mathbf{K_{3u}}[n-1]}
\end{aligned}
\tag{16}
$$

where "T_c" stands for the Runge-Kutta integration stepsize (Efe and Kaynak 1999), $\mathbf{f}_c$ indicates the control signal functions and $\mathbf{\Omega}[n]$ vector covers all signals such as the reference, system outputs etc except for control signal. However, the dynamics of control signal functions($\mathbf{f}_c$) are unavailable for the controller. Therefore, the optimization aim in (15) is to acquire the optimal values of slopes ($\mathbf{K_{1u}}\ \mathbf{K_{2u}}\ \mathbf{K_{3u}}\ \mathbf{K_{4u}}$) without knowing $\mathbf{f}_c$ functions. Thus, the computed control signal via $\mathrm{RK}_{\mathrm{controller}}$ is rexpressed in (17) where

$$
\mathbf{u}[\mathbf{n}] = f_{RK}(\mathbf{u}, \mathbf{K}) = \mathbf{u}[\mathbf{n-1}] + \frac{1}{6}\mathbf{K_{1u}}[n-1] + \frac{2}{6}\mathbf{K_{2u}}[n-1] + \frac{2}{6}\mathbf{K_{3u}}[n-1] + \frac{1}{6}\mathbf{K_{4u}}[n-1] \tag{17}
$$

in which $\mathbf{K_{1u}}\ \mathbf{K_{2u}}\ \mathbf{K_{3u}}\ \mathbf{K_{4u}}$ are unknown and adjustable parameters of the $\mathrm{RK}_{\mathrm{controller}}$. The structure of the $\mathrm{RK}_{\mathrm{controller}}$ is illustrated in Figure 2.

3.2 Adjustment laws for $\mathrm{RK}_{\mathrm{controller}}$

In this subsection, the adjustment laws for $\mathrm{RK}_{\mathrm{controller}}$($\mathbf{K} = \big[\mathbf{K_{1u}}\ \mathbf{K_{2u}}\ \mathbf{K_{3u}}\ \mathbf{K_{4u}}\big]^T$) exploited to attain feasible control vector in (15) are derived. $\mathrm{RK}_{\mathrm{controller}}$ parameters to be optimized are given as follows:

$$
\mathbf{K} = \big[\mathbf{K_{1u}}\ \mathbf{K_{2u}}\ \mathbf{K_{3u}}\ \mathbf{K_{4u}}\big]^T \tag{18}
$$

Thus, using LM optimization rule in (3), $\mathrm{RK}_{\mathrm{controller}}$ parameters can be optimized as given in (4–7). As given in (7), the Jacobian matrix can be partitioned into two part as ($\boldsymbol{J} = \boldsymbol{J}_m \boldsymbol{J}_c$) where $\boldsymbol{J}_m$ is system sensitivity and $\boldsymbol{J}_c$ denotes $\mathrm{RK}_{\mathrm{controller}}$ sensitivity. The construction of $\boldsymbol{J}_m$ matrix is detailed in (Iplikci 2013; Uçak 2019; Uçak 2020). In order to form $\boldsymbol{J}_c$ matrix in (7), it is essential to derive the $\frac{\partial \mathbf{u}[n]}{\partial \mathbf{K_{1_u}}[n-1]}$, $\frac{\partial \mathbf{u}[n]}{\partial \mathbf{K_{2_u}}[n-1]}$, $\frac{\partial \mathbf{u}[n]}{\partial \mathbf{K_{3_u}}[n-1]}$ and $\frac{\partial \mathbf{u}[n]}{\partial \mathbf{K_{4_u}}[n-1]}$ terms. Employing chain rule, the mentioned terms are expressed in (19–23):

$$
\frac{\partial \mathbf{u}[n]}{\partial \mathbf{K_{4_u}}[n-1]} = \frac{1}{6} \tag{19}
$$

$$
\begin{aligned}
\frac{\partial \mathbf{u}[n]}{\partial \mathbf{K_{3_u}}[n-1]} &= \frac{\partial \mathbf{u}[n]}{\partial \mathbf{K_{3_u}}[n-1]} + \frac{\partial \mathbf{u}[n]}{\partial \mathbf{K_{4_u}}[n-1]} \frac{\partial \mathbf{K_{4_u}}[n-1]}{\partial \mathbf{K_{3_u}}[n-1]} \\
\frac{\partial \mathbf{u}[n]}{\partial \mathbf{K_{3_u}}[n-1]} &= \left[\tfrac{2}{6}\ \tfrac{1}{6}\right] \left[\mathrm{T}_c \frac{\partial \mathbf{f}\big(\mathbf{x}_4[n-1], \mathbf{\Omega}[n]\big)}{\partial \mathbf{x}_4[n-1]} \right]_{\mathbf{x}_4[n-1]=\mathbf{u}[n-1]+\mathbf{K_{3u}}[n-1]}
\end{aligned}
\tag{20}
$$

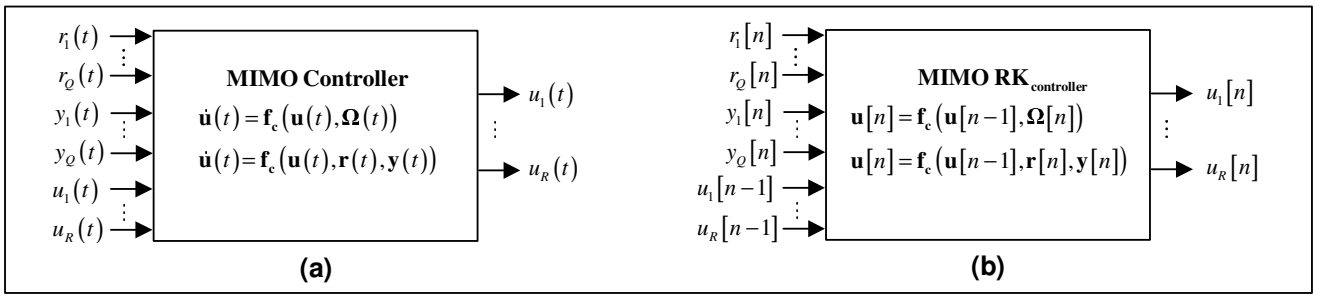

Fig. 2 (**a**) A continuous MIMO controller and (**b**) its RK counterpart.

$$\frac{\partial \mathbf{u}[n]}{\partial \mathbf{K}_{2_\mathbf{u}}[n-1]} = \frac{\partial \mathbf{u}[n]}{\partial \mathbf{K}_{2_\mathbf{u}}[n-1]} + \frac{\partial \mathbf{u}[n]}{\partial \mathbf{K}_{4_\mathbf{u}}[n-1]}\frac{\partial \mathbf{K}_{4_\mathbf{u}}[n-1]}{\partial \mathbf{K}_{2_\mathbf{u}}[n-1]} + \frac{\partial \mathbf{u}[n]}{\partial \mathbf{K}_{3_\mathbf{u}}[n-1]}\frac{\partial \mathbf{K}_{3_\mathbf{u}}[n-1]}{\partial \mathbf{K}_{2_\mathbf{u}}[n-1]}$$

$$\frac{\partial \mathbf{u}[n]}{\partial \mathbf{K}_{2_\mathbf{u}}[n-1]} = \left[\tfrac{2}{6}\ \tfrac{1}{6}\ \tfrac{2}{6}\right]\begin{bmatrix} \mathrm{T}_c\dfrac{\partial \mathbf{f}\left(\mathbf{x}_4[n-1],\mathbf{\Omega}[n]\right)}{\partial \mathbf{x}_4[n-1]} & 1 \\ \mathrm{T}_c\dfrac{\partial \mathbf{f}\left(\mathbf{x}_3[n-1],\mathbf{\Omega}[n]\right)}{\partial \mathbf{x}_3[n-1]}\tfrac{1}{2} & \mathrm{T}_c\dfrac{\partial \mathbf{f}\left(\mathbf{x}_3[n-1],\mathbf{\Omega}[n]\right)}{\partial \mathbf{x}_3[n-1]}\tfrac{1}{2} \end{bmatrix}\begin{bmatrix} \mathbf{x}_3[n-1] = \mathbf{u}[n-1] + \tfrac{1}{2}\mathbf{K}_{2\mathbf{u}}[n-1] \\ \mathbf{x}_4[n-1] = \mathbf{u}[n-1] + \mathbf{K}_{3\mathbf{u}}[n-1] \end{bmatrix} \tag{21}$$

$$\frac{\partial \mathbf{u}[n]}{\partial \mathbf{K}_{1_\mathbf{u}}[n-1]} = \frac{\partial \mathbf{u}[n]}{\partial \mathbf{K}_{1_\mathbf{u}}[n-1]} + \frac{\partial \mathbf{u}[n]}{\partial \mathbf{K}_{4_\mathbf{u}}[n-1]}\frac{\partial \mathbf{K}_{4_\mathbf{u}}[n-1]}{\partial \mathbf{K}_{1_\mathbf{u}}[n-1]} + \frac{\partial \mathbf{u}[n]}{\partial \mathbf{K}_{3_\mathbf{u}}[n-1]}\frac{\partial \mathbf{K}_{3_\mathbf{u}}[n-1]}{\partial \mathbf{K}_{1_\mathbf{u}}[n-1]} + \frac{\partial \mathbf{u}[n]}{\partial \mathbf{K}_{2_\mathbf{u}}[n-1]}\frac{\partial \mathbf{K}_{2_\mathbf{u}}[n-1]}{\partial \mathbf{K}_{1_\mathbf{u}}[n-1]}$$

$$\frac{\partial \mathbf{u}[n]}{\partial \mathbf{K}_{1_\mathbf{u}}[n-1]} = \left[\tfrac{1}{6}\ \tfrac{1}{6}\ \tfrac{2}{6}\ \tfrac{2}{6}\right]\begin{bmatrix} 1 \\ \dfrac{\partial \mathbf{K}_{4_\mathbf{u}}[n-1]}{\partial \mathbf{K}_{1_\mathbf{u}}[n-1]} \\ \dfrac{\partial \mathbf{K}_{3_\mathbf{u}}[n-1]}{\partial \mathbf{K}_{1_\mathbf{u}}[n-1]} \\ \dfrac{\partial \mathbf{K}_{2_\mathbf{u}}[n-1]}{\partial \mathbf{K}_{1_\mathbf{u}}[n-1]} \end{bmatrix} \tag{22}$$

where

$$
\begin{aligned}
\frac{\partial \mathbf{K}_{2_\mathbf{u}}[n-1]}{\partial \mathbf{K}_{1_\mathbf{u}}[n-1]} &= \frac{\partial \mathbf{K}_{2_\mathbf{u}}[n-1]}{\partial \mathbf{f}(\mathbf{x}_2[n-1], \mathbf{\Omega}[n])} \frac{\partial \mathbf{f}(\mathbf{x}_2[n-1], \mathbf{\Omega}[n])}{\partial \mathbf{x}_2[n-1]} \frac{\partial \mathbf{x}_2[n-1]}{\partial \mathbf{K}_{1_\mathbf{u}}[n-1]} \\
\frac{\partial \mathbf{K}_{2_\mathbf{u}}[n-1]}{\partial \mathbf{K}_{1_\mathbf{u}}[n-1]} &= \mathrm{T}_c \frac{\partial \mathbf{f}(\mathbf{x}_2[n-1], \mathbf{\Omega}[n])}{\partial \mathbf{x}_2[n-1]}_{\mathbf{x}_2[n-1]=\mathbf{u}[n-1]+\frac{1}{2}\mathbf{K}_{1_\mathbf{u}}[n-1]} \frac{1}{2} \\
\frac{\partial \mathbf{K}_{3_\mathbf{u}}[n-1]}{\partial \mathbf{K}_{1_\mathbf{u}}[n-1]} &= \frac{\partial \mathbf{K}_{3_\mathbf{u}}[n-1]}{\partial \mathbf{f}(\mathbf{x}_3[n-1], \mathbf{\Omega}[n])} \frac{\partial \mathbf{f}(\mathbf{x}_3[n-1], \mathbf{\Omega}[n])}{\partial \mathbf{x}_3[n-1]} \frac{\partial \mathbf{x}_3[n-1]}{\partial \mathbf{K}_{2_\mathbf{u}}[n-1]} \frac{\partial \mathbf{K}_{2_\mathbf{u}}[n-1]}{\partial \mathbf{K}_{1_\mathbf{u}}[n-1]} \\
\frac{\partial \mathbf{K}_{3_\mathbf{u}}[n-1]}{\partial \mathbf{K}_{1_\mathbf{u}}[n-1]} &= \mathrm{T}_c \frac{\partial \mathbf{f}(\mathbf{x}_3[n-1], \mathbf{\Omega}[n])}{\partial \mathbf{x}_3[n-1]}_{\mathbf{x}_3[n-1]=\mathbf{u}[n-1]+\frac{1}{2}\mathbf{K}_{2_\mathbf{u}}[n-1]} \frac{1}{2} \frac{\partial \mathbf{K}_{2_\mathbf{u}}[n-1]}{\partial \mathbf{K}_{1_\mathbf{u}}[n-1]} \\
\frac{\partial \mathbf{K}_{4_\mathbf{u}}[n-1]}{\partial \mathbf{K}_{1_\mathbf{u}}[n-1]} &= \frac{\partial \mathbf{K}_{4_\mathbf{u}}[n-1]}{\partial \mathbf{f}(\mathbf{x}_4[n-1], \mathbf{\Omega}[n])} \frac{\partial \mathbf{f}(\mathbf{x}_4[n-1], \mathbf{\Omega}[n])}{\partial \mathbf{x}_4[n-1]} \frac{\partial \mathbf{x}_4[n-1]}{\partial \mathbf{K}_{3_\mathbf{u}}[n-1]} \frac{\partial \mathbf{K}_{3_\mathbf{u}}[n-1]}{\partial \mathbf{K}_{1_\mathbf{u}}[n-1]} \\
\frac{\partial \mathbf{K}_{4_\mathbf{u}}[n-1]}{\partial \mathbf{K}_{1_\mathbf{u}}[n-1]} &= \mathrm{T}_c \frac{\partial \mathbf{f}(\mathbf{x}_4[n-1], \mathbf{\Omega}[n])}{\partial \mathbf{x}_4[n-1]}_{\mathbf{x}_4[n-1]=\mathbf{u}[n-1]+\mathbf{K}_{3_\mathbf{u}}[n-1]} \frac{1}{2} \frac{\partial \mathbf{K}_{3_\mathbf{u}}[n-1]}{\partial \mathbf{K}_{1_\mathbf{u}}[n-1]}
\end{aligned}
\tag{23}
$$

As can be seen from derived update rules, in order to acquire $\frac{\partial \mathbf{u}[n]}{\partial \mathbf{K}_{1_\mathbf{u}}[n-1]}$, $\frac{\partial \mathbf{u}[n]}{\partial \mathbf{K}_{2_\mathbf{u}}[n-1]}$, $\frac{\partial \mathbf{u}[n]}{\partial \mathbf{K}_{3_\mathbf{u}}[n-1]}$ and $\frac{\partial \mathbf{u}[n]}{\partial \mathbf{K}_{4_\mathbf{u}}[n-1]}$ terms, it is required to know $\frac{\partial \mathbf{f}(\mathbf{u}[n-1],\mathbf{\Omega}[n])}{\partial \mathbf{u}[n-1]}$ term although $\mathbf{f}(\mathbf{u}[n-1],\mathbf{\Omega}[n])$ function is unknown.

Assumption:
For convenience, it is assumed that the relation between $\mathbf{u}[n-1]$ and $\mathbf{f}$ is known in the following form:

$$\mathbf{f}(\mathbf{u}[n-1],\mathbf{\Omega}[n]) = \mathbf{u}^d[n-1] + \mathbf{f}_{\text{unknown terms}}(\mathbf{r},\mathbf{y}) \quad (24)$$

where $\mathbf{f}_{\text{unknown terms}}(\mathbf{r},\mathbf{y})$ represents the unknown part of the function and d indicates the degree of $\mathbf{u}[n-1]$. Thus, the missing piece of the derivations $\frac{\partial \mathbf{f}(\mathbf{u}[n-1],\mathbf{\Omega}[n])}{\partial \mathbf{u}[n-1]}$ is achieved in (25)

$$\frac{\partial \mathbf{f}(\mathbf{u}[n-1],\mathbf{\Omega}[n])}{\partial \mathbf{u}[n-1]} = d\mathbf{u}^{d-1}[n-1] \quad (25)$$

The most prominent feature of the introduced architecture is that under the assumption that the controller architecture in the controller block is composed of an unknown pure differential equation, this control signal is discretized and derived primarily by the Runge-Kutta integration method. Then, with the Levenberg-Marquardt optimization method, the information required to adapt these controller dynamics is analyzed. As a result of this analysis, when the control signal is discretized by Runge-Kutta integration, the degree of dependency of the control signal with the control signal attained in the previous step given in (24,25) is sufficient to transform the controller structure into an adaptive controller and adjust controller parameters.

The outstanding feature that distinguishes this study from previous works (Uçak 2019; Uçak 2020) and all other publications based on Runge-Kutta given in (Iplikci 2013; Cetin and Iplikci 2015; Beyhan 2013; Efe and Kaynak 2000; Efe and Kaynak 1999; Wang and Lin 1998) is that the Runge-Kutta integration is directly presented as a Runge-Kutta controller in the controller block for the first time. No machine learning method is utilized to store statistical information of the control signal or the controller parameters, such as a neural network etc.

4 Simulation Results

$\mathrm{RK}_{\mathrm{controller}}$ performance has been evaluated on a nonlinear VdV system. However, it is possible to deploy the introduced architecture to a wide varity of control systems to overcome characteristics rarifying control task such as nonlinearity, instability, etc. In order to better reveal the efficiency of $\mathrm{RK}_{\mathrm{controller}}$, competencies of $\mathrm{RK}_{\mathrm{controller}}$ such as tracking, robustness etc. have been examined under three different situations that are essential in control systems: nominal conditions, noise in measurement and parametric uncertainty. VdV systems are frequently deployed for performance examination of MIMO controller architectures (Iplikci 2013; Cetin and Iplikci 2015; Iplikci 2010b). Due to its non-minimum-phase behavior and harsh nonlinearity (Iplikci 2013; Uçak 2019; Uçak 2020), it is significant to be controlled adaptively in order to attune the occuring divergent behaviours. The reaction scheme of VdV is given as follows:

$$\begin{aligned} &\mathbf{A} \xrightarrow{k_1} \mathbf{B} \xrightarrow{k_2} \mathbf{C} \\ &2\mathbf{A} \xrightarrow{k_3} \mathbf{D} \end{aligned} \quad (26)$$

where cyclopentadiene ($\boldsymbol{A}$) is the inlet reactant, cyclopentenol ($\boldsymbol{B}$) indicates the indended compound, dicyclopentadiene ($\boldsymbol{D}$) is produced by Diels-Alder reaction, and cyclopentanediol ($\boldsymbol{C}$) is a resulting compound emerging as another water molecule is added (Uçak 2019; Engell and Klatt 1993), and k_i's stand for the reaction rates (Engell and Klatt 1993; Chen et al. 1995; Vojtesek and Dostál 2010; Jørgensen 2007; Kulikov and Kulikova 2014). In chemical reaction given in (26), the aim is to produce $\boldsymbol{B}$ from $\boldsymbol{A}$ (Uçak 2019; Engell and Klatt 1993). The reaction in (26) is detailed via corresponding chemical compounds as follows:

$$\begin{aligned} &\underset{Cyclopentadiene(\mathbf{A})}{\mathbf{C_5H_6}} \xrightarrow{+H_2O(k_1)} \underset{Cyclopentenol(\mathbf{B})}{\mathbf{C_5H_7OH}} \xrightarrow{+H_2O(k_2)} \underset{Cyclopentanediol(\mathbf{C})}{\mathbf{C_5H_8(OH)_2}} \\ &\underset{Cyclopentadiene(\mathbf{A})}{\mathbf{2C_5H_6}} \xrightarrow{k_3} \underset{Dicyclopentadiene(\mathbf{D})}{\mathbf{C_{10}H_{12}}} \end{aligned} \quad (27)$$

The dynamics of the reaction in (26,27) can be expressed by ODE's in (28):

$$
\begin{aligned}
\dot{C}_A(t) &= \frac{F}{V}\Big(C_{A0} - C_A(t)\Big) - k_{10}e^{\frac{-E_1}{T}}C_A(t) - k_{30}e^{\frac{-E_3}{T}}C_A^2(t) \\
\dot{C}_B(t) &= -\frac{F}{V}C_B(t) + k_{10}e^{\frac{-E_1}{T}}C_A(t) - k_{20}e^{\frac{-E_2}{T}}C_B(t) \\
\dot{T}(t) &= \frac{1}{\rho C_p}\Big[k_{10}e^{\frac{-E_1}{T}}C_A(t)\big(-\Delta H_1\big) + k_{20}e^{\frac{-E_2}{T}}C_B(t)\big(-\Delta H_2\big) + k_{30}e^{\frac{-E_3}{T}}C_A^2(t)\big(-\Delta H_3\big)\Big] \\
&+ \frac{F}{V}\big(T_0 - T(t)\big) + \frac{Q}{\rho C_p}
\end{aligned}
\tag{28}
$$

where physico-chemical parameters of system are tabulated in Table 1 (Iplikci 2013; Uçak 2019; Chen et al. 1995; Vojtesek and Dostál 2010; Niemiec and Kravaris 2003; Kravaris et al. 1998). The $\mathrm{RK}_{\mathrm{controller}}$ intends to regulate $y_1 = C_B$ and $y_2 = T$ by obtaining the optimal $u_1 = F/V$ and $u_2 = Q$ (Iplikci 2013; Niemiec and Kravaris 2003). The details about system are available in (Iplikci 2013; Uçak 2019; Chen et al. 1995; Vojtesek and Dostál 2010; Niemiec and Kravaris 2003; Kravaris et al. 1998).

4.1 Performance evaluation of $\mathrm{RK}_{\mathrm{controller}}$

Expecting a controller that fails in nominal conditions to succeed in difficult cases is an unrealistic anticipation. Therefore, firstly, the performance of the $\mathrm{RK}_{\mathrm{controller}}$ is tested for nominal case where all the information about the system is exacly known. When staircase signal is applied to the closed–loop adaptive system with nominal conditions, the obtained system outputs, control signals applied to the system and corrections terms are illustrated in Figure 3. The abrupt alternations in reference signals induce strong coupling among $y_1 = C_B$ and $y_2 = T$. Still, the closed-loop system dynamics are controlled as desired. $\mathrm{RK}_{\mathrm{controller}}$ generates the optimal control signals illustrated in Figure 3 (**b**,**e**). The optimal control signal applied to the system is composed of two terms(Iplikci 2013; Uçak 2019; Uçak 2020). These are $\boldsymbol{u}_{RK-C}[n]$ term which is the control signal produced by only $\mathrm{RK}_{\mathrm{controller}}$ and the correction term($\delta\boldsymbol{u}[n]$) used to improve the transient state behaviour of the controlled system since the adjusted controlled parameters may not be optimal at transient state(Iplikci 2010a; Uçak 2019; Uçak 2020). Therefore, the task share between $\mathrm{RK}_{\mathrm{controller}}$ ($\boldsymbol{u}_{RK-C}[n]$) and correction terms ($\delta\boldsymbol{u}[n]$) is also examined in this paper. The correction terms that provide large contribution to control signal, especially in transient state, are also given in Figure 3 (**c**,**f**). If it is focused on the occuring abrupt changes at $[30, 40]$ h and $[10, 20]$ h, the control signals are succesfully updated to comply with the arising strong coupling among C_B and reactor temperature (T). $\mathrm{RK}_{\mathrm{controller}}$ succesfully tracks the desired signals as illustrated in Figure 3 (**a**,**d**), and $u_i(t)$ and $\delta u_i(t)$ terms produced by $\mathrm{RK}_{\mathrm{controller}}$ are also demonstrated in Figure 3 (**b**,**c**,**e**,**f**). In order to show the adaptation of the $\mathrm{RK}_{\mathrm{controller}}$ parameters, the convergence and also adjustment of control signals are given in Figure 4. The adjustable parameters of $\mathrm{RK}_{\mathrm{controller}}$ converge to their optimal values on short notice. It is aimed to apperceive the duty share between $\mathrm{RK}_{\mathrm{controller}}(\mathbf{u}_{\text{RK-C}}[n])$ and correction term($\delta\mathbf{u}[n]$).The duty share percentages of $\mathbf{u}_{\text{RK-C}}[n]$ and $\delta\mathbf{u}[n]$ terms are illustrated in Figure 5. As can be seen from Figure 5 (**a**,**b**), $\mathbf{u}_{\text{RK-C}}[n]$ and $\delta\mathbf{u}[n]$ are carrying out control task together. Especially in transient states, $\delta\mathbf{u}[n]$ term demonstrates its influence on the control mechanism. Therefore, $\delta\mathbf{u}[n]$ is a vital part of the adjustment mechanism. As can be seen from Figure 5 (**a**,**b**), $\mathrm{RK}_{\mathrm{controller}}$ performs dominant behaviour in comparison to $\delta\mathbf{u}[n]$ term. It is explicity seen that $\delta\mathbf{u}[n]$ immediately hands over the control to $\mathrm{RK}_{\mathrm{controller}}$, and $\mathrm{RK}_{\mathrm{controller}}$ performs the control procedure. In addition to staircase reference signal, the performance evaluation has been carried out for sinusoidal type reference inputs. For this purpose, the reference signal for T is chosen as sinusoidal signal and reference for C_B is assigned as fixed value during control. The closed-loop response of the $\mathrm{RK}_{\mathrm{controller}}$, $\mathbf{u}_{\text{RK-C}}[n]$ and $\delta\mathbf{u}[n]$ for $u_1(t)$ terms are shown in Figure 6. The control failover of $\mathbf{u}_{\text{RK-C}}[n]$ and $\delta\mathbf{u}[n]$ are indicated in Figure 7. The percentage of the control task sharing between $\mathrm{RK}_{\mathrm{controller}}(\mathbf{u}_{\text{RK-C}}[n])$ and $\delta\mathbf{u}[n]$ term for sinusoidal inputs is shown in Figure 7. Since the noise resulting from measurement devices contaminates the control procedure/operation, robustness examination of the $\mathrm{RK}_{\mathrm{controller}}$ interms of measurement noise is requisite. The robustness and also tracking performance of the $\mathrm{RK}_{\mathrm{controller}}$ against measurement noise have been perused. Therefore, $y_1(t)$ and $y_2(t)$ are exposed to measurement noise with $\sigma_{C_B(t)} = \sigma_{T(t)} = 0.0003$ standart deviations for C_B and for T. The tracking performance, control and correction terms are depicted in Figure 8. Inspite of noisy conditions, the system outputs can be compelled to the desired set points. When the control task sharings in Figure 9 are evaluated, it is observed that $\mathrm{RK}_{\mathrm{controller}}$ takes over the control task as soon as possible. As given in Figure 9, it is clear that $\delta\mathbf{u}[n]$ is only effective when $\mathbf{u}_{\text{RK-C}}[n]$ is not optimal initially; however, $\delta\mathbf{u}[n]$ instantly converges to zero and $\mathrm{RK}_{\mathrm{controller}}$ has taken on virtually all control task.

Another significant case to assess the robustness of the $\mathrm{RK}_{\mathrm{controller}}$ is evaluation of control performance when one of the system parameter is unknown or not determined accurately(Iplikci 2013; Uçak 2019; Uçak 2020). The control performance of the $\mathrm{RK}_{\mathrm{controller}}$ depends concretely on the preciseness of the $\mathrm{RK}_{\mathrm{model}}$ since the parameters of $\mathrm{RK}_{\mathrm{controller}}$ are adjusted by considering future trajectories of the system via iterative predictive system model. Therefore, to obtain unknown $\mathrm{RK}_{\mathrm{model}}$ parameters, $\mathrm{RK}_{\mathrm{estimator}}$ subblock in $\mathrm{RK}_{\mathrm{model}}$ is employed to approximate the system model parameters as given in (Iplikci 2013; Uçak 2019; Uçak 2020). For this purpose, it is essential to appraise not only the tracking performance of the $\mathrm{RK}_{\mathrm{controller}}$ but also system model parameter approximation accuracy

Table 1 System Parameters for VdV (Iplikci 2013; Uçak 2019; Chen et al. 1995; Vojtesek and Dostál 2010; Niemiec and Kravaris 2003; Kravaris et al. 1998).

Description of parameter	Symbol	Value of parameter
Molar concentrations of A	C_A	$----$
Molar concentrations of B	C_B	$----$
Reactor temperature	T	$----$
Dilution rate	F/V	$----$
Added/removed heat rate per unit volume	Q	$----$
Reaction k_1: Collision factor	k_{10}	$1.287x10^{12}\left[h^{-1}\right]$
Reaction k_2: Collision factor	k_{20}	$1.287x10^{12}\left[h^{-1}\right]$
Reaction k_3: Collision factor	k_{30}	$9.043x10^{9}\left[h^{-1}l/mol\right]$
Reaction k_1: Activation energy	E_1	$9758.3\left[K\right]$
Reaction k_2: Activation energy	E_2	$9758.3\left[K\right]$
Reaction k_3: Activation energy	E_3	$8560.0\left[K\right]$
Reaction k_1: Enthalpy	ΔH_1	$4.2\left[kJ/mol\right]$
Reaction k_2: Enthalpy	ΔH_2	$-11\left[kJ/mol\right]$
Reaction k_3: Enthalpy	ΔH_3	$-41.85\left[kJ/mol\right]$
Substance A: Concentration in the feed stream	C_{A0}	$5.0\left[mol/l\right]$
Feed temperature	T_0	$403.15\left[K\right]$
Density	ρ	$0.9342\left[kg/l\right]$
Heat capacity	C_p	$3.01\left[kJ/kgK\right]$
Reactor volume	V	$10.0\left[l\right]$

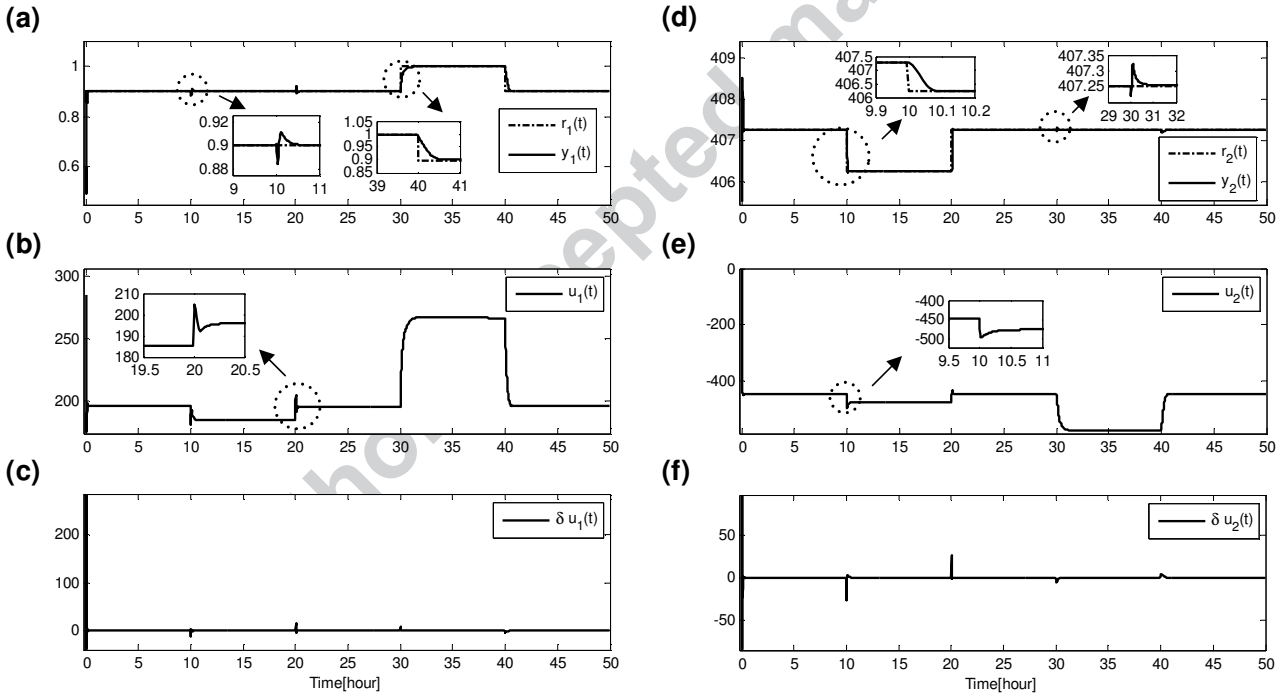

Fig. 3 (**a**,**d**) Tracking beahviours, (**b**,**e**) control signals and (**c**,**f**) correction terms for nominal case.

of $\mathrm{RK}_{\mathrm{estimator}}$ under parametric uncertainty case. As a scenario, a nonlinear uncertainity is introduced to system parameter as $C_{A0}(t) = 5 + 0.5sin\left(\frac{2}{25}\pi t\right)$ and input signals are set to as 0.95 and 407.25. The behaviour of the closed-loop system and approximation ability of $\mathrm{RK}_{\mathrm{estimator}}$ subblock are depicted in Figure 10. The accurate value of uncertain system parameter is precisely estimated in a timely manner and then $\mathrm{RK}_{\mathrm{estimator}}$ retains approximation during control(Iplikci 2013). Figure 11 indicates the impact percentage of the control signal produced by $\mathrm{RK}_{\mathrm{controller}}$. The $\mathbf{u}_{\mathrm{RK\text{-}C}}[n]$ term becomes dominant in a very short time in comparison with $\delta\mathbf{u}[n]$ term as given in Figure 11. Since the computational load of an adaptive controler is vital in applications, real time applicability of $\mathrm{RK}_{\mathrm{controller}}$ has been appraised. For this purpose, computational time

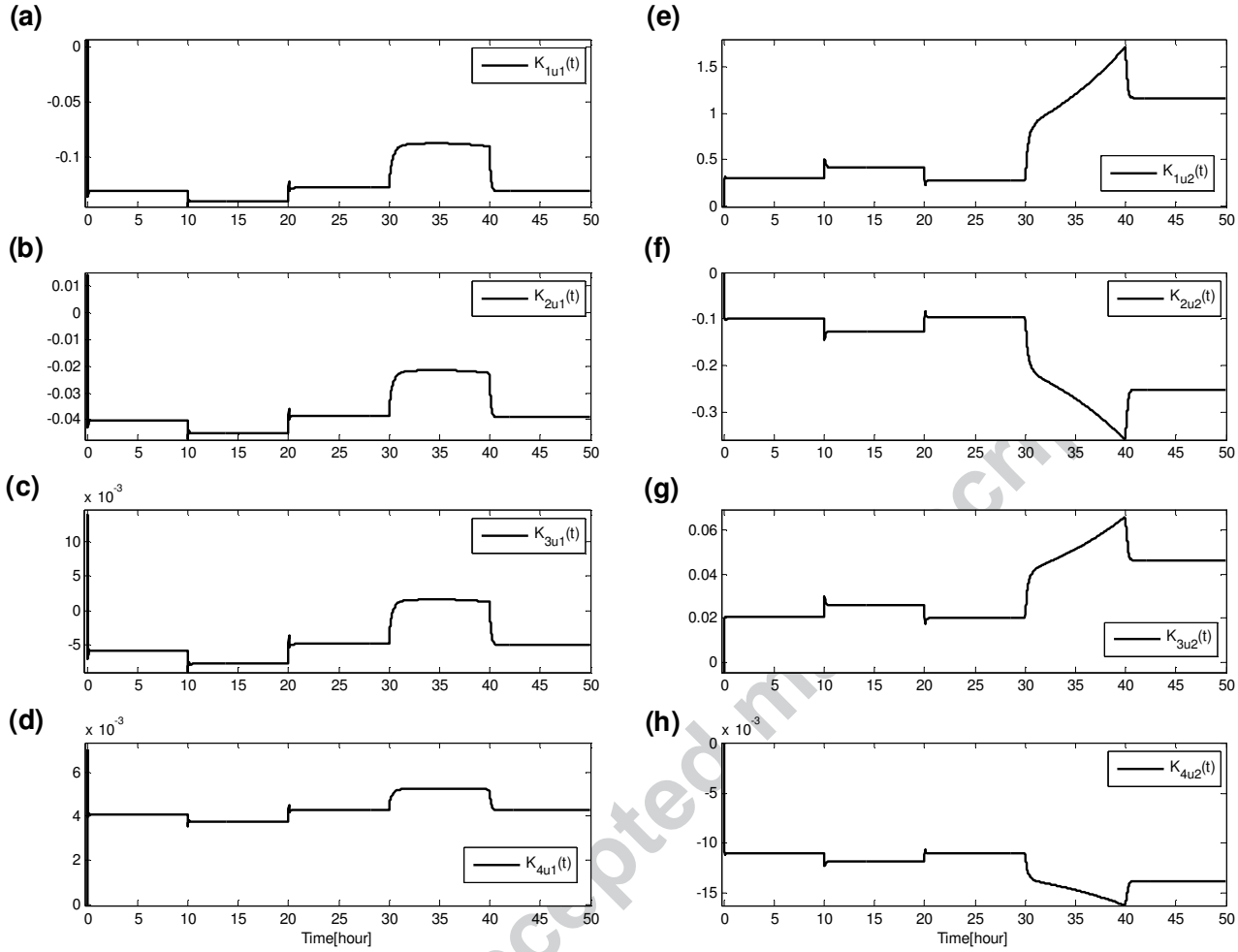

Fig. 4 Adaptation of $RK_{controller}$ parameters.

for every sampling period is computed and registered. The stored execution times for each situation are illustrated in Figure 12. Then, using the maximum values of computation times, Table 2 has been constituted. As can be clearly seen from Figure 12, the computational load of the control algorithm is rarely exposed to the maximum computation time and this occurs only momentarily. The maximum response times for $RK_{controller}$ are less than 7 ms and very smaller than sampling time of VdV system for each case, which enables to deploy $RK_{controller}$ in real time. As can be seen from Figure 12, the momentarily valid execution times can be decreased and enhanced by optimizing and then implementing the control algorithm on convenient hardwares like FPGA. The performance assessment has been performed on a computer with core i7 CPU(2.2 GHz), 8 GB RAM and solid state disc(SSD) features.

4.2 Comparison with Runge-Kutta model based PID Controller

Runge-Kutta model based PID controller presented in (Cetin and Iplikci 2015) has been deployed to assess the control performance evaluation of $RK_{controller}$ in order to obtain a meaningful comparison. The controllers are examined with respect to tracking performances for nominal, measurement noise, parametric uncertainty cases and computational loads of control algorithms. Runge-Kutta model based PID controller is an adaptive controller that combines the robustness of the PID and integration ability of Runge-Kutta method. In Runge-Kutta model based PID, the MIMO PID controller parameters are optimized by Levenberg-Marquardt rule where Runge-Kutta system model is used to constitute predictive model for Jacobian information. The adjustment mechanism is composed of two components: adaptive PID and control signal correction block. Therefore, the task sharing among these two parts is also

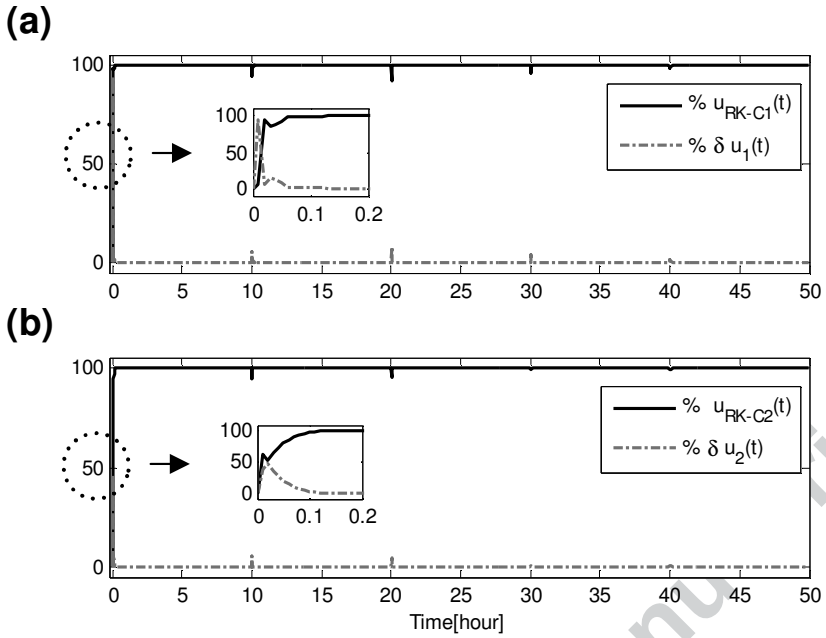

Fig. 5 (**a**) Percentage of task sharing of $\mathbf{u}_{\text{RK-C}}[n]$ and $\delta\mathbf{u}[n]$ for $u_1(t)$ and (**b**) $u_2(t)$ (staircase reference input-nominal case).

Table 2 Execution times [ms] for $\text{RK}_{\text{controller}}$.

Operations \Cases	Noiseless	Noiseless(Sin)	Noisy	Uncertain
EKF State Estimation (RK_{EKF})	1.1953	1.2487	1.2275	1.1684
P_H-step prediction (RK_{model})	1.5726	1.8308	1.5422	1.597
Computation $\mathbf{J}_m$ (RK_{model})	1.0213	0.929	1.0236	0.9724
Computation $\mathbf{J}_c$ ($\text{RK}_{\text{controller}}$)	0.3514	0.3816	0.3685	0.3444
$\delta\mathbf{u}[n]$ term training(LM)	0.0113	0.0737	0.0832	0.095
$\text{RK}_{\text{controller}}$ training	0.0107	0.0572	0.0889	0.093
Control law	0.2271	0.1561	0.2074	0.1904
System response	0.4503	0.369	0.472	0.4215
$\text{RK}_{\text{estimator}}$ Training	———	———	———	0.9715
Miscellaneous Tasks	0.5381	0.4494	0.5463	0.2295
Total loop time	5.3781	5.4955	5.5596	6.0831

examined to reveal and examine which part takes on the control task. For a fair evaluation, the same conditions applied to $\text{RK}_{\text{controller}}$ have been implemented in Runge-Kutta model based PID controller. The control performance of Runge Kutta model based PID controller are depicted in Figures 13,15,17,19. The task sharings are illustrated in Figures 14,16,18,20. The computational load of the mentioned controller is depictured in Figure 21. The controlled system outputs for staircase reference inputs are given in Figure 13. As illustrated in Figure 14, at the beginning of the control, the correction term is more dominant than PID block in system control because of the non-optimal PID parameters. Then, the control task is transfused to adaptive PID part in a very short time. Similarly, the closed-loop system behaviour in response to sinusoidal input, measurement noise and parametric uncertainty cases are shown in Figures 15, 17 and 19. The percentage(%) of tasks sharing for mentioned cases are depictured in Figures 16, 18, 20. The total computational load of Runge-Kutta model based PID controller is illustrated in Figure 21 for all cases. The computational load of each operation in Runge-Kutta model based PID controller algorithm are detailed in Table 3. When computational loads of $\text{RK}_{\text{controller}}$ and Runge-Kutta model based PID are compared, it is clear that $\text{RK}_{\text{controller}}$ has better performance except for uncertainty case. The bar graph in Figure 22 is constructed to compare the tracking performances of $\text{RK}_{\text{controller}}$ and Runge-Kutta model based PID controller using the following performance index for each system output.

$$P_1 = \sum_{n=1}^{\inf} \Big[r_1[n] - y_1[n] \Big]^2$$
$$P_2 = \sum_{n=1}^{\inf} \Big[r_2[n] - y_2[n] \Big]^2 \qquad (29)$$

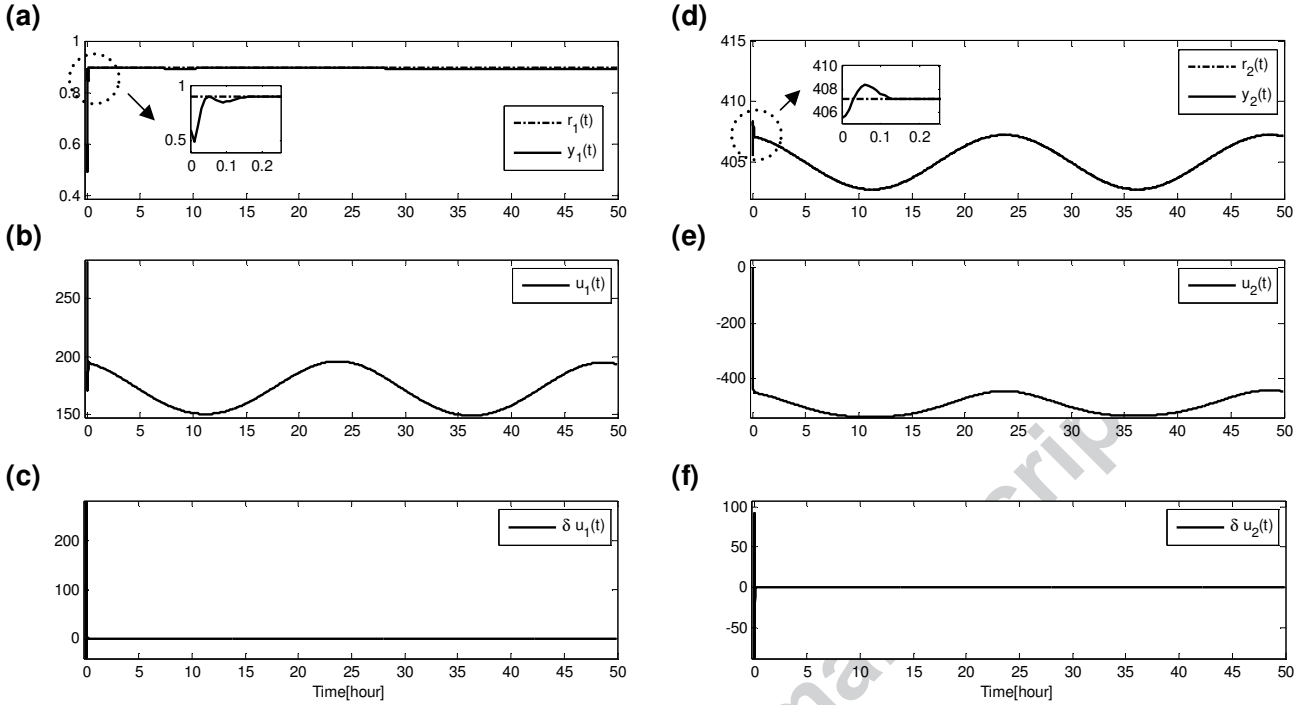

Fig. 6 (**a**,**d**) Tracking behaviours, (**b**,**e**) control signals and (**c**,**f**) correction terms (sinusoidal reference input-nominal case).

Table 3 Execution times [ms] for Runge-Kutta model based PID.

Operations \Cases	Noiseless	Noiseless(Sin)	Noisy	Uncertain
EKF State Estimation (RK_{EKF})	1.2525	1.6257	1.8996	1.0885
P_H-step prediction (RK_{model}) and Computation of **J** (Jacobian Matrix)	3.1561	2.6708	4.1038	2.4304
$RK_{estimator}$ Training	———	———	———	0.9288
Miscellaneous Tasks	2.104	2.3493	4.7528	1.2456
Total loop time	6.5126	6.6458	10.7562	5.6933

The comparison graph in Figure 22 is detailed in Table 4. As given in Figure 22 and Table 4, $RK_{controller}$ has better performance than RK model based PID for $y_1(P_1)$ while RK model based PID has low tracking error for $y_2(P_2)$. The reason for this situation can be considered that RK model based PID has more controller parameters to be adjusted.

5 Conclusion

In this study, a new predictive $RK_{controller}$ methodology is proposed for nonlinear dynamical systems. The main novelty of the architecture is that RK discretization technique is utilized to form adaptive control law by considering the control law is as ODE form. The adjustment mechanism comprises of $RK_{controller}$ and predictive RK_{model} to forecast the future system behaviours. The adjustable parameters are tuned via LM learning law.

The performance of adaptive $RK_{controller}$ is evaluated on VdV system. The robustness of $RK_{controller}$ has been assessed for various situations significant for control systems. Also, the performance of the controller has been compared with Runge-Kutta model based PID controller. The attained results indicate that adaptive $RK_{controller}$ exhibits a forceful characteristic against measurement noise and uncertainties in nonlinear systems. In future studies, it is contemplated to propose new stable adaptive controller and system modeling architectures based on RK integration approach for nonlinear dynamical systems.

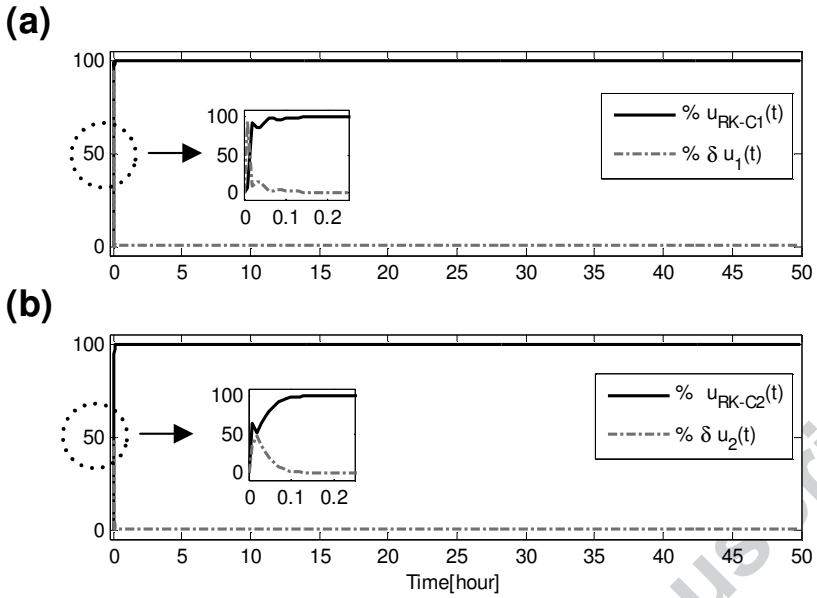

Fig. 7 (**a**) Percentage of task sharing of $\mathbf{u}_{\text{RK-C}}[n]$ and $\delta\mathbf{u}[n]$ for $u_1(t)$ and (**b**) $u_2(t)$(sinusoidal reference input-nominal case).

Table 4 Tracking Performance Comparison for P_1 and P_2 in (29).

Performance Index	P_1				P_2			
Controller	Noiseless (Staircase)	Noiseless (Sinusoidal)	Noisy	Uncertain	Noiseless (Staircase)	Noiseless (Sinusoidal)	Noisy	Uncertain
$\text{RK}_{\text{controller}}$	0.5601	0.3944	0.5285	0.5235	19.1657	12.0708	14.9303	18.2081
RK model based PID	3.2736	2.6165	9.7252	2.9750	7.3507	5.2191	7.3236	10.4055

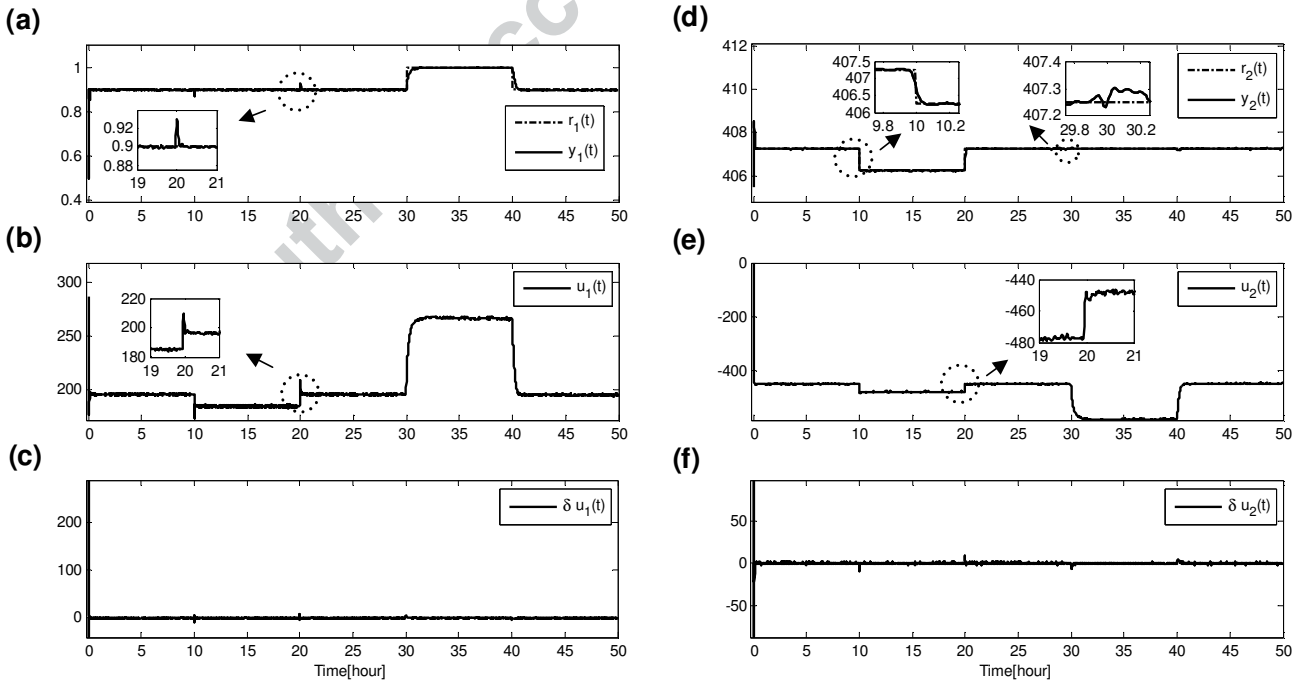

Fig. 8 (**a**,**d**) Tracking behaviours, (**b**,**e**) control signals and (**c**,**f**) correction terms for measurement noise(staircase reference inputs).

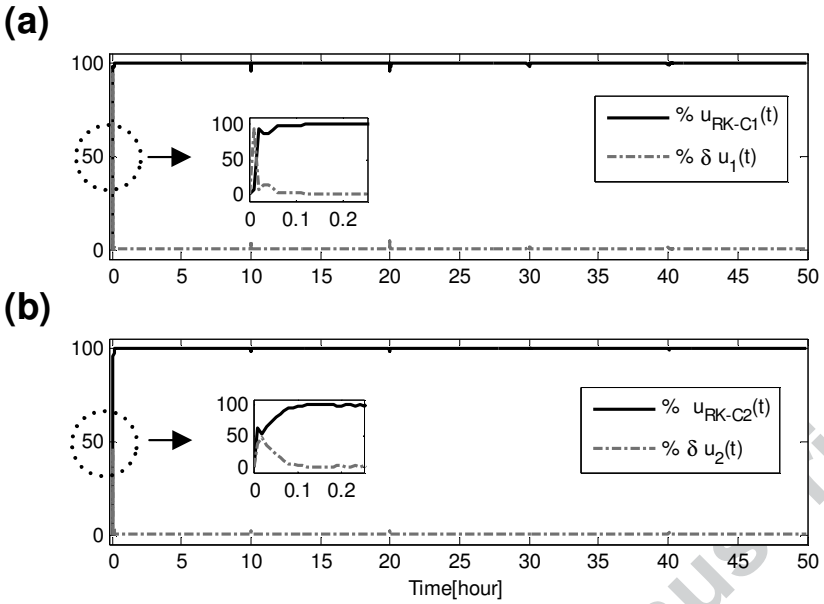

Fig. 9 (**a**) Percentage of task sharing of $\mathbf{u}_{\mathrm{RK\text{-}C}}[n]$ and $\delta\mathbf{u}[n]$ for $u_1(t)$ and (**b**) $u_2(t)$(staircase reference input-measurement noise case).

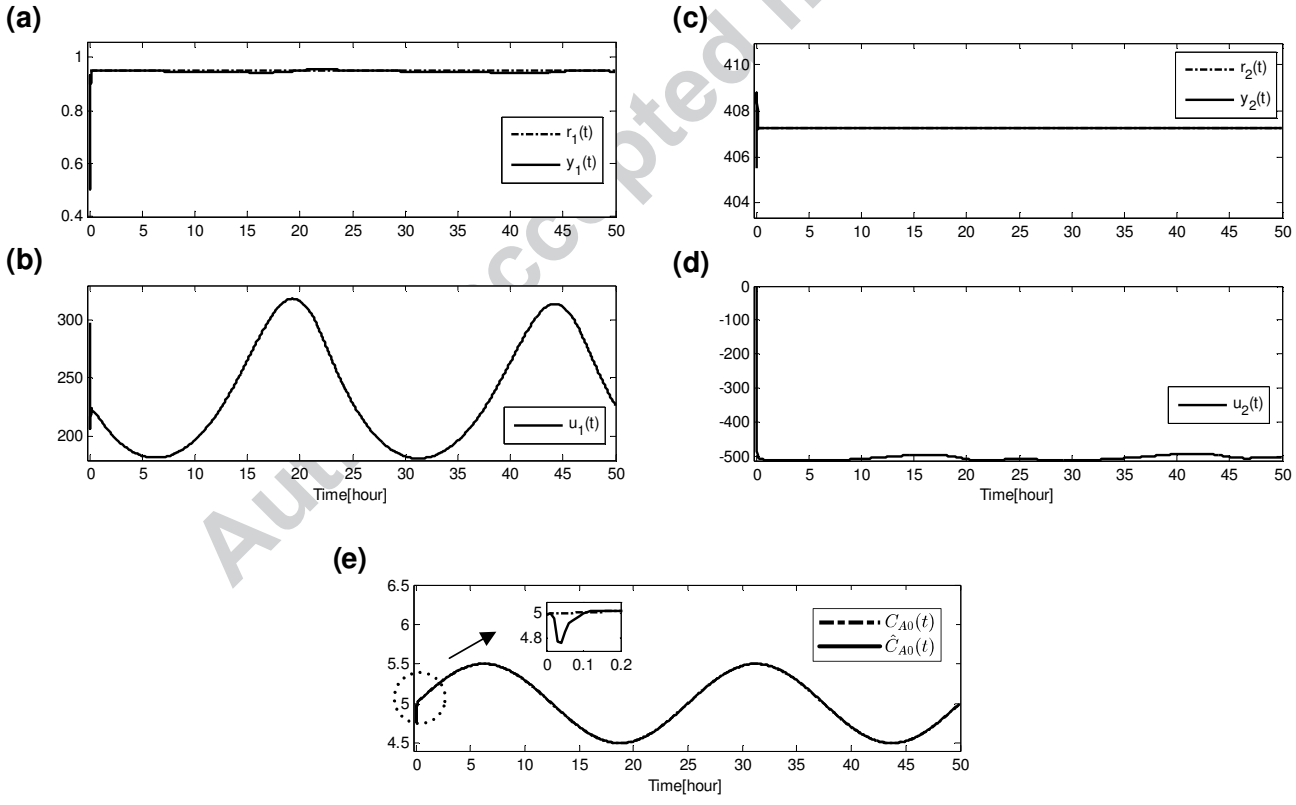

Fig. 10 (**a**,**c**) Tracking behaviours, (**b**,**d**) control signals and (**e**) uncertain parameter ($C_{A0}(t)$) its approximation.

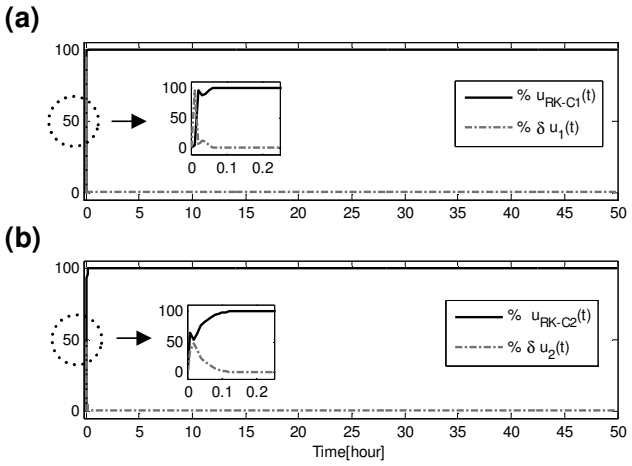

Fig. 11 (**a**) Percentage of task sharing of $\mathbf{u}_{\text{RK-C}}[n]$ and $\delta\mathbf{u}[n]$ for $u_1(t)$ and (**b**) $u_2(t)$(staircase reference input-parametric uncertainty case).

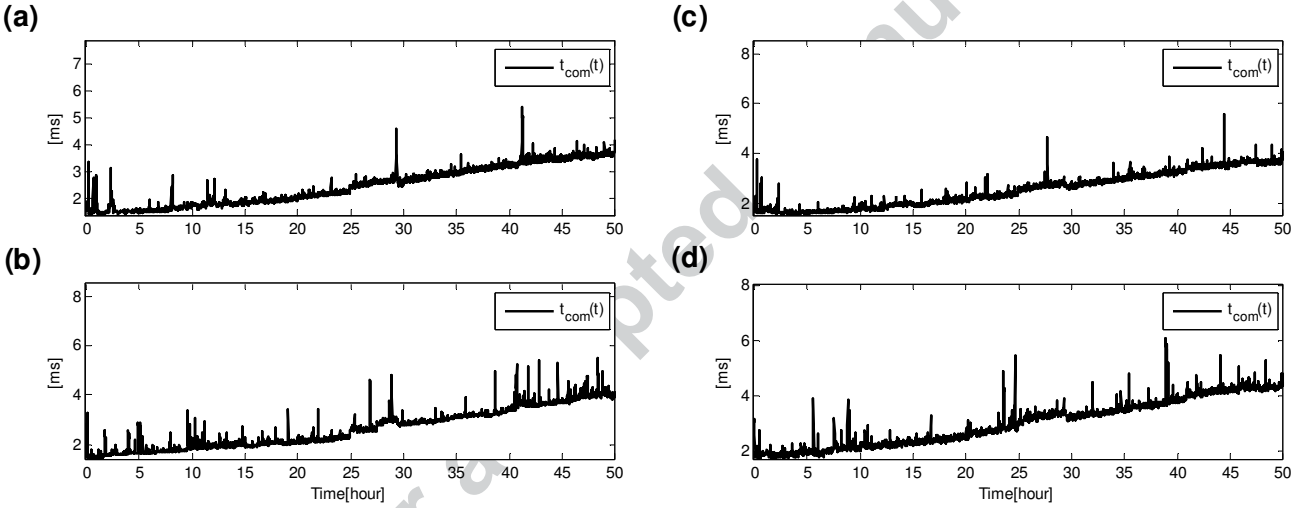

Fig. 12 Computation times of adaptive (nominal case staircase(**a**), nominal case sinusoidal(**b**), measurement noise case staircase(**c**), parametric uncertainty case(**d**)).

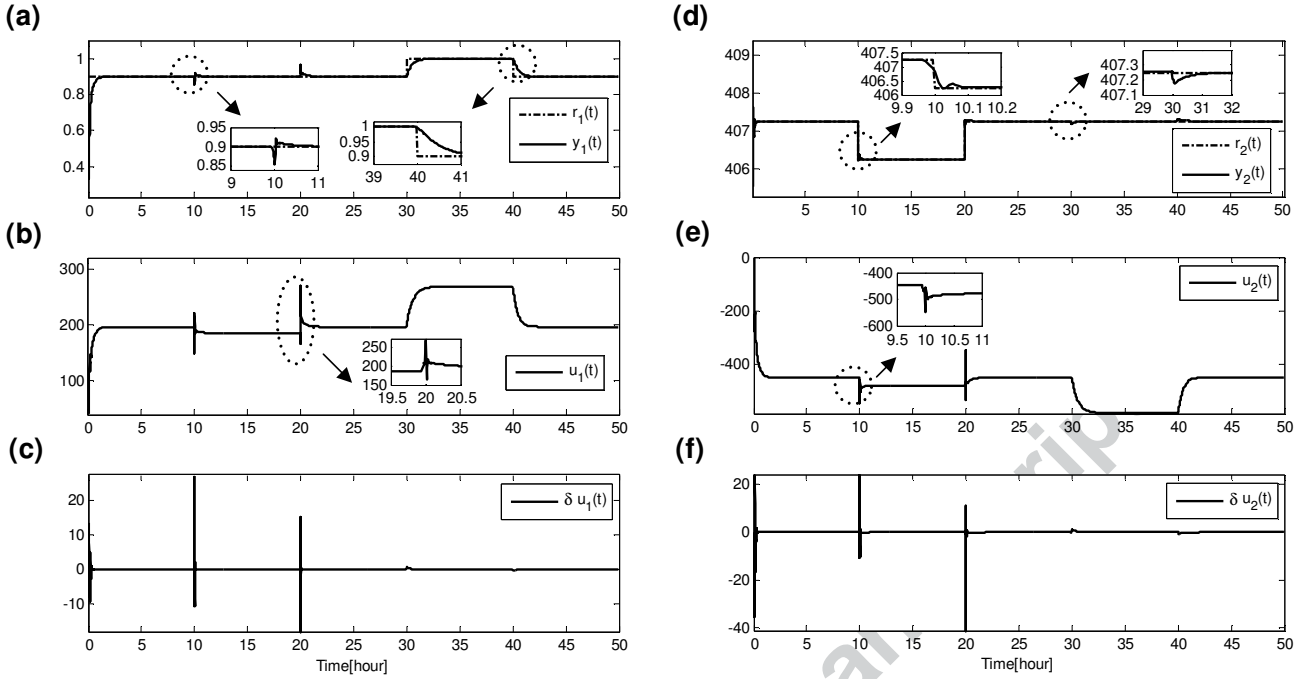

Fig. 13 (**a**,**d**) Tracking behaviours, (**b**,**e**) control signals and (**c**,**f**) correction terms for measurement noise(staircase reference inputs) (Runge-Kutta model based PID).

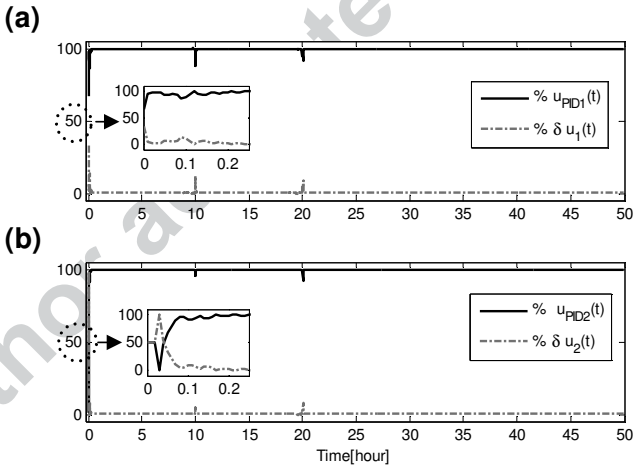

Fig. 14 (**a**) Percentage of task sharing of $\mathbf{u}_{\mathrm{PID}}[n]$ and $\delta\mathbf{u}[n]$ for $u_1(t)$ and (**b**) $u_2(t)$(staircase reference input- nominal case)(Runge-Kutta model based PID).

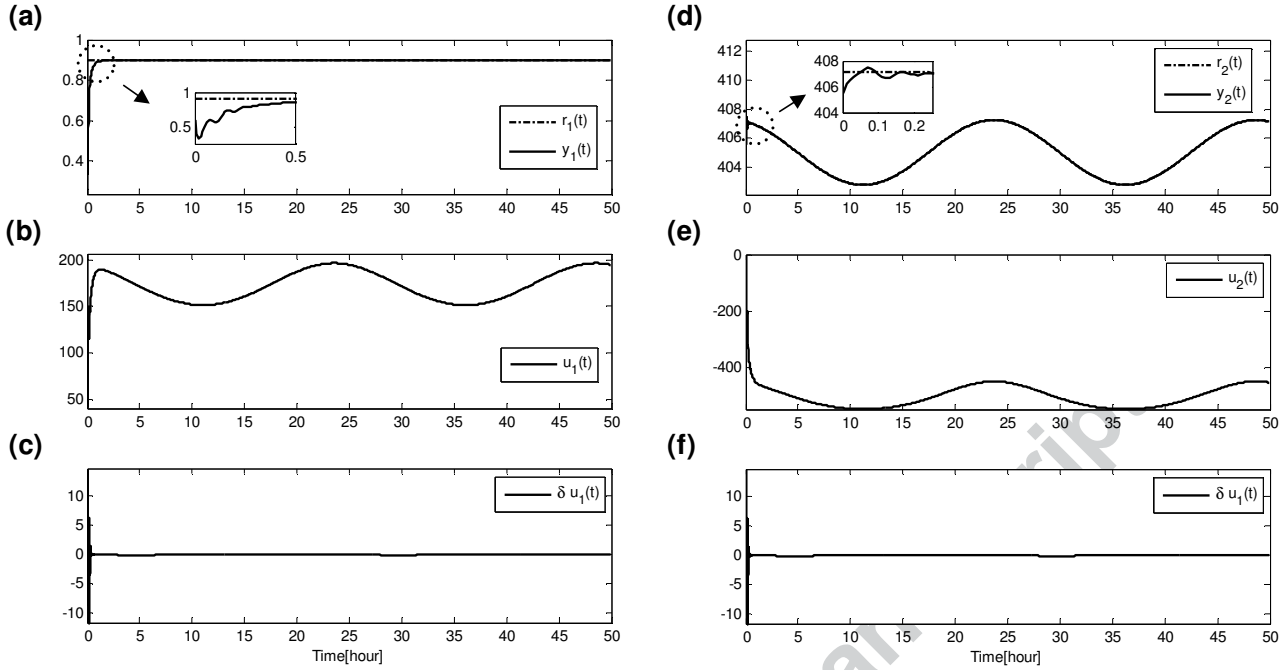

Fig. 15 (**a**,**d**) Tracking behaviours, (**b**,**e**) control signals and (**c**,**f**) correction terms (sinusoidal reference input-nominal case) (Runge-Kutta model based PID)

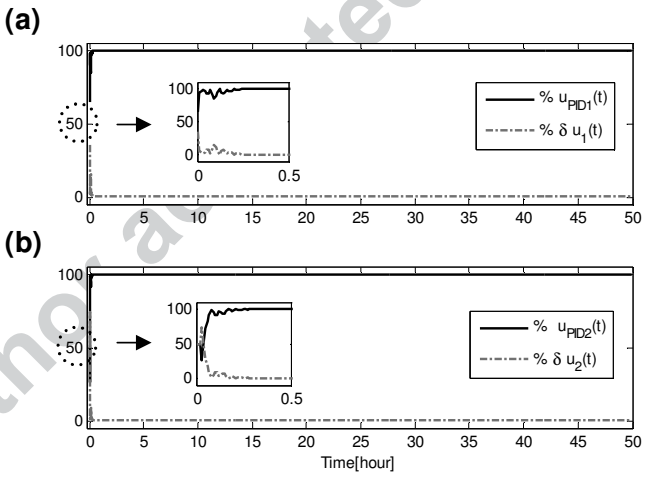

Fig. 16 (**a**) Percentage of task sharing of $\mathbf{u}_{\mathrm{PID}}[n]$ and $\delta\mathbf{u}[n]$ for $u_1(t)$ and (**b**) $u_2(t)$(Sinusoidal reference input- measurement noise case) (Runge-Kutta model based PID).

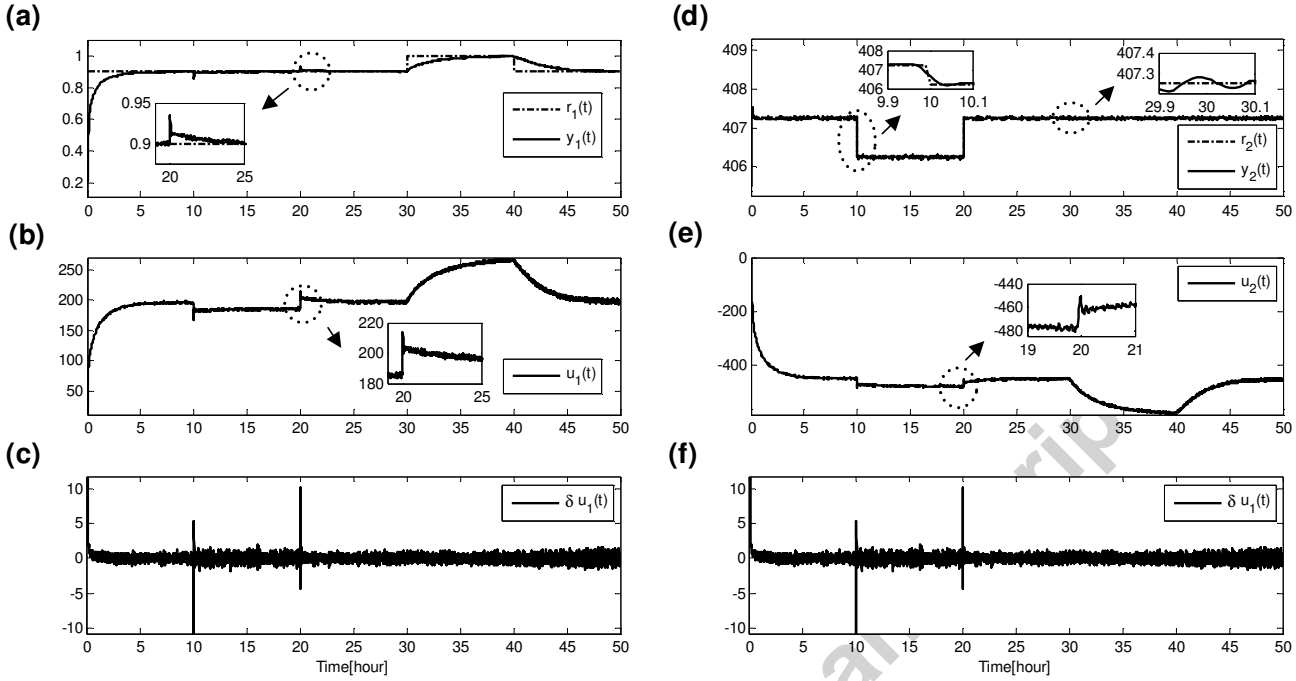

Fig. 17 (**a**,**d**) Tracking behaviours, (**b**,**e**) control signals and (**c**,**f**) correction terms for measurement noise(staircase reference inputs) (Runge-Kutta model based PID).

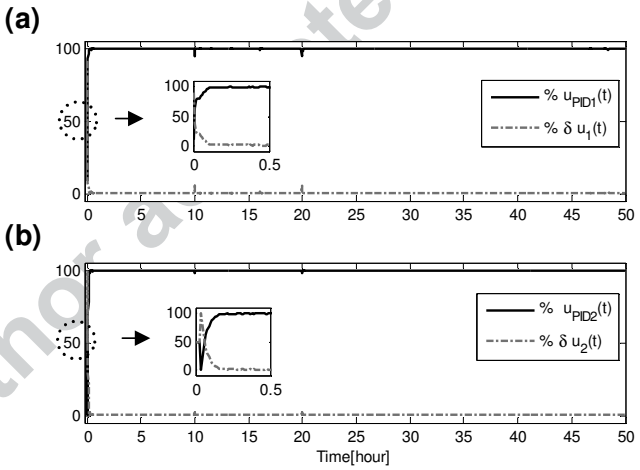

Fig. 18 (**a**) Percentage of task sharing of $\mathbf{u}_{\mathrm{PID}}[n]$ and $\delta\mathbf{u}[n]$ for $u_1(t)$ and (**b**) $u_2(t)$(staircase reference input-measurement noise case) (Runge-Kutta model based PID).

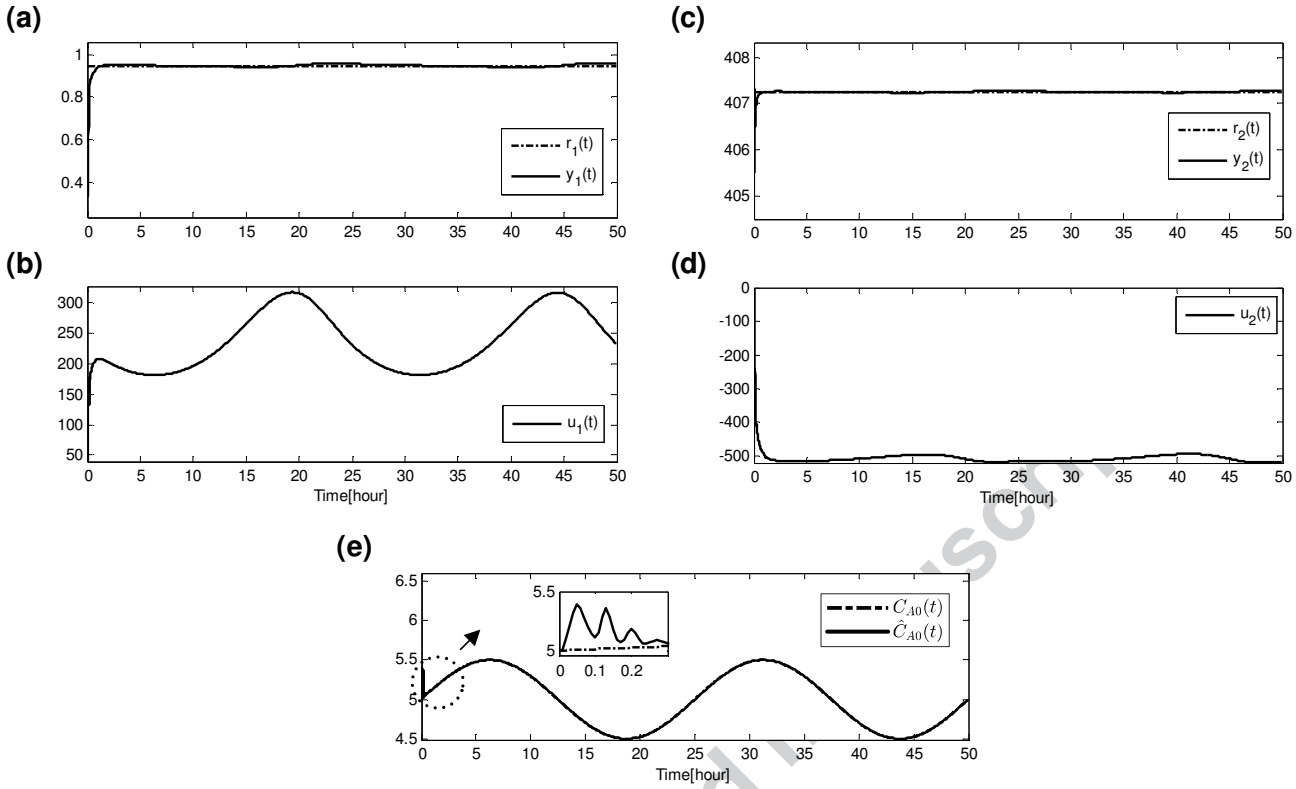

Fig. 19 (**a**,**c**) Tracking behaviours, (**b**,**d**) control signals and (**e**) uncertain parameter ($C_{A0}(t)$) its approximation(Runge-Kutta model based PID).

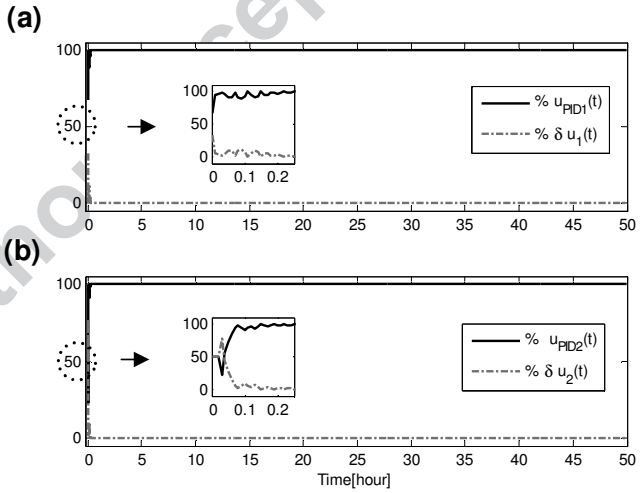

Fig. 20 (**a**) Percentage of task sharing of $\mathbf{u}_{\mathrm{PID}}[n]$ and $\delta\mathbf{u}[n]$ for $u_1(t)$ and (**b**) $u_2(t)$(staircase reference input-parametric uncertainty case) (Runge-Kutta model based PID).

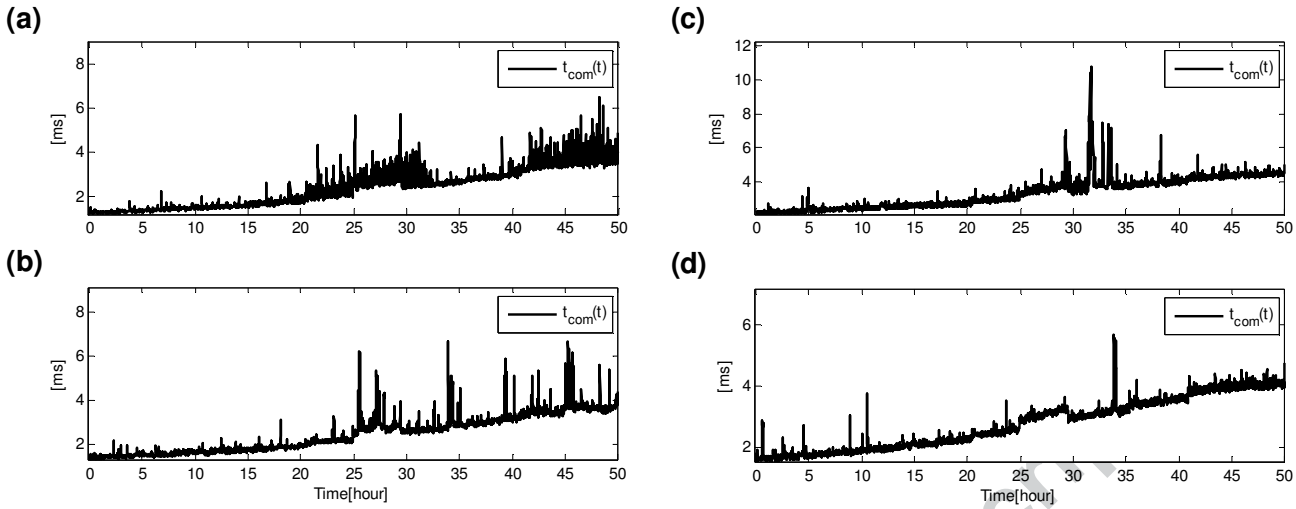

Fig. 21 Computation times of adaptive (nominal case staircase(**a**), nominal case sinusoidal(**b**), measurement noise case staircase(**c**), parametric uncertainty case(**d**))(Runge-Kutta model based PID).

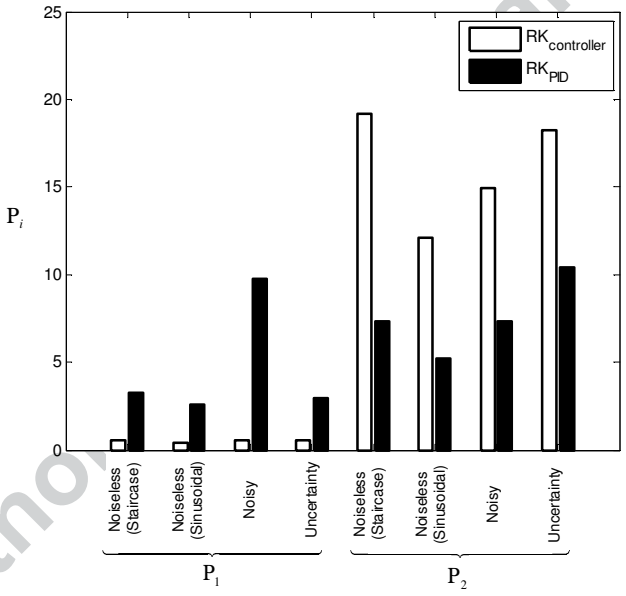

Fig. 22 Tracking error comparison according to index in (29).

Compliance with Ethical Standards

Conflict of Interest

The author declares 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

Aström KJ, Borisson U, Ljung L, Wittenmark B (1977) Theory and applications of self-tuning regulators. Automatica 13(5): 457–476. doi: 10.1016/0005-1098(77)90067-X.

Beyhan S (2013) Runge–Kutta model-based nonlinear observer for synchronization and control of chaotic systems. ISA Transactions 52(4): 501–509. doi: 10.1016/j.isatra.2013.04.005.

Bishr M, Yang YG, Lee G (2000) Self-Tuning Pid Control Using an Adaptive Network–Based Fuzzy Inference System. Intelligent Automation & Soft Computing 6(4): 271–280. doi: 10.1080/10798587.2000.10642795.

Bittinger ML, Ellenbogen DJ, Surgent SA (2012) Calculus and Its Applications. Addison Wesley / Pearson

Bouallègue S, Haggège J, Ayadi M, Benrejeb M (2012) PID-type fuzzy logic controller tuning based on particle swarm optimization. Engineering Applications of Artificial Intelligence 25(3): 484–493. doi: 10.1016/j.engappai.2011.09.018.

Cetin M, Iplikci S (2015) A novel auto-tuning PID control mechanism for nonlinear systems. ISA Transactions 58: 292–308. doi: 10.1016/j.isatra.2015.05.017.

Chen H, Kremling H, Allgöwer F (1995) Nonlinear Predictive Control of a Benchmark CSTR. In: 3rd European Control Conference.

Davis PJ, Rabinowitz P (1984) Methods of Numerical Integration. Academic Press

Denai MA, Palis F, Zeghbib A (2004) ANFIS based modelling and control of non-linear systems : a tutorial. In: 2004 IEEE International Conference on Systems, Man and Cybernetics. The Hague, Netherlands

Efe MO, Kaynak O (1999) A comparative study of neural network structures in identification of nonlinear systems. Mechatronics 9(3): 287–300. doi: 10.1016/S0957-4158(98)00047-6.

Efe MO, Kaynak O (2000) A comparative study of soft-computing methodologies in identification of robotic manipulators. Robotics and Autonomous Systems 30(3): 221–230. doi: 10.1016/S0921-8890(99)00087-1.

Efe MO (2011) Neural Network Based Control. In: The Industrial Electronics Handbook, CRC Press pp. 3-1–3-26.

Engell S, Klatt KU (1993) Nonlinear Control of a Non-Minimum-Phase CSTR. In: American Control Conference. San Francisco

Fasshauer G(2020) Numerical Methods for Differential Equations/Computational Mathematics II Class Notes: Chapter 3-Runge-Kutta Methods [Online]. Available: http://math.iit.edu/ fass/478_578_handouts.html, Accessed on: September 28, 2020

Hagan MT, Demuth HB, Orlando De Jesús (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.

Kravaris C, Niemiec M, Berber R, Brosilow CB (1998) Nonlinear Model-Based Control of Nonminimum-Phase Processes. In: Kravaris C, Berber R(eds) Nonlinear Model Based Process Control, Springer, Dordrecht, pp 115–142

Iplikci S (2006) Online trained support vector machines-based generalized predictive control of nonlinear systems. International Journal of Adaptive Control and Signal Processing 20(10): 599–621. doi: 10.1002/acs.919.

Iplikci S (2010a) 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. doi: 10.1002/rnc.1524.

Iplikci S (2010b) A support vector machine based control application to the experimental three-tank system. ISA Transactions 49(3): 376–386. doi: 10.1016/j.isatra.2010.03.013.

Iplikci S (2013) Runge–Kutta model-based adaptive predictive control mechanism for nonlinear processes. Transactions of the Institute of Measurement and Control 35(2): 166–180. doi:10.1177/0142331212438910

Jang JR (1993) ANFIS: adaptive-network-based fuzzy inference system. IEEE Transactions on Systems, Man, and Cybernetics 23(3): 665–685. doi: 10.1109/21.256541.

Jørgensen JB (2007) A Critical Discussion of the Continuous-Discrete Extended Kalman Filter. In: European Congress of Chemical Engineering - 6.

Kulikov GY, Kulikova MV (2014) Accurate state estimation in the Van der Vusse reaction. In: IEEE Conference on Control Applications (CCA). Nice

Niemiec MP, Kravaris C (2003) Nonlinear model-state feedback control for nonminimum-phase processes. Automatica 39(7):1295–1302. doi:10.1016/S0005-1098(03)00103-1

Pham DT, Karaboga D (1999) Self-tuning fuzzy controller design using genetic optimisation and neural network modelling. Artificial Intelligence in Engineering 13(2): 119–130. doi: 10.1016/S0954-1810(98)00017-X.

Psaltis D, Sideris A, Yamamura AA (1988) A multilayered neural network controller. IEEE Control Systems Magazine 8(2): 17–21. doi: 10.1109/37.1868.

Roberts CE (2010) Ordinary Differential Equations: Applications, Models and Computing. CRC Press

Saerens M, Soquet A (1991) Neural controller based on back-propagation algorithm. IEE Proceedings F - Radar and Signal Processing 138(1): 55–62. doi: 10.1049/ip-f-2.1991.0009.

Sharkawy AB (2010) Genetic fuzzy self-tuning PID controllers for antilock braking systems. Engineering Applications of Artificial Intelligence 23(7): 1041–1052. doi: 10.1016/j.engappai.2010.06.011.

Tanomaru J, Omatu S (1992) Process control by on-line trained neural controllers. IEEE Transactions on Industrial Electronics 39(6): 511–521. doi: 10.1109/41.170970.

Uçak K, Günel GÖ (2016) An adaptive support vector regressor controller for nonlinear systems. Soft Computing 20(7): 2531–2556. doi: 10.1007/s00500-015-1654-0.

Uçak K, Günel GÖ (2017) Generalized self-tuning regulator based on online support vector regression. Neural Computing and Applications 28(1): 775–801. doi: 10.1007/s00521-016-2387-4.

Uçak K (2019) A Runge–Kutta neural network-based control method for nonlinear MIMO systems. Soft Computing 23(17):7769–7803.

Uçak K (2020) A Novel Model Predictive Runge–Kutta Neural Network Controller for Nonlinear MIMO Systems. Neural Processing Letters 51: 1789—1833.

Vojtesek J, Dostál P (2010) Adaptive Control of Chemical Reactor. In: International Conference Cybernetics and Informatics. VYŠNÁ BOCA, Slovak Republic

Wang YJ, Lin CT (1998) Runge-Kutta neural network for identification of dynamical systems in high accuracy. IEEE Transactions on Neural Networks 9(2): 294–307. doi: 10.1109/72.661124.

Zhang Y, Sen P, Hearn GE (1995) An on-line trained adaptive neural controller. IEEE Control Systems Magazine 15(5): 67–75. doi: 10.1109/37.466260.