APPLICATION OF SOFT COMPUTING

A novel adaptive Runge–Kutta controller for nonlinear dynamical systems

Kemal Uçak

Accepted: 5 April 2021 / Published online: 5 May 2021
© The Author(s), under exclusive licence to Springer-Verlag GmbH Germany, part of Springer Nature 2021

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” attributed to Galileo Galilei is possibly the best aphorism 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 architectures in order to ingen 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. 2000; Denai et al. 2004) and SVR (Uçak and Günel 2016, 2017; Iplikci 2010a, 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
modelling 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 realize. Therefore, the RK-based identification with low computational load in comparison with soft computing methods, given in Iplikci (2013) and Uçak (2019), is employed to identify non-linear system dynamics.

A variety of controller architectures based on RK-system model have been introduced. 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. Cetin and Iplikci (2015) deployed the predictive RK system model to derive adjustment rules for an adaptive MIMO PID controller. The proposed autotuning mechanism for MIMO PID combines the robustness and fast convergence features of PID and MPC (Cetin and Iplikci 2015). 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 computed. Accurate representation of system behaviours against adjustment to be realized in control vector. The possible system behaviours against adjustment in control parameters.

The most important feature that radically distinguishes this study from previous (Uçak 2019, 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) and 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 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 fourth-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 modelling technique introduced by 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 (2013), respectively.

In Sect. 2, adaptive mechanism for RK controller is presented. The adjustment rules and control algorithm for proposed RK controller are detailed in Sect. 3. The evaluation of the introduced adaptive RK controller is scrutinized on a nonlinear VdV system in Sect. 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 performance 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 Sect. 5.

### 2 Adaptive Runge–Kutta controller

In control systems, approximating the dynamic behaviour 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 insert 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, 1999), ANFIS (Denai et al. 2004; Jang 1993), SVR (Iplikci 2010a, b, 2006), 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 Fig. 1 where $R$ expresses 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 scrutinized: 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 RKcontroller and system model is RKmodel 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 RKcontroller produces a candidate $(u^*[n])$ signal as

$$
u^*[n] = u[n - 1] + \left\{\frac{1}{6}[K_{1u}[n - 1] + 2K_{2u}[n - 1] + 2K_{3u}[n - 1] + K_{4u}[n - 1]]ight.$$

$$+ 2K_{3u}[n - 1] + K_{4u}[n - 1]\right.$$

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

$$K = \begin{bmatrix} K_{1u} & K_{2u} & K_{3u} & K_{4u} \end{bmatrix}$$

Then, by sequentially applying the obtained $u^*[n]$ to RKmodel, the system behaviour and system Jacobian required to adjust controller parameters $(K)$ can be acquired. RKmodel 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 $(\theta)$ 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 modelling techniques or deviation in system parameters $(\theta)$, system parameters $(\theta)$ 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 $(\theta)$. By using raw RK model, RK_EKF, and RK_estimator, RKmodel can be constituted to forecast $P_H$—step future system action with high accuracy. The detailed information about subblock of RKmodel is given in Iplikci (2013), Uçak (2019, 2020).

### 2.2 Training Phase

The adjustment laws to attain the feasible RKcontroller parameters can be derived via objective function in (3)

$$F(u[n], \hat{e}_q) = \sum_{q=1}^{Q} \sum_{p=1}^{P_H} \beta_q [\hat{e}_q[n + p]]^2$$

$$+ \sum_{r=1}^{R} \lambda_r \left[u_r[n] - u_r[n - 1]\right]^2$$

where $\hat{e}_q[n + p] = \bar{r}_q[n + p] - \hat{y}_q[n + p]$. $P_H$ indicates the prediction horizon, $\beta_q$ and $\lambda_r$ denote penalty coefficients to hamper chattering in control signals. LM optimization rule can be deployed to optimize the RKcontroller parameters $(K)$ as follows:

$$K^\text{new} = K^\text{old} + \Delta K, \quad \Delta K = -[J^T J + \mu I]^{-1} J^T \hat{e}$$

where $J$ is given as

$$J = \begin{bmatrix} \frac{\partial e_1[n+1]}{\partial K_{1u}[n-1]} & \cdots & \frac{\partial e_1[n+1]}{\partial K_{4u}[n-1]} \\
\vdots & \ddots & \vdots \\
\frac{\partial e_Q[n+K]}{\partial K_{1u}[n-1]} & \cdots & \frac{\partial e_Q[n+K]}{\partial K_{4u}[n-1]} \\
\sqrt{\lambda_1} \frac{\partial u_1[n]}{\partial K_{1u}[n-1]} & \cdots & \sqrt{\lambda_1} \frac{\partial u_1[n]}{\partial K_{4u}[n-1]} \\
\vdots & \ddots & \vdots \\
\sqrt{\lambda_R} \frac{\partial u_R[n]}{\partial K_{1u}[n-1]} & \cdots & \sqrt{\lambda_R} \frac{\partial u_R[n]}{\partial K_{4u}[n-1]} \end{bmatrix}$$

and $\hat{e}$ is error vector.
The Jacobian matrix \( J \) can be decomposed into two parts representing the sensitivity of the system \( J_m \) and controller \( J_c \) depending on their adjustable inputs as in (7).

\[
\hat{e} = \left[ \begin{array}{c}
\beta_1 \hat{e}_1[n + 1] \\
\vdots \\
\beta_Q \hat{e}_Q[n + P_H] \\
\sqrt{\lambda_1 \Delta u_1[n]} \\
\vdots \\
\sqrt{\lambda_R \Delta u_R[n]}
\end{array} \right]
\]

\[
= \left[ \begin{array}{c}
\beta_1 [r_1[n + 1] - \hat{y}_1[n + 1]] \\
\vdots \\
\beta_Q [r_Q[n + P_H] - \hat{y}_Q[n + P_H]] \\
\sqrt{\lambda_1 [u_1[n] - u_1[n - 1]]} \\
\vdots \\
\sqrt{\lambda_R [u_R[n] - u_R[n - 1]]}
\end{array} \right]
\]

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

\[
F(u[n] + \delta u[n]) \approx F(u[n]) + \frac{\partial F(u[n])}{\partial u[n]}\delta u[n] + \frac{1}{2} \frac{\partial^2 F(u[n])}{\partial^2 u[n]}(\delta u[n])^2
\]

For optimality of \(\delta u[n]\) (Ipliçci 2010a, 2013)

\[
\frac{\partial F(u[n])}{\partial u[n]} + \frac{\partial^2 F(u[n])}{\partial^2 u[n]}\delta u[n] = 0
\]

Thus, \(\delta u[n]\) term is concluded as (3) (Ipliçci 2010a, 2013)

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

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

\[
\frac{\partial^2 F(u[n])}{\partial^2 u[n]} = 2J^T_m \hat{e}
\]

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

\[
\frac{\partial^2 F(u[n])}{\partial^2 u[n]} = 2J^T_m J_m
\]

Thus, Eq. (10) can be re-expressed as:

\[
\delta u[n] = -[J^T_m J_m]^{-1}J^T_m \hat{e}
\]

### 2.3 Control Phase

Then, employing the trained \(K_{\text{controller}}\) parameters \((K_{\text{new}})\) obtained in (4) and suboptimal correction term \(\delta u[n]\), the updated new control action \((u'[n] = u[n] + \delta u[n])\) 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 \(K_{\text{controller}}\) is detailed in the next section.

### 3 Adaptive \(K_{\text{controller}}\)

#### 3.1 An overview of \(K_{\text{controller}}\)

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

\[
u(t) = f(u(t), \Omega(t))
\]

with the initial condition \(u(0) = u_0\) (Uçak 2019). If it is assumed that \(f_e\) is known, one-step ahead control signal vector can be computed via fourth-order RK ODE solver given in (15):

\[
u[n] = u[n-1] + \frac{1}{6}[K_{1u}[n-1] + 2K_{2u}[n-1] + 2K_{3u}[n-1] + K_{4u}[n-1]]
\]

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

\[
\begin{align*}
K_{1u}[n-1] &= T_e f_e(x_1[n-1], \Omega[n])\big|_{x_1[n-1]=u[n-1]} \\
K_{2u}[n-1] &= T_e f_e(x_2[n-1], \Omega[n])\big|_{x_5[n-1]=u[n-1]+\frac{1}{2}K_{1u}[n-1]} \\
K_{3u}[n-1] &= T_e f_e(x_3[n-1], \Omega[n])\big|_{x_5[n-1]=u[n-1]+\frac{1}{2}K_{2u}[n-1]} \\
K_{4u}[n-1] &= T_e f_e(x_3[n-1], \Omega[n])\big|_{x_5[n-1]=u[n-1]+\frac{1}{2}K_{3u}[n-1]}
\end{align*}
\]
\[ \begin{align*}
K_{4u}[n-1] &= T_c f_c(x_4[n-1], \Omega[n]) \bigg|_{x_4[n-1]=u[n-1]+K_{3u}[n-1]} \\
&= T_c f_c(x_4[n-1], \Omega[n]) \bigg|_{x_4[n-1]=u[n-1]+K_{3u}[n-1]} \\
\end{align*} \]  

(16)

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

\[ \begin{align*}
\mathbf{u}[n] &= f_{RK}(\mathbf{u}, \mathbf{K}) = \mathbf{u}[n-1] + \frac{1}{6} K_{1u}[n-1] \\
&+ \frac{2}{6} K_{2u}[n-1] + \frac{2}{6} K_{3u}[n-1] \\
&+ \frac{1}{6} K_{4u}[n-1] \\
\end{align*} \]  

(17)

in which \( K_{1u} K_{2u} K_{3u} K_{4u} \) are unknown and adjustable parameters of the RK controller. The structure of the RK controller is illustrated in Fig. 2.

### 3.2 Adjustment laws for RKcontroller

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

\[ \begin{align*}
\mathbf{K} &= [K_{1u} K_{2u} K_{3u} K_{4u}]^T \\
\end{align*} \]  

(18)

Thus, using LM optimization rule in (3), RKcontroller parameters can be optimized as given in (4–7). As given in (7), the Jacobian matrix can be partitioned into two part as (\( J = J_m J_c \)) where \( J_m \) is system sensitivity and \( J_c \) denotes RKcontroller sensitivity. The construction of \( J_m \) matrix is detailed in Iplikci (2013), Uçak (2019, 2020). In order to form \( J_c \) matrix in (7), it is essential to derive the terms. Employing chain rule, the mentioned terms are expressed in (19–23):

\[ \begin{align*}
\frac{\partial u[n]}{\partial K_{4u}[n-1]} &= \frac{1}{6} \\
\end{align*} \]  

(19)

![Fig. 2](image-url)

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

© Springer
A novel adaptive Runge–Kutta controller for nonlinear dynamical systems

\[
\begin{align*}
\frac{\partial u[n]}{\partial K_{3u}[n-1]} &= \frac{\partial u[n]}{\partial K_{3u}[n-1]} + \frac{\partial u[n]}{\partial K_{3u}[n-1]} \frac{\partial K_{3u}[n-1]}{\partial K_{3u}[n-1]}
\end{align*}
\]

\[
\frac{\partial u[n]}{\partial K_{2u}[n-1]} = \begin{bmatrix}
\frac{\alpha}{2} & \frac{\alpha}{2} & \frac{\alpha}{2} \\
\frac{\alpha}{2} & \frac{\alpha}{2} & \frac{\alpha}{2} \\
\frac{\alpha}{2} & \frac{\alpha}{2} & \frac{\alpha}{2}
\end{bmatrix}
\begin{bmatrix}
T_c \frac{\alpha}{\lambda} f(x[n-1], \Omega[n]) \\
T_c \frac{\alpha}{\lambda} f(x[n-1], \Omega[n]) \\
T_c \frac{\alpha}{\lambda} f(x[n-1], \Omega[n])
\end{bmatrix}
\]

\[
\frac{\partial u[n]}{\partial K_{1u}[n-1]} = \begin{bmatrix}
\frac{\alpha}{2} & \frac{\alpha}{2} & \frac{\alpha}{2} \\
\frac{\alpha}{2} & \frac{\alpha}{2} & \frac{\alpha}{2} \\
\frac{\alpha}{2} & \frac{\alpha}{2} & \frac{\alpha}{2}
\end{bmatrix}
\begin{bmatrix}
T_c \frac{\alpha}{\lambda} f(x[n-1], \Omega[n]) \\
T_c \frac{\alpha}{\lambda} f(x[n-1], \Omega[n]) \\
T_c \frac{\alpha}{\lambda} f(x[n-1], \Omega[n])
\end{bmatrix}
\]

where

\[
\begin{align*}
\frac{\partial K_{3u}[n-1]}{\partial K_{3u}[n-1]} &= \frac{\partial K_{3u}[n-1]}{\partial K_{3u}[n-1]} \frac{\partial f(x[n-1], \Omega[n])}{\partial x[n-1]} \\
\frac{\partial K_{2u}[n-1]}{\partial K_{2u}[n-1]} &= \frac{\partial K_{2u}[n-1]}{\partial K_{2u}[n-1]} \frac{\partial f(x[n-1], \Omega[n])}{\partial x[n-1]} \\
\frac{\partial K_{1u}[n-1]}{\partial K_{1u}[n-1]} &= \frac{\partial K_{1u}[n-1]}{\partial K_{1u}[n-1]} \frac{\partial f(x[n-1], \Omega[n])}{\partial x[n-1]}
\end{align*}
\]

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

**Assumption 1** For convenience, it is assumed that the relation between \( u[n-1] \) and \( f \) is known in the following form:

\[
f(u[n-1], \Omega[n]) = u'n[n-1] + f \text{ unknown terms}(r, y)
\]

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

\[
\frac{\partial f(u[n-1], \Omega[n])}{\partial u[n-1]} = d'\text{unknown terms}(r, y) + K_{3u}[n-1]
\]

The most prominent feature of the introduced architecture is that under the assumption that the controller architecture
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              | $k_{10}$ | $1.287 \times 10^{12} \text{(h}^{-1})$ |
| Reaction $k_2$: Collision factor              | $k_{20}$ | $1.287 \times 10^{12} \text{(h}^{-1})$ |
| Reaction $k_3$: Collision factor              | $k_{30}$ | $9.043 \times 10^9 \text{(h}^{-1}/\text{mol})$ |
| Reaction $k_1$: Activation energy            | $E_1$  | 9758.3 (K)         |
| Reaction $k_2$: Activation energy            | $E_2$  | 9758.3 (K)         |
| Reaction $k_3$: Activation energy            | $E_3$  | 8560.0 (K)         |
| Reaction $k_1$: Enthalpy                     | $\Delta H_1$ | 4.2 (kJ/mol)       |
| Reaction $k_2$: Enthalpy                     | $\Delta H_2$ | $-11 \text{(kJ/mol)}$ |
| Reaction $k_3$: Enthalpy                     | $\Delta H_3$ | $-41.85 \text{(kJ/mol)}$ |
| Substance A: Enthalpy                         | $C_{A0}$ | 5.0 (mol/l)        |
| Feed temperature                              | $T_0$  | 403.15 (K)         |
| Density                                       | $\rho$ | 0.9342 (kg/l)      |
| Heat capacity                                 | $C_p$  | 3.01 (kJ/kg K)     |
| Reactor volume                                | $V$    | 10.0 (l)           |

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 analysed. 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

![Fig. 3](image-url)

Fig. 3  a, d Tracking behaviours, b, e control signals and c, f correction terms for nominal case

© Springer
A novel adaptive Runge–Kutta controller for nonlinear dynamical systems

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, 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) and 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 neural network.

4 Simulation results

RKcontroller performance has been evaluated on a nonlinear VdV system. However, it is possible to deploy the introduced architecture to a wide variety of control systems to overcome characteristics rarifying control task such as nonlinearity, instability, etc. In order to better reveal the efficiency of RKcontroller, competencies of RKcontroller such as tracking and robustness 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 2010b, 2013; Cetin and Iplikci 2015). Due to its non-minimum-phase behaviour and harsh nonlinearity (Iplikci 2013; Uçak 2019, 2020), it is significant to be controlled adaptively in order to attune the occurring divergent behaviours. The reaction scheme of VdV is given as follows:

\[
\begin{align*}
A & \xrightarrow{k_1} B \xrightarrow{k_2} C \\
2A & \xrightarrow{k_3} D
\end{align*}
\]

\( (26) \)
where cyclopentadiene (A) is the inlet reactant, cyclopentenol (B) indicates the intended compound, dicyclopentadiene (D) is produced by Diels–Alder reaction, and cyclopentanediol (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 $B$ from $A$ (Uçak 2019; Engell and Klatt 1993). The reaction in (26) is detailed via corresponding chemical compounds as follows:

\[
\begin{align*}
C_5H_6 & \xrightarrow{+H_2O(k_1)} C_5H_7OH \\
C_5H_7OH & \xrightarrow{+H_2O(k_2)} C_5H_8(OH)_2 \\
2C_5H_6 & \xrightarrow{k_3} C_{10}H_{12}
\end{align*}
\]  

(27)

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

![Figures 5 and 6](image-url)
A novel adaptive Runge–Kutta controller for nonlinear dynamical systems

In Table 1 (Iplikci 2013; Uçak 2019; Chen et al. 1995; Vojtechsek and Dostál 2010; Niemiec and Kravaris 2003; Kravaris et al. 1998).

The RKcontroller 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, 2020), Chen et al. (1995), Vojtechsek and Dostál (2010), Niemiec and Kravaris (2003) and Kravaris et al. (1998).

### 4.1 Performance evaluation of RKcontroller

Expecting a controller that fails in nominal conditions to succeed in difficult cases is an unrealistic anticipation. Therefore, firstly, the performance of the RKcontroller is tested for nominal case where all the information about the system is exactly 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 Fig. 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. RKcontroller generates the optimal control signals illustrated in Fig. 3b, e. The optimal control signal applied to the system is composed of two terms (Iplikci 2013; Uçak 2019, 2020). These are \( u_{RK}\)-\( C \[n \] \) term which is the control signal produced by only RKcontroller and the correction term \( \delta u[n] \) used to improve the transient state behaviour of the controlled system since the adjusted controlled parameters may not be optimal at tran-
sient state (Iplikci 2010a; Uçak 2019, 2020). Therefore, the task share between $u_{RK} - C_n$ and correction terms $\delta u[n]$ is also examined in this paper. The correction terms that provide large contribution to control signal, especially in transient state, are also given in Fig. 3c, f. If it is focused on the occurring abrupt changes at $[30, 40]$ h and $[10, 20]$ h, the control signals are successfully updated to comply with the arising strong coupling among $C_B$ and reactor temperature $T$. RKcontroller successfully tracks the desired signals as illustrated in Fig. 3a, d, and $u_i(t)$ terms produced by RKcontroller are also demonstrated in Fig. 3b, c, e, f.

In order to show the adaptation of the RKcontroller parameters, the convergence and also adjustment of control signals are given in Fig. 4. The adjustable parameters of RKcontroller converge to their optimal values on short notice. It is aimed to apperceive the duty share between RKcontroller($u_{RK} - C_n$) and correction term($\delta u[n]$). The duty share percentages of $u_{RK} - C_n$ and $\delta u[n]$ terms are illustrated in Fig. 5. As can be seen from Fig. 5a, b, $u_{RK} - C_n$ and $\delta u[n]$ are carrying out control task together. Especially in transient states, $\delta u[n]$ term demonstrates its influence on the control mechanism. Therefore, $\delta u[n]$ is a vital part of the adjustment mechanism. As can be seen from Fig. 5a, b, RKcontroller performs dominant behaviour in comparison with $\delta u[n]$ term. It is explicitly seen that $\delta u[n]$ immediately hands over the control to RKcontroller, and RKcontroller 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

---

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

**Fig. 11** a Percentage of task sharing of $u_{RK} - C_n$ and $\delta u[n]$ for $u_1(t)$ and b $u_2(t)$ (staircase reference input-parametric uncertainty case)
A novel adaptive Runge–Kutta controller for nonlinear dynamical systems

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

Table 2 Execution times (ms) for RKcontroller

| 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 J_m (RK_model)              | 1.0213    | 0.929           | 1.0236 | 0.9724    |
| Computation J_c (RKcontroller)          | 0.3514    | 0.3816          | 0.3685 | 0.3444    |
| ∆υ[n] term training (LM)                | 0.0113    | 0.0737          | 0.0832 | 0.095     |
| RKcontroller 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    |
| RKestimator Training                    | –         | –               | –      | 0.9715    |
| Miscellaneous Tasks                     | 0.5381    | 0.4494          | 0.5463 | 0.2295    |
| Total loop time                         | 5.3781    | 5.4955          | 5.5596 | 6.0831    |

RKcontroller, uRK-C[n] and ∆υ[n] for u_i(t) terms is shown in Fig. 6. The control failover of uRK-C[n] and ∆υ[n] are indicated in Fig. 7. The percentage of the control task sharing between RKcontroller(uRK-C[n]) and ∆υ[n] term for sinusoidal inputs is shown in Fig. 7.

Since the noise resulting from measurement devices contaminates the control procedure/operation, robustness examination of the RKcontroller in terms of measurement noise is requisite. The robustness and also tracking performance of the RKcontroller against measurement noise have been perused. Therefore, y1(t) and y2(t) are exposed to measurement noise with σ_CB(t) = σ_T(t) = 0.0003 standard deviations for C_B and for T. The tracking performance, control and correction terms are depicted in Fig. 8.

In spite of noisy conditions, the system outputs can be compelled to the desired set points. When the control task sharings in Fig. 9 are evaluated, it is observed that RKcontroller takes over the control task as soon as possible. As given in Fig. 9, it is clear that ∆υ[n] is only effective when uRK-C[n] is not optimal initially; however, ∆υ[n] instantly converges to zero and RKcontroller has taken on virtually all control task.

Another significant case to assess the robustness of the RKcontroller is evaluation of control performance when one of the system parameter is unknown or not determined accurately (Iplikci 2013; Uçak 2019, 2020). The control performance of the RKcontroller depends concretely on the preciseness of the RKmodel since the parameters of RKcontroller are adjusted by considering future trajectories of the system via iterative predictive system model. Therefore, to
obtain unknown $RK_{\text{model}}$ parameters, $RK_{\text{estimator}}$ subblock in $RK_{\text{model}}$ is employed to approximate the system model parameters as given in Iplikci (2013), Uçak (2019, 2020). For this purpose, it is essential to appraise not only the tracking performance of the $RK_{\text{controller}}$ but also system model parameter approximation accuracy of $RK_{\text{estimator}}$ under parametric uncertainty case. As a scenario, a nonlinear uncertainty is introduced to system parameter as $C_{A_0}(t) = 5 + 0.5 \sin \left( \frac{2}{5} \pi t \right)$ and input signals are set to as 0.95 and 407.25. The behaviour of the closed-loop system and approximation ability of $RK_{\text{estimator}}$ subblock are depicted in Fig. 10. The accurate value of uncertain system parameter is precisely estimated in a timely manner and then $RK_{\text{estimator}}$ retains approximation during control (Iplikci 2013). Figure 11 indicates the impact percentage of the control signal produced by $RK_{\text{controller}}$.

The $u_{RK-C}[n]$ term becomes dominant in a very short time in comparison with $\delta u[n]$ term as given in Fig. 11. Since the computational load of an adaptive controller is vital in applications, real-time applicability of $RK_{\text{controller}}$ has been appraised. For this purpose, computational time for every sampling period is computed and registered. The stored execution times for each situation are illustrated in Fig. 12.

Then, using the maximum values of computation times, Table 2 is constituted. As can be clearly seen from Fig. 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 $RK_{\text{controller}}$ are less than 7 ms and very smaller than sampling time of VdV system for each case, which enables to deploy $RK_{\text{controller}}$ in real time. As can be seen from Fig. 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 RKcontroller 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 examined to reveal and examine which part takes on the control task. For a fair evaluation, the same conditions applied to RKcontroller have been implemented in Runge–Kutta model-based PID controller. The control performance of Runge Kutta model-based PID controller is depicted in Figs. 13, 15, 17 and 19. The task sharings are illustrated in Figs. 14, 16, 18 and 20. The computational load of the mentioned controller is depicted in Fig. 21. The controlled system outputs for staircase reference inputs are given in Fig. 13.

As illustrated in Fig. 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 Figs. 15, 17 and 19. The percentage (%) of tasks sharing for mentioned cases is depicted in Figs. 16, 18 and 20.
The total computational load of Runge–Kutta model-based PID controller is illustrated in Fig. 21 for all cases. The computational load of each operation in Runge–Kutta model-based PID controller algorithm is detailed in Table 3. When computational loads of RKcontroller and Runge–Kutta model-based PID are compared, it is clear that RKcontroller has better performance except for uncertainty case.

The bar graph in Fig. 22 is constructed to compare the tracking performances of RKcontroller and Runge–Kutta model-based PID controller using the following performance index for each system output.

\[
P_1 = \inf \sum_{n=1}^{\text{inf}} \left[ r_1[n] - y_1[n] \right]^2 \\
P_2 = \inf \sum_{n=1}^{\text{inf}} \left[ r_2[n] - y_2[n] \right]^2
\]

(29)

The comparison graph in Fig. 22 is detailed in Table 4. As given in Fig. 22 and Table 4, RKcontroller 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 RKcontroller 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 as ODE form. The adjustment mechanism comprises of RKcontroller and predictive RKmodel to forecast the future system behaviours. The adjustable parameters are tuned via LM learning law.
A novel adaptive Runge–Kutta controller for nonlinear dynamical systems

0
5
10
15
20
25
30
35
40
45
50
4.5
5
5.5
6
6.5
0
5
10
15
20
25
30
35
40
45
50
0
405
406
407
408
50
100
150
200
250
300
0
3
6
9
12
15
18
21
24
27
30
0.4
0.6
0.8
1
Time [hour]

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

Table 3  Execution times (ms) for Runge–Kutta model-based PID

| Operations / Cases                                    | Noiseless | Noiseless (Sin) | Noisy  | Uncertain |
|-------------------------------------------------------|-----------|-----------------|--------|-----------|
| EKF State Estimation (RK\textsubscript{EKF})           | 1.2525    | 1.6257          | 1.8996 | 1.0885    |
| \( P_H \)-step prediction (RK\textsubscript{model}) and computation of \( J \) (Jacobian Matrix) | 3.1561    | 2.6708          | 4.1038 | 2.4304    |
| RK\textsubscript{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    |

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) | Noisy | Uncertain |
| RK\textsubscript{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 |

The performance of adaptive RK\textsubscript{controller} is evaluated on VdV system. The robustness of RK\textsubscript{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 RKcontroller 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 modelling architectures based on RK integration approach for nonlinear dynamical systems.

Declarations

Conflict of interest The author declares that they have no conflict of interest.

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. https://doi.org/10.1016/0005-1098(77)90067-X

Beyhan S (2013) Runge–Kutta model-based nonlinear observer for synchronization and control of chaotic systems. ISA Trans 52(4):491–509. https://doi.org/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. Intell Autom Soft Comput 6(4):271–280. https://doi.org/10.1080/10798587.2000.10642795

Bittinger ML, Ellenbogen DJ, Surgent SA (2012) Calculus and its applications. Addison Wesley/Pearson, London

Bouallègue S, Haggège J, Ayadi M, Benrejeb M (2012) PID-type fuzzy logic controller tuning based on particle swarm optimization. Eng Appl Artif Intell 25(3):484–493. https://doi.org/10.1016/j.engappai.2011.09.018

Cetin M, Iplikci S (2015) A novel auto-tuning PID control mechanism for nonlinear systems. ISA Trans 58:292–308. https://doi.org/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. University of the Netherlands

Denai MA, Palis F, Zeghib 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. https://doi.org/10.1016/S0957-4158(98)00047-6

Efe MO, Kaynak O (2000) A comparative study of soft-computing methodologies in identification of robotic manipulators. Robot Auton Syst 30(3):221–230. https://doi.org/10.1016/S0921-8890(99)00087-1

Efe MO (2011) Neural network based control. In: The industrial electronics handbook. CRC Press, London, pp 3–1–3–26

Engell S, Klett 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, De Jesús O (2002) An introduction to the use of neural networks in control systems. Int J Robust Nonlinear Control 12(11):959–985. https://doi.org/10.1002/rnc.727

Kravaris C, Niemiec MP, Kravaris C (eds) Nonlinear model-based process control. CRC Press, London

Kulikov YJ, Kulikova MV (2014) Accurate state estimation in the Van der Pol system. 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. https://doi.org/10.1016/S0005-1098(03)00103-1

Pham DT, Karaboga D (1999) Self-tuning fuzzy controller design using genetic optimisation and neural network modelling. Artif Intell Eng 13(2):119–130. https://doi.org/10.1016/S0955-1859(98)00017-X

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

Roberts CE (2010) Ordinary differential equations: applications, models and computing. CRC Press, London

Sačens M, Soquet A (1991) Neural controller based on back-propagation algorithm. IEEE Proc F Radar Signal Process 138(1):55–62. https://doi.org/10.1049/ip-f-2.1991.0009

Sharkawy AB (2010) Genetic fuzzy self-tuning PID controllers for antilock braking systems. Eng Appl Artif Intell 23(7):1041–1052. https://doi.org/10.1016/j.engappai.2010.06.011

Tanomaru J, Omatu S (1992) Process control by on-line trained neural controllers. IEEE Trans Ind Electron 39(6):511–521. https://doi.org/10.1109/41.170970

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

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

Uçak K (2010) A Runge–Kutta neural network-based control method for nonlinear MIMO systems. Soft Comput 23(17):7769–7803

Uçak K (2020) A novel model predictive Runge–Kutta neural network controller for nonlinear MIMO systems. Neural Process Lett 51:1789–1833

Vojtesek J, Dostál P (2010) Adaptive control of chemical reactor. In: International conference cybernetics and informatics. VYSNA BOCA, Slovak Republic

Wang YJ, Lin CT (1998) Runge–Kutta neural network for identification of dynamical systems in high accuracy. IEEE Trans Neural Netw 9(2):294–307. https://doi.org/10.1109/21.66124

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

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.