Concurrent Learning-Based Adaptive Control of an Uncertain Robot Manipulator With Guaranteed Safety and Performance

Cong Li, Fangzhou Liu, *Member, IEEE*, Yongchao Wang, and Martin Buss, *Fellow, IEEE*

***Abstract*—This article investigates the tracking problem of an uncertain *n*-link robot manipulator with guaranteed safety and performance. To tackle parametric uncertainties, the torque filtering-augmented concurrent learning (CL) method is introduced for online identification of the unknown system without requirements of joints acceleration. By using CL, the parameter convergence is guaranteed by exploiting the current and historical data simultaneously. This technique enjoys practicability compared with common methods that need to incorporate external noises to satisfy the persistence of excitation condition for the parameter convergence. Based on the estimated model, we design a barrier Lyapunov function (BLF)-based adaptive control law by the backstepping technique and Lyapunov analysis. By ensuring the boundness of the BLF, the system output and the tracking error are proved to lie in the safety set and performance set, respectively. Numerical simulation results and experiment tests validate the proposed strategy.**

***Index Terms*—Adaptive control, backstepping method, barrier Lyapunov function (BLF), concurrent learning (CL), output constraints, parametric uncertainties, torque filtering.**

I. Introduction

SAFETY issues due to physical human–robot interactions and parametric uncertainties caused by environmental influences are inevitable in robotic systems. For given tasks in real applications, uncertain robot manipulators need to work in consideration of requirements for both safety and performance. Control strategies that partially focus on performance without safety guarantee [1] often result in a lack of practicability and vice verse [2]. This motivates us to develop an effective control strategy for a robot manipulator such that safety, performance, and uncertainty could be considered together.

A. Prior and Related Works

In terms of the safety guarantee problem, existing works include reference governors [3], invariant sets [4], quadratic programming (QP)-based methods [5], [6], etc. However, although the importance of safety has been stressed, these works do not fully consider requirements of task performance during the controller design process. Model predictive control (MPC) [7] is an effective method that considers objectives of both safety and performance, whereas it suffers from high computational complexity, which limits its application in real-time scenarios [8]. Given the task performance, methods based on Bellman's principle of optimality [9] or Pontryagin's maximum principle [10] are common. Nonetheless, based on these existing optimization frameworks [11], safety issues are difficult to be tackled. As an alternative, learning-based methods proposed in [2] and [12] can tackle safe optimization problems but with no stability guarantee. Clearly, it is nontrivial work to achieve the objectives of safety and performance simultaneously. To this end, we integrate the requirements of both safety and performance together and formulate a comprehensive constrained adaptive control problem.

To deal with constraints, the barrier Lyapunov function (BLF), which enjoys simplicity and low computational complexity for practical applications [13], emerges as an efficient tool based on the backstepping technique [14], [15]. BLF can be categorized as symmetric BLF (SBLF) and asymmetric BLF (ABLF). Symmetric and asymmetric constraints can be effectively addressed by SBLF and ABLF-based controllers, respectively. SBLFs are originally proposed in [16] and extended in [13] to deal with symmetric output constraints. In addition, multiple SBLFs in a weighted form are used to construct ABLFs to deal with asymmetric output constraints [13]. For further extensions (see [17]–[20]).

Although BLF-based controllers have been used for various aims, their effectiveness relies on an explicit model that is not always available in real applications [21]. For example, even for a well-structured robot manipulator, parametric uncertainties may still occur [22]. To tackle model uncertainties, function approximation-based methods are widely utilized. The present function approximation-based methods can be categorized in terms of different approximation schemes, such as polynomials [23], trigonometric series [24], orthogonal functions [25], splines [26], fuzzy systems [27], and neural networks (NNs) [28]. Among these approximation schemes, NNs play a vital role in learning-based control methods [29]–[31]. Normally, the NN approximation scheme is first adopted to approximate an unknown dynamics and then, a BLF-based control law is designed based on this approximated model to deal with corresponding constraints. See [20], [32], [33] for its applications. Note that the adopted

Manuscript received August 19, 2020; revised January 5, 2021; accepted February 22, 2021. Date of publication March 19, 2021; date of current version April 15, 2022. This article was recommended by Associate Editor W. Sun. *(Corresponding author: Fangzhou Liu.)*

The authors are with the Chair of Automatic Control Engineering, Technical University of Munich, 80333 Munich, Germany (e-mail: cong.li@tum.de; fangzhou.liu@tum.de; ywang@tum.de; mb@tum.de).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TSMC.2021.3064971.

Digital Object Identifier 10.1109/TSMC.2021.3064971

NNs are in essence linearly parameterized two-layer NNs with a hidden layer and an output layer. The hidden layer performs a fixed nonlinear transformation of the input, and NN weights appear linearly between the hidden layer and the output layer. The radial basis function (RBF) NN approximation scheme reported in [20] is a local approximation scheme where each basis function only contributes to the output locally. In this regard, the approximation ability of the RBF NNs depends on the valid region of each chosen RBF. The valid region of the RBF is determined by their centers and widths. These hyperparameters should be determined *a priori* by considering the whole working space of the investigated system. However, the working space is often unknown, especially when there exist uncertainties. To give full play of the effectiveness of the RBF NN approximation scheme, the choice of suitable RBFs needs trials and fails.

Among all the aforementioned applications of BLF-based controllers via the NN approximation scheme, the fundamental problem about weight convergence of NNs is yet to be discussed. In general, weight (parameter) convergence is not required in adaptive control, and it has been illustrated in [20], [21], [34], and [35] that parameter estimation errors may not converge to 0 (indeed it may not converge at all) even though an acceptable tracking performance is achieved. However, it has been proved that robustness and transient performance of adaptive control methods can be improved with guaranteed parameter convergence [36]–[38]. For the aforementioned literature based on the NN approximation scheme [20], [32], [33], the problem of guaranteed parameter convergence to the actual value, nonetheless, is out of consideration. Conventionally, the persistence of excitation (PE) condition is used to check the parameter convergence. Parameter convergence to the actual value is guaranteed if the PE condition is satisfied [35], [39]. Among the existing works [21], [40], although the PE condition can be satisfied by incorporating external noises to control inputs, this method is restrictive to practical applications in terms of the following aspects: 1) as for the robot control, it may degrade control performance and cause nuisance, waste of energy, etc.; 2) since the satisfaction of the PE condition is a problem relying on future signals, it is often impractical to monitor online whether the PE condition is satisfied or not; and 3) there is no generic method to construct the appropriate form of external noises to satisfy the PE condition since it correlates with scenarios [41]. Thus, instead of using external noises to satisfy the PE condition, we adopt the concurrent learning (CL) method by using the current and historical data simultaneously to provide the required excitation for the parameter convergence. The CL technique is first introduced in [42] to tackle parametric uncertainties of the model reference adaptive control (MRAC). Then, CL is extended to identify parameters of an Euler–Lagrange (E–L) system [43]. However, information of joints acceleration that is sensitive to noises is required in [43] for parameter identification. To control a robot manipulator efficiently for real applications, we need to develop a kind of CL technique that avoids using the information of joints acceleration directly. Closely related to CL is the experience replay technique, which replays experience data to accelerate the online learning process, adopted in the reinforcement learning field [44]. However, its usage accompanies with extensive parameter tuning efforts and the subsequent way of data usage suffers from a data deficiency problem.

B. Contributions

The main contribution of this article is to develop a stable adaptive control strategy that can deal with the following problems simultaneously: 1) safety issues in terms of output constraints; 2) guaranteed performance concerning tracking errors; and 3) parametric uncertainties of a partially unknown robot manipulator. By integrating requirements of safety and performance together, a simple BLF is first designed to account for the safety and performance-related constraints simultaneously. Then, a novel double regressor matrix technique is developed to enable the combination of the BLF-based control strategy and the torque filtering augmented CL (TF-CL) aided online system identification feasible. The torque filtering technique is integrated into CL to get rid of joints acceleration for the parameter estimation, and the guaranteed parameter convergence to the actual value is achieved for the function approximation-based BLF control strategy. Additionally, an efficient history stack management algorithm is proposed to collect sufficient rich data to enable the estimated parameters to converge to the desired values fast.

C. Organization of This Article

The remainder of this article is organized as follows. Section II introduces preliminaries and the problem formulation of the stable adaptive control of a partially unknown robot manipulator with guaranteed safety and performance. The torque filtering technique is illustrated in Section III, which serves to Section IV to construct a TF-CL aided online parameter estimation update law based on the double regressor matrix technique and the history stack management algorithm. Section V elucidates the recursive design process of the BLF-based controller by the backstepping technique and presents stability proofs, compact sets of both tracking errors and system outputs. Numerical simulation and experimental results shown in Section VI illustrate the effectiveness of the proposed control strategy. The conclusion is provided in Section VII.

Notations: Throughout this article, $\mathbb{R}$ denotes the set of real numbers, $\mathbb{R}^n$ is the Euclidean space of the n-dimensional real vector, $\mathbb{R}^{n\times m}$ is the Euclidean space of $n \times m$ real matrices, $\lambda_{\max}(M)$ and $\lambda_{\min}(M)$ are the maximum and minimum eigenvalues of a symmetric real matrix M, respectively, and $\|\cdot\|$ represents the Euclidean norm for vectors and induced norm for matrices. Denoting x_i as the ith entry of a vector $x \in \mathbb{R}^n$, $\mathrm{diag}(x)$ is the $n \times n$ diagonal matrix with the ith diagonal entry equals x_i. For any two real vectors $a, b \in \mathbb{R}^n$, $a \leq (\ll) b$ is the component-wise comparison, i.e., $a_i \leq (<) b_i$ for $\forall i \in \{1, \ldots, n\}$.

II. Preliminaries and Problem Formulation

The dynamics of an n-link robot manipulator is represented by the following E–L equation:

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) + F\dot{q} = \tau \tag{1}$$

where $M(q) : \mathbb{R}^n \to \mathbb{R}^{n\times n}$ is the symmetric positive-definite inertia matrix, $C(q,\dot{q}) : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^{n\times n}$ is the matrix of centrifugal and Coriolis terms, $G(q) : \mathbb{R}^n \to \mathbb{R}^n$ represents the gravitational terms, $F \in \mathbb{R}^{n\times n}$ denotes values of viscous friction, $q \in \mathbb{R}^n$, $\dot{q} \in \mathbb{R}^n$, and $\ddot{q} \in \mathbb{R}^n$ are the vectors of joint angles, velocities, and accelerations, respectively, and $\tau \in \mathbb{R}^n$ represents the vector of input torques applied at each joint.

Property 1 [22]: The left-hand side of system equation (1) can be written as the following linear in parameter (LIP) form:

$$Y(q,\dot{q},\ddot{q})\theta^* = \tau \tag{2}$$

where $Y(q,\dot{q},\ddot{q}) : \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^{n\times m}$ is the regressor matrix and $\theta^* \in \mathbb{R}^m$ is the desired coefficient vector of the E–L system dynamics.

Remark 1: Property 1 exploits the known model properties at hands to construct the regression matrix $Y(q,\dot{q},\ddot{q})$. Considering the E–L equation (1), model uncertainties include varying masses or lengths of joints, varying friction parameters, and unknown payloads. The aforementioned model uncertainties can all be incorporated into the coefficient vector θ^* of (2). Note that for the model-free NN approximation scheme [28], the known physical structure of the investigated system is abandoned, which usually inevitably suffers from the well known poor sample complexity problem.

The E–L equation (1) can be written in the following state-space form by setting $x_1 = q$ and $x_2 = \dot{q}$:

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= M^{-1}(x_1)(\tau - C(x_1,x_2)x_2 - G(x_1) - Fx_2) \\ y &= x_1 \end{aligned} \tag{3}$$

where $y \in \mathbb{R}^n$ is the system output that denotes joint angles of the n-link robot manipulator, and assuming that it lies in the following set:

$$\mathcal{C} = \{y \in \mathbb{R}^n : k_e \ll y \ll k_f\}. \tag{4}$$

Here, we consider a trajectory tracking problem, where the robot manipulator is driven to track the desired trajectory y_d precisely. Throughout this article, we confine ourselves that the desired trajectory y_d satisfies the following assumption.

Assumption 1: The desired trajectory $y_d \in \mathbb{R}^n$ satisfies $-\underline{y}_d \le y_d \le \bar{y}_d$, where $\underline{y}_d$ and $\bar{y}_d$ are positive constant vectors. Based on the system output y of (3) and the desired trajectory y_d, we define the tracking error $e_1 \in \mathbb{R}^n$ as

$$e_1 = y - y_d. \tag{5}$$

For the safety issues considered during the trajectory tracking process, following the definition of barrier functions illustrated in [45], here, the safety set regarding the system output y is defined as:

$$\mathcal{S} = \{y \in \mathbb{R}^n : h(y) \le 0\} \tag{6}$$

where $h(y) : \mathbb{R}^n \to \mathbb{R}$ is a continuous function. The explicit form of $h(y)$ is determined by considering various safety issues during the tracking process. As for the investigated n-link robot manipulator, considering the human–robot interactions or limited spaces, its safety set is usually defined as an allowable operation region [46], [47] that follows:

$$\bar{\mathcal{S}} = \{y \in \mathbb{R}^n : k_c \ll y \ll k_d\} \tag{7}$$

where $k_c = [k_{c_1},\ldots,k_{c_n}]^\top \in \mathbb{R}^n$ and $k_d = [k_{d_1},\ldots,k_{d_n}]^\top \in \mathbb{R}^n$ are known constant vectors determined by controller designers. The safety set $\bar{\mathcal{S}}$ in (7) is a representative and explicit form of $\mathcal{S}$ in (6). Note that $k_c \ll -\underline{y}_d$ and $\bar{y}_d \ll k_d$ hold, i.e., y_d lies in the safety set $\bar{\mathcal{S}}$.

For the performance issues, we demand that the tracking error e_1 (5) lies in the following performance set:

$$\mathcal{P} = \{e_1 \in \mathbb{R}^n : -k_a \ll e_1 \ll k_b\} \tag{8}$$

where $k_a = [k_{a_1},\ldots,k_{a_n}]^\top \in \mathbb{R}^n$ and $k_b = [k_{b_1},\ldots,k_{b_n}]^\top \in \mathbb{R}^n$ are the predefined constant vectors. According to (5), the resulting working space based on the desired tracking error bound (8) would be

$$\bar{\mathcal{P}} = \left\{y \in \mathbb{R}^n : -k_a - \underline{y}_d \ll y \ll k_b + \bar{y}_d\right\}. \tag{9}$$

To counter the constraints concerning safety in (7) and performance in (8), BLF emerges as an efficient tool. To deal with both symmetric and asymmetric constraints, a simple indicator function-based BLF is proposed as

$$V(z) = p(z)\frac{z^2}{k_u^2 - z^2} + (1 - p(z))\frac{z^2}{k_l^2 - z^2} \tag{10}$$

where $z \in \mathbb{R}$, $k_l \in \mathbb{R}$ and $k_u \in \mathbb{R}$ are constraint bounds. When $z \to k_l$ or $z \to k_u$, $V(z) \to \infty$, and $p(z)$ is an indicator function that follows:

$$p(z) = \begin{cases} 1, & z > 0 \\ 0, & z \le 0. \end{cases} \tag{11}$$

According to [13, Definition 2], the proposed BLF in (10) is an effective BLF. Note that when the value of $p(z)$ in (10) is always set as 0 or 1 regardless of the value of z, (10) is an SBLF, and it can only deal with the symmetric constraint of which the constraint boundaries are $\pm k_l$ or $\pm k_u$. When the value of $p(z)$ alternates between 0 and 1 as defined in (11), (10) is an ABLF that can counter an asymmetric constraint. For the ABLF case, we set $k_l < 0$ and $k_u > 0$, where k_l and k_u are the lower bound and upper bound of the asymmetric constraint set, respectively.

Remark 2: From the perspective of the guaranteed performance represented by (8), prescribed performance control (PPC) [48] is closely related to our work, which exploits the prescribed performance function (PPF)-based system transformation technique to guarantee that the tracking error converges to an explicit residual set, the convergence rate is no less than a predefined value, and a maximum overshoot is less than a prespecified constant. However, although multiple performance criteria could be provided by PPC, we found in practice that its efficient application requires extensive parameter tuning efforts because the adopted PPF is sensitive and easier lead to singularity. Moreover, the system transformation process results in additional complexity. Thus, a simple BLF is chosen here to achieve guaranteed performance and safety. Although no explicit values of the final residual set and convergence rate are provided by our designed

BLF-based control strategy in Section V, both simulation and experimental results in Section VI have shown satisfying performance comparable to the PPC-based work [48].

For the investigated tracking problem, the priority of safety is over performance. To improve the tracking performance while always guaranteeing safety, $\mathcal{C} \subseteq \bar{\mathcal{S}}$ and $\mathcal{C} \subseteq \bar{\mathcal{P}}$ should be satisfied together. For the purpose of achieving these considerations simultaneously, we could first choose the values of $-k_a$ and k_b to meet $\bar{\mathcal{P}} \subseteq \bar{\mathcal{S}}$, and then design a tracking controller to enforce $\mathcal{C} \subseteq \bar{\mathcal{P}}$. For example, consider a scenario that a robot manipulator works close to humans, where the preplanned y_d for given tasks ensures collision free with humans. To guarantee the collision avoidance while accomplishing the predefined tasks, we need to restrict the operation range of the robot manipulator such that $\mathcal{C} \subseteq \bar{\mathcal{S}}$ and enable the robot manipulator to track y_d precisely to satisfy $\mathcal{C} \subseteq \bar{\mathcal{P}}$, respectively. The aforementioned safety and performance requirements can be integrated together by choosing the explicit values (i.e., $-k_a$ and k_b) of the guaranteed tracking performance such that $\bar{\mathcal{P}} \subseteq \bar{\mathcal{S}}$ holds. Then, a BLF-based controller that drives the robot manipulator to track y_d with the guaranteed tracking performance also enforces the executed trajectory to lie in the restricted operation range at the same time. Although it seems to be a conservative approach, comparing to works that can only consider partial objectives of performance [1] or safety [49], the resulting BLF-based controller can drive the robot manipulator to track the desired trajectory while satisfying the requirements of both safety and performance together.

Based on the aforementioned settings, the tracking problem with guaranteed safety and performance is formulated as follows.

Problem 1: Given the uncertain robot manipulator (1), the desired trajectory y_d is within the prior known safety set $\bar{\mathcal{S}}$ (7). Choose appropriate bounds for the performance set $\mathcal{P}$ (8) and design a stable adaptive control strategy based on the proposed BLF (10) to drive the uncertain robot manipulator to track the desired trajectory y_d while satisfying requirements of safety characterized by $\bar{\mathcal{S}}$ and performance denoted as $\mathcal{P}$ together.

III. Torque Filtering Technique

For the LIP form of the E–L equation given in (2), measurements of the acceleration $\ddot{q}$ are required to construct the regressor matrix $Y(q, \dot{q}, \ddot{q})$. Since the information of joint acceleration is sensitive to measurement noises, it is not applicable to use it directly to design a controller. To eliminate the need for this information, the torque filtering technique is adopted here to reformulate the original LIP form (2) to get a new equivalent LIP form without requirements for the information of joints acceleration. Comparing to the common Kalman filter that highly depends on prior knowledge (e.g., noises to be filtered) and requires extensive parameter tuning efforts [50], the adopted torque filtering technique is a simple and easily implemented method for real applications.

To introduce the torque filtering technique, two auxiliary vectors $h(q, \dot{q}) : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^n$ and $g(q, \dot{q}) : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^n$ are first defined as

$$\begin{aligned} h(q, \dot{q}) &= M(q)\dot{q} = Y_1(q, \dot{q})\theta^* \\ g(q, \dot{q}) &= -\dot{M}(q)\dot{q} + C(q, \dot{q})\dot{q} + G(q) + F\dot{q} = Y_2(q, \dot{q})\theta^* \end{aligned} \tag{12}$$

where $Y_1(q, \dot{q}) : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^{n\times m}$ and $Y_2(q, \dot{q}) : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^{n\times m}$ are two new regressor matrices without incorporating the information of $\ddot{q}$.

Based on the auxiliary vectors in (12), system equation (1) can be rewritten as

$$\dot{h}(q, \dot{q}) + g(q, \dot{q}) = (\dot{Y}_1(q, \dot{q}) + Y_2(q, \dot{q}))\theta^* = \tau \tag{13}$$

where $\dot{h}(q, \dot{q}) = \dot{M}(q)\dot{q} + M(q)\ddot{q} = \dot{Y}_1(q, \dot{q})\theta^*$.

The advantage of writing the robot manipulator model in the form (13) is that this new equivalent form of (1) has been separated in a way that allows $\ddot{q}$ to be filtered out. To filter out $\ddot{q}$ existing in $\dot{Y}_1(q, \dot{q})$, a linear stable filter is introduced as

$$f(s) = \frac{1}{ks + 1} \tag{14}$$

where s is the Laplace operator and $k \in \mathbb{R}$ is a time constant. By filtering (13) based on (14), we can get the filtered version of (13) as

$$\dot{h}_f(q, \dot{q}) + g_f(q, \dot{q}) = \tau_f \tag{15}$$

where $\dot{h}_f(q, \dot{q}) : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^n$ and $g_f(q, \dot{q}) : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^n$ are the filtered versions of $\dot{h}(q, \dot{q})$ and $g(q, \dot{q})$, respectively. $\tau_f \in \mathbb{R}^n$ is the filtered version of τ.

Based on (13), the corresponding LIP form of the filtered system (15) is given as

$$(\dot{Y}_{1_f}(q, \dot{q}) + Y_{2_f}(q, \dot{q}))\theta^* = \tau_f \tag{16}$$

where $Y_{1_f}(q, \dot{q}) : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^{n\times m}$ and $Y_{2_f}(q, \dot{q}) : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^{n\times m}$ are the filtered versions of the regressor matrices $Y_1(q, \dot{q})$ and $Y_2(q, \dot{q})$, respectively.

For the filter given in (14), the filtered variables and their original forms satisfy the following equations:

$$\begin{aligned} k\dot{Y}_{1_f}(q, \dot{q}) + Y_{1_f}(q, \dot{q}) &= Y_1(q, \dot{q}), \quad Y_{1_f}(q, \dot{q})|_{t=0} = 0 \\ k\dot{Y}_{2_f}(q, \dot{q}) + Y_{2_f}(q, \dot{q}) &= Y_2(q, \dot{q}), \quad Y_{2_f}(q, \dot{q})|_{t=0} = 0 \\ k\dot{\tau}_f + \tau_f &= \tau, \quad \tau_f|_{t=0} = 0. \end{aligned} \tag{17}$$

Substituting the first equation of (17) into (16), finally, we get the filtered LIP form of the E–L equation (1) as

$$Y_f(q, \dot{q})\theta^* = \tau_f \tag{18}$$

where $Y_f(q, \dot{q}) = (Y_1(q, \dot{q}) - Y_{1_f}(q, \dot{q}))/k + Y_{2_f}(q, \dot{q}) : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^{n\times m}$ is the new filtered regressor matrix without requirements for information of joints acceleration.

Now, the new filtered regressor matrix $Y_f(q, \dot{q})$ and the resulting filtered LIP form (18) can be adopted to identify the unknown coefficient vector θ^* without requirements for information of joint acceleration, which is clarified in detail in the next section.

IV. Concurrent Learning Aided System Identification

Since the ideal coefficient vector θ^* in (18) is unknown and only the estimated parameter vector $\hat{\theta}$ is available, the identification problem to be addressed here is to obtain $\hat{\theta}$ online based on the system input u, the output state y, and the filtered regression matrix Y_f. Online identification of $\hat{\theta}$ is an adaptive parameter estimation problem, where the PE condition needs to be satisfied before estimated parameters converge to the desired values. Unlike common methods that introduce external noises to satisfy the PE condition, based on the LIP form in (18), the TF-CL method is adopted here to guarantee the parameter convergence by utilizing both the current and historical data.

A. Parameter Estimation Update Law

Denoting the parameter estimation error as $\tilde{\theta} = \hat{\theta} - \theta^*$, the corresponding model approximation error follows:

$$e_f = Y_f \tilde{\theta}. \tag{19}$$

Define the quadratic cost of the approximation error as $V_{e_f} = 1/2 e_f^\top e_f$, by following the common gradient descent method to minimize V_{e_f}, the adaptive parameter estimation update law is derived as:

$$\dot{\hat{\theta}} = -\Gamma Y_f^\top e_f \tag{20}$$

where $\Gamma \in \mathbb{R}^{m\times m}$ is a constant positive-definite matrix. It is well known that the estimated $\hat{\theta}$ converges to the desired θ^*, iff the regression matrix Y_f satisfies the PE condition [51]

$$\int_t^{t+T} Y_f^\top(\tau) Y_f(\tau) d\tau \geq \gamma I \tag{21}$$

where γ, $T \in \mathbb{R}$ are appropriate positive constants. The PE condition in (21) can be interpreted as requirements for a degree of data richness: when the regressor matrix Y_f varies sufficiently enough over the time interval T so that the entire γ dimension parameter space is spanned, the estimated parameters are guaranteed to converge to the desired values.

Common methods usually adopt the parameter estimation update law in (20) and introduce external noises, e.g., signals in sin or cos form, to satisfy the PE condition shown as (21). However, the PE condition in (21) is hard to check online whether it is satisfied or not. An online verification condition is desirable to tell controller designers that under this condition, the estimated parameters are guaranteed to converge to the desired values. Observing that the PE condition is, in essence, a condition about data richness, and only current data contribute to the common parameter estimation update law (20); therefore, to get rich enough data, historical data will also be exploited here to construct the parameter estimation update law.

In this section, a parameter estimation update law is proposed by using the current and historical data simultaneously. The need for adding external noises to satisfy the PE condition (21) can be avoided with the benefit of the recorded historical data. Based on the TF-CL method, the parameter estimation update law for the unknown coefficient vector θ^* is designed as

$$\dot{\hat{\theta}} = -\Gamma k_t Y_f^\top e_f - \sum_{j=1}^{P} \Gamma k_h Y_{f_j}^\top e_{f_j} \tag{22}$$

where $k_t, k_h \in \mathbb{R}$ are the positive constant gains to tradeoff the relative importance between the current and historical data to the parameter estimation update law. P is a positive natural number denoting the volume of the history stacks H and E. The history stacks H and E are collections of historical data, where the filtered regressor matrix Y_{f_j} and the filtered approximation error e_{f_j} denote the jth collected data of the history stack H and E, respectively.

The parameter estimation update law in (22) contains two parts. The first part $-\Gamma k_t Y_f^\top e_f$ relates to the current data, which is a common gradient descent update law to minimize the quadratic model approximation error V_{ef}, as like (20). However, an update law only with the first part cannot guarantee parameter convergence. Thus, the second part $-\sum_{j=1}^{P} \Gamma k_h Y_{f_j}^\top e_{f_j}$, which is constructed by historical data, is introduced to provide the sufficient excitation required for the parameter convergence. To analyze the parameter convergence problem based on the parameter estimation update law (22), a rank condition is first clarified in Assumption 2.

Assumption 2: Given a history stack $H = [Y_{f_1}^\top, \ldots, Y_{f_P}^\top] \in \mathbb{R}^{m\times(n\times P)}$, where $Y_{f_j} \in \mathbb{R}^{n\times m}$ is the jth collected data of H, there holds $\text{rank}(HH^\top) = m$.

Note that comparing to the traditional PE condition in (21), the rank condition of a history stack H in Assumption 2 provides an index about the richness of the historical data that can be checked online. If the rank condition is satisfied, it guarantees that the estimated parameters will converge to the desired values and vice versa. Proofs for parameter convergence are given as follows.

Theorem 1: Given Assumption 2 and the parameter estimation update law in (22), the parameter estimation error $\tilde{\theta}$ converges to 0 asymptotically.

Proof: See Appendix A. ∎

B. History Stack Management Algorithm

The parameter estimation update law and the corresponding convergence proof have been provided in Theorem 1. The premise of Theorem 1 is the satisfaction of the rank condition in Assumption 2, i.e., a history stack H containing sufficiently different data is needed. Besides, according to (56) and $Q = HH^\top$, the convergence rate of the estimated parameters is related to the minimum eigenvalues of the history stack H, i.e., $\lambda_{\min}(HH^\top)$. With the above analysis, we know that the convergence of the estimated parameters to the desired values with a fast speed equals: 1) the satisfaction of the rank condition in Assumption 2 and 2) the enlargement of the minimum eigenvalues $\lambda_{\min}(HH^\top)$. Thus, to achieve parameter convergence with a fast speed, in our algorithm, the history stacks H and E are updated with new data points based on two criteria: one is the data threshold ε that acts as a criterion for data difference, and guides the algorithm to collect different enough data to

Algorithm 1 History Stack Management Algorithm

Input: Iteration number: $i \geq 1$; Data threshold: ε; Volume: P; Auxiliary variables: T_h, T_e; Index: $I = P$; Empty set: S; State dimension: n.
Output: History stacks H, E.

if $i \leq P$ **then**
  if $\|Y_f - H(:,\ ni-n+1:ni)\|/\|Y_f\| \geq \varepsilon$ **then**
    $H(:,\ ni-n+1:ni) = Y_f^\top$ in (18)
    $E(:,\ n) = e_f$ in (19)
    $i = i+1$
  end if
else
  if $\|Y_f - H(:, nI-n+1{:}nI)\|/\|Y_f\| \geq \varepsilon$ **then**
    $T_h = H$; $T_e = E$; $V = \sigma_{\min}(HH^\top)$
    for $l = 1{:}P$ **do**
      $H(:, nl-n+1{:}nl) = Y_f^\top$ in (18)
      $S(l) = \sigma_{\min}(HH^\top)$; $H = T_h$
    end for
    $[V_{\max}, I] = \max(S)$
    if $V_{\max} \geq V$ **then**
      $H(:, nI-n+1{:}nI) = Y_f^\top$ in (18)
      $E(:, I) = e_f$ in (19)
    else
      $H = T_h$; $E = T_e$
    end if
  end if
end if

satisfy the rank condition; the other is the minimum eigenvalues of the history stack H that relates to the convergence rate of the estimated parameters. Note that for computation simplicity, the minimum singular value $\sigma_{\min}(HH^\top)$ replaces with $\lambda_{\min}(HH^\top)$ to act as a criterion for data storage given that $\sigma_{\min}(HH^\top) = \sqrt{\lambda_{\min}(HH^\top)}$.

The details of Algorithm 1 are as follows. First, the hyperparameter data threshold ε ensures that only new data that is sufficiently different from the latest collected data will be incorporated into the history stacks H and E. Second, to improve the parameter convergence speed, when H reaches its volume limit P, only data points that lead to an increment of the minimum singular values of the history stack H will be collected. As for the method proposed in [42], the same data might be used multiple times in the history stack H (data richness deteriorates), and the monotonic increment of the minimum singular values cannot be guaranteed (convergence rate of the estimated parameter is discouraged). To ensure monotonic increment of the minimum singular values, in our algorithm, the newly coming data always compares with the latest data inserted into the history stack H. Note that the history stack volume P is a hyperparameter that requires careful tuning, which requires $P \geq m$ to satisfy the rank condition in Assumption 2, where m is the dimension of the desired coefficient vector θ^*. The pseudocode of the history stack management algorithm is shown as Algorithm 1.

Remark 3: In very special cases, it is still possible that the collected historical data from one single trajectory might not be rich enough to satisfy the rank condition in Assumption 2. To counter this potential data deficiency problem, in the initial learning period, a random noise Δ can be incorporated into the regressor matrix, i.e., $Y_f \leftarrow Y_f + \Delta$, within a short time to enable Algorithm 1 to collect the historical data that the real system does not experience. The random noise Δ is abandoned once the rank condition in Assumption 2 is satisfied.

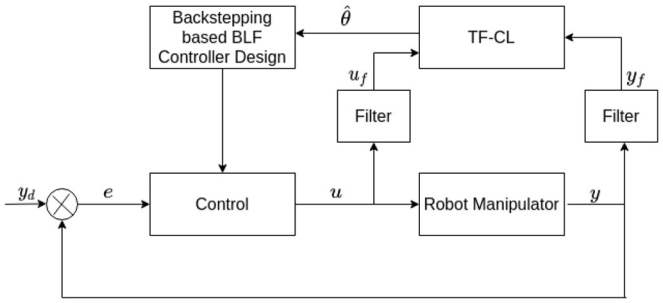

Fig. 1. Schematic of the proposed method that consists of the TF-CL aided system identification process followed by the BLF-based controller design.

V. Controller Design for Robot Manipulator

In this section, based on the identified system from Section III, a recursive controller design process is clarified to yield a stable adaptive control strategy by the backstepping technique and Lyapunov analysis, as shown in Fig. 1.

The resulting control strategy renders the time derivative of the BLF (10) to be always negative semidefinite, which guarantees that with a finite initial value of the BLF, then the BLF value will always be bounded during the tracking process. The boundness of the BLF implies that the safety set (7) and the performance set (8) will not be transgressed, i.e., requirements of safety and performance are both satisfied. Note that since an SBLF is a special case of an ABLF, the following derivation is based on an ABLF, which makes the controller design more challenging.

Recall the tracking error $e_1 = x_1 - y_d$ in (5) and define the error $e_2 = x_2 - \alpha$, where $\alpha \in \mathbb{R}^n$ is a stabilizing function to be designed.

Step 1: For the n-link robot manipulator, the following ABLF candidate is chosen to design a controller:

$$V_1 = \frac{1}{2}\sum_{i=1}^{n} p(e_{1_i})\frac{e_{1_i}^2}{k_{b_i}^2 - e_{1_i}^2} + \big(1 - p(e_{1_i})\big)\frac{e_{1_i}^2}{k_{a_i}^2 - e_{1_i}^2}. \tag{23}$$

Taking the time derivative of V_1 yields

$$\dot{V}_1 = \sum_{i=1}^{n} p(e_{1_i})\frac{k_{b_i}^2 e_{1_i}\dot{e}_{1_i}}{\left(k_{b_i}^2 - e_{1_i}^2\right)^2} + \big(1 - p(e_{1_i})\big)\frac{k_{a_i}^2 e_{1_i}\dot{e}_{1_i}}{\left(k_{a_i}^2 - e_{1_i}^2\right)^2}. \tag{24}$$

The time derivative of e_1 is

$$\dot{e}_1 = \dot{x}_1 - \dot{y}_d = x_2 - \dot{y}_d = e_2 + \alpha - \dot{y}_d. \tag{25}$$

Substituting (25) into (24) yields

$$\dot{V}_1 = \sum_{i=1}^{n} p(e_{1_i})\frac{k_{b_i}^2 e_{1_i}(e_{2_i} + \alpha_i - \dot{y}_{d_i})}{\left(k_{b_i}^2 - e_{1_i}^2\right)^2} + \big(1 - p(e_{1_i})\big)\frac{k_{a_i}^2 e_{1_i}(e_{2_i} + \alpha_i - \dot{y}_{d_i})}{\left(k_{a_i}^2 - e_{1_i}^2\right)^2} \tag{26}$$

where α_i and $\dot{y}_{d_i}$ are the ith dimension of α and $\dot{y}_d$, respectively. In order to make (26) be negative semidefinite, the stabilizing function α is designed as

$$\alpha = \dot{y}_d - p(e_1)\left(k_b^\top k_b - e_1^\top e_1\right)^2 k_1 e_1 - (1 - p(e_1))\left(k_a^\top k_a - e_1^\top e_1\right)^2 k_1 e_1 \tag{27}$$

where $k_1 \in \mathbb{R}^{n\times n}$ is a diagonal matrix of positive constants, and its ith diagonal entry is denoted as k_{1i}. Since asymmetric constraints are considered in this article, the last two terms of (27) are designed to characterize the upper and lower constraint boundaries, i.e., k_b and $-k_a$, respectively.

For simplicity, denoting $L = \sum_{i=1}^{n} p(e_{1_i})\ [(k_{b_i}^2\ e_{1_i}\ e_{2_i})/((k_{b_i}^2 - e_{1_i}^2)^2)] + (1-p(e_{1_i}))\ [(k_{a_i}^2\ e_{1_i}\ e_{2_i})/((k_{a_i}^2 - e_{1_i}^2)^2)]$. Then, (26) can be rewritten as the following form based on (27):

$$\begin{aligned}\dot{V}_1 &= -\sum_{i=1}^{n} p(e_{1_i}) k_{1_i} k_{b_i}^2 e_{1_i}^2 + (1 - p(e_{1_i})) k_{1_i} k_{a_i}^2 e_{1_i}^2 + L \\ &= -e_1^\top k_1 [p(e) k_b^\top k_b + (1 - p(e)) k_a^\top k_a] e_1 + L \\ &= -e_1^\top K_1 e_1 + L \end{aligned} \tag{28}$$

where $K_1 = k_1[p(e)k_b^\top k_b + (1 - p(e))k_a^\top k_a] \in \mathbb{R}^{n\times n}$ is a positive-definite matrix.

Step 2: We define

$$V_2 = \frac{1}{2} e_2^\top M(x_1) e_2 \tag{29}$$

and choose

$$V_{blf} = V_1 + V_2. \tag{30}$$

The time derivative of V_{blf} is

$$\dot{V}_{blf} = \dot{V}_1 + \dot{V}_2 = \dot{V}_1 + e_2^\top M(x_1)\dot{e}_2 + \frac{1}{2} e_2^\top \dot{M}(x_1) e_2. \tag{31}$$

Considering (3), the time derivative of e_2 follows:

$$\begin{aligned}\dot{e}_2 &= \dot{x}_2 - \dot{\alpha} \\ &= M^{-1}(x_1)(\tau - C(x_1, x_2)x_2 - G(x_1) - Fx_2) - \dot{\alpha}. \end{aligned} \tag{32}$$

Invoking (28), (31), and (32) yields

$$\dot{V}_{blf} = -e_1^\top K_1 e_1 + L + e_2^\top \Big[\tau - C(x_1, x_2)x_2 - G(x_1) - Fx_2 - M(x_1)\dot{\alpha} + \frac{1}{2}\dot{M}(x_1)e_2\Big] \tag{33}$$

where the stabilizing function α is defined in (27).

If an accurate model is available, a stabilizing control law can be directly designed as

$$\tau = M(x_1)\dot{\alpha} + C(x_1, x_2)x_2 + G(x_1) + Fx_2 - k_2 e_2 - \left(e_2^\top\right)^+ L - \frac{1}{2}\dot{M}(x_1)e_2 \tag{34}$$

where $k_2 \in \mathbb{R}^{n\times n}$ is a matrix of positive constants to be designed, $(e_2^\top)^+ L$ is a stabilizing term, and $+$ stands for the Moore–Penrose inverse.

Accurate model information is required in (34) to design a stabilizing control law, but it is not always available. To provide an approximation of the unknown model information existing in the right-hand side of (34), comparing the difference between (2) and (34), a double regressor matrices technique for the TF-CL is introduced here. Specifically speaking, like the regressor matrix $Y(q, \dot{q}, \ddot{q})$ that is proposed for approximation of the system (2), a new regressor matrix $X(x_1, x_2, \alpha, \dot{\alpha})$ is formulated to approximate the unknown model in (34). Based on the newly designed $X(x_1, x_2, \alpha, \dot{\alpha})$, the following approximation equation establishes:

$$X(x_1, x_2, \alpha, \dot{\alpha})\theta^* = M(x_1)\dot{\alpha} + C(x_1, x_2)x_2 + G(x_1) + Fx_2 - \frac{1}{2}\dot{M}(x_1)e_2 \tag{35}$$

where $X(x_1, x_2, \alpha, \dot{\alpha}) \in \mathbb{R}^{n\times m}$ is a regressor matrix constructed based on the information of x_1, x_2, α, and $\dot{\alpha}$. We defer a detailed discussion of the relationship between regressor matrices $X(x_1, x_2, \alpha, \dot{\alpha})$ and $Y(q, \dot{q}, \ddot{q})$ in Remark 4 and focus now on the design of the parameter estimation update law for the BLF-based controller with help of the new regressor matrix $X(x_1, x_2, \alpha, \dot{\alpha})$.

Based on (35), the model based control law (34) can be reformulated as

$$\tau = X(x_1, x_2, \alpha, \dot{\alpha})\theta^* - k_2 e_2 - (e_2^\top)^+ L. \tag{36}$$

Since θ^* is unknown and only $\hat{\theta}$ is available, based on the double regressor matrices technique, a TF-CL-aided parameter estimation update law for the BLF-based controller is designed as

$$\dot{\hat{\theta}} = -\Gamma X^\top e_2 - \Gamma k_t Y_f^\top e_f - \sum_{j=1}^{P} \Gamma k_h Y_{f_j}^\top e_{f_j}. \tag{37}$$

Comparing the difference between (22) and (37), the first term of (37) is designed as a stabilizing term, which serves to provide the stability proof in Theorem 2. By adjusting the values of Γ, k_t, and k_h, the importance of each part to the parameter estimation update law is traded off. Finally, based on the estimated parameter vector $\hat{\theta}$ from the TF-CL method, the stabilizing control law (36) can be rewritten as

$$\tau = X(x_1, x_2, \alpha, \dot{\alpha})\hat{\theta} - k_2 e_2 - \left(e_2^\top\right)^+ L. \tag{38}$$

Remark 4: Observing (2) and (35), we find that these two equations share the same coefficient vector θ^* but with different regressor matrices. The double regressor matrices technique illustrated here makes a combination of the TF-CL method and the BLF-based control strategy feasible. $Y(q, \dot{q}, \ddot{q})$ is a regressor matrix fully dependents on the model structure. $X(x_1, x_2, \alpha, \dot{\alpha})$ is a regressor matrix constructed based on both model properties and the stabilizing function α.

In the reaming part of this section, the main conclusions of this article and the corresponding proofs are given based on the parameter estimation update law (37) and the stabilizing control law (38).

Theorem 2: Consider an n-link robot manipulator in (1), the parameter estimation update law (37) and the control policy (38). Given Assumptions 1 and 2, for initial values of the system output and the tracking error lying in the predefined

safety set (7) and performance set (8), the following properties hold.

1) The tracking error e_1, error e_2, and parameter estimation error $\tilde{\theta}$ are stable and converge to 0 asymptotically.
2) The tracking error e_1 is bounded by Ω_{e_1}, where

$$\Omega_{e_1} = \left\{ e_1 \in \mathbb{R}^n : -\underline{U}_{e_1} \leq e_1 \leq \overline{U}_{e_1} \right\} \in \mathcal{P} \quad (39)$$

where $\underline{U}_{e_1} = [\underline{U}_{e_{1_i}}, \ldots, \underline{U}_{e_{1_n}}]^\top \in \mathbb{R}^n$, $\underline{U}_{e_{1_i}} = k_{a_i}\sqrt{[(2V(0))/(1+2V(0))]}$; $\overline{U}_{e_1} = [\overline{U}_{e_{1_i}}, \ldots, \overline{U}_{e_{1_n}}]^\top \in \mathbb{R}^n$, $\overline{U}_{e_{1_i}} = k_{b_i}\sqrt{[(2V(0))/(1+2V(0))]}$; $V(0)$ is the value of the BLF at $t = 0$. The error e_2 remains in the compact set Ω_{e_2}

$$\Omega_{e_2} = \left\{ e_2 \in \mathbb{R}^n : \|e_2\| \leq \sqrt{\frac{2V(0)}{\lambda_{\min}(M)}} \right\}. \quad (40)$$

3) For all $t > 0$, there holds $y(t) \in \Omega_y$, where

$$\Omega_y = \left\{ y \in \mathbb{R}^n : -\underline{U}_{e_1} - \underline{y}_d \ll y \ll \overline{U}_{e_1} + \overline{y}_d \right\} \in \bar{\mathcal{S}}. \quad (41)$$

Proof: See Appendix B. ■

VI. Simulation and Experimental Results

In this section, both numerical simulations and experiments are implemented to show the effectiveness of the proposed parameter estimation update law (37) and the BLF-based control strategy (38). Numerical simulations are first conducted on a popular 2-DoF robot manipulator [40], [52] in two cases: 1) a static coefficient vector and 2) a time-varying coefficient vector. The 2-DoF robot manipulator model can serve as a benchmark problem to test whether the TF-CL method can enable the estimated parameters to converge to the desired values. Then, experiments are applied to a more complex 3-DoF robot manipulator to show the superiority of our method. Unlike the numerical simulations presented in Section VI-A, the desired values of θ^* is unknown for the 3-DoF robot manipulator case. By using the collected historical data, the parameter convergence result is shown, and the satisfaction of both safety and performance criteria are achieved.

A. Illustrated Example on 2-DoF Robot Manipulator

1) 2-DoF Robot Manipulator With Static Coefficient Vector: To test the effectiveness of the proposed method, a 2-DoF robot manipulator is chosen as a benchmark example. For brevity, denoting $c_2 = \cos q_2$, $s_2 = \sin q_2$. The white-box model of the 2-DoF robot manipulator is given as

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + F\dot{q} = \tau \quad (42)$$

where $q = [q_1, q_2]^\top \in \mathbb{R}^2$, $\dot{q} = [\dot{q}_1, \dot{q}_2]^\top \in \mathbb{R}^2$, $M(q) = \begin{bmatrix} p_1 + 2p_3c_2 & p_2 + p_3c_2 \\ p_2 + p_3c_2 & p_2 \end{bmatrix} \in \mathbb{R}^{2\times 2}$, $C(q,\dot{q}) = \begin{bmatrix} -p_3\dot{q}_2s_2 & -p_3(\dot{q}_1+\dot{q}_2)s_2 \\ p_3\dot{q}_1s_2 & 0 \end{bmatrix} \in \mathbb{R}^{2\times 2}$, and $F = \begin{bmatrix} f_1 & 0 \\ 0 & f_2 \end{bmatrix} \in \mathbb{R}^{2\times 2}$.

The filtered regressor matrices $Y_1(q,\dot{q}) \in \mathbb{R}^{2\times 5}$ and $Y_2(q,\dot{q}) \in \mathbb{R}^{2\times 5}$ and the coefficient vector $\theta^* \in \mathbb{R}^5$ for the TF-CL method are given as

$$Y_1(q,\dot{q}) = \begin{bmatrix} \dot{q}_1 & \dot{q}_2 & 2\dot{q}_1c_2 + \dot{q}_2c_2 & 0 & 0 \\ 0 & \dot{q}_1 + \dot{q}_2 & \dot{q}_1c_2 & 0 & 0 \end{bmatrix} \quad (43)$$

$$Y_2(q,\dot{q}) = \begin{bmatrix} 0 & 0 & 0 & \dot{q}_1 & 0 \\ 0 & 0 & \dot{q}_1\dot{q}_2s_2 + \dot{q}_1^2s_2 & 0 & \dot{q}_2 \end{bmatrix} \quad (44)$$

$$\theta^* = \begin{bmatrix} p_1 & p_2 & p_3 & f_1 & f_2 \end{bmatrix}^\top. \quad (45)$$

For simulation, according to [52], the desired values of θ^* are set as $p_1 = 3.473$, $p_2 = 0.196$, $p_3 = 0.242$, $f_1 = 5.3$, and $f_2 = 1.1$. The regressor matrix $X(q,\dot{q},\alpha,\dot{\alpha}) \in \mathbb{R}^{2\times 5}$ in (38) follows:

$$X(q,\dot{q},\alpha,\dot{\alpha}) = \begin{bmatrix} \dot{\alpha}_1 & 0 & X_1 & \dot{q}_1 & 0 \\ 0 & \dot{\alpha}_1 + \alpha_2 & X_2 & 0 & \dot{q}_2 \end{bmatrix} \quad (46)$$

where $\alpha = [\alpha_1, \alpha_2]^\top \in \mathbb{R}^2$, $X_1 = 2\dot{\alpha}_1c_2 + \dot{\alpha}_2c_2 - 1/2\dot{q}_2^2s_2 - \dot{q}_1\dot{q}_2s_2 + (1/2\alpha_2 - \alpha_1)\dot{q}_2s_2$ and $X_2 = \dot{\alpha}_1c_2 + \dot{q}_1^2s_2 + 1/2\dot{q}_1\dot{q}_2s_2 - 1/2\dot{\alpha}_1$. In simulation, hyperparameters for the TF-CL part are set as $\Gamma = I_{5\times 1}$, $k_t = 200$, $k_h = 0.001$, $P = 7$, and $\varepsilon = 0.1$. The time constant of the filter is set as $k = 0.001$. The initial values of the estimated parameters are set as $\hat{\theta}(0) = [0, 9, 6, 0, 0]^\top$.

As for the ABLF-based control law (38), the control parameters are set as $k_1 = \mathrm{diag}(40, 30)$ and $k_2 = \mathrm{diag}(30, 40)$. $y_d = [\sin 0.5t, 2\cos 0.5t]^\top$ is chosen as the desired trajectory. The required safety and performance issues for the two joints are set as follows. For joint 1, the safety set is chosen as $\bar{\mathcal{S}}_1 = \{-1.17 < q_1 < 1.2\}$, the performance set follows

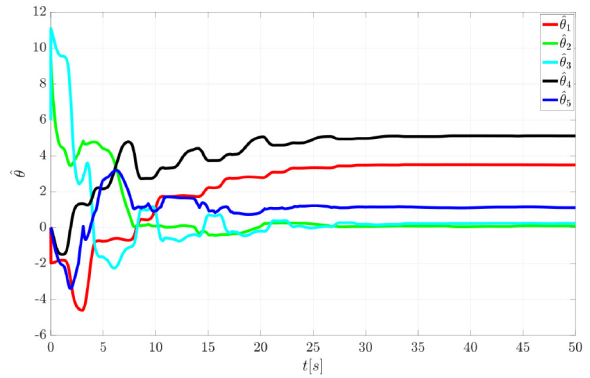

Fig. 2. Evolution of values for the estimated parameter vector $\hat{\theta}$ using the TF-CL method.

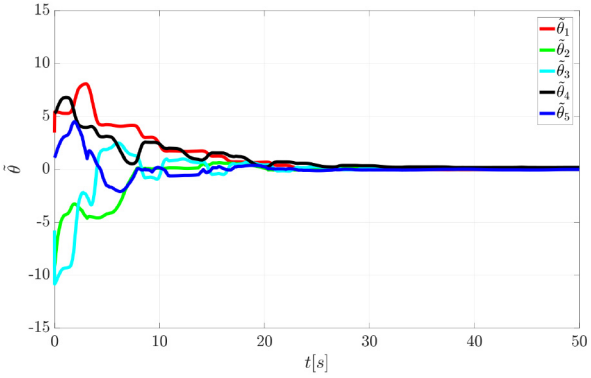

Fig. 3. Evolution of values for the parameter estimation errors $\tilde{\theta}$ using the TF-CL method.

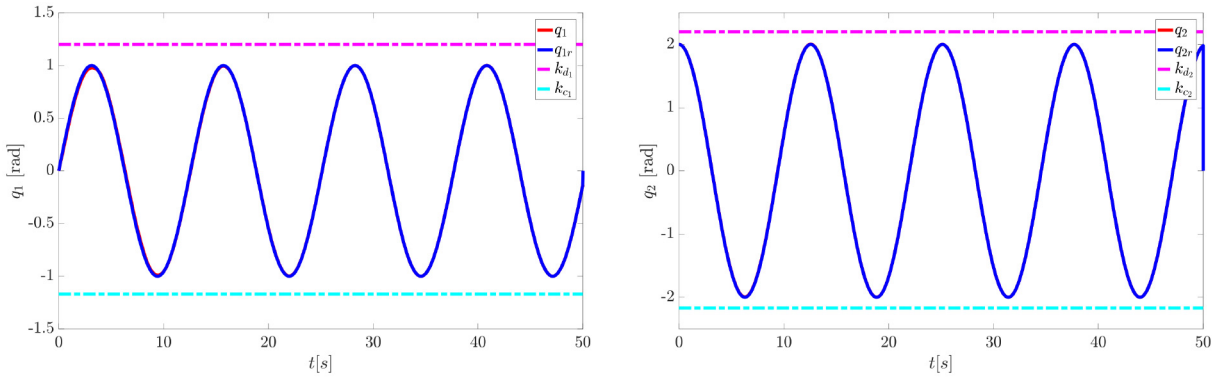

Fig. 4. Trajectories of q_1, q_2 and their reference trajectories q_{1r}, q_{2r} under the proposed control strategy (38). (a) Trajectory of q_1 and reference q_{1r}, safety bound k_{d_1} and k_{c_1}. (b) Trajectory of q_2 and reference q_{2r}, safety bound k_{d_2} and k_{c_2}.

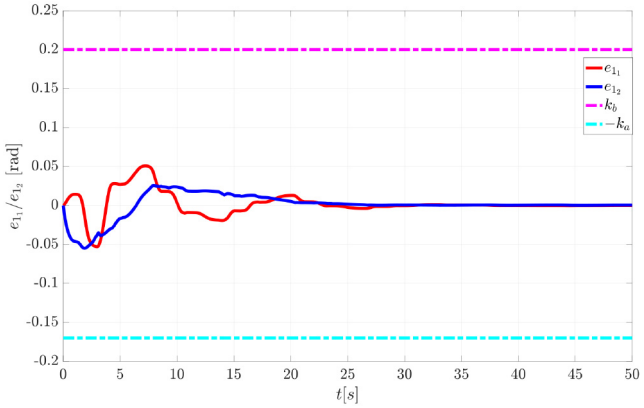

Fig. 5. Trajectories of the tracking errors of joint 1 e_{1_1} and joint 2 e_{1_2}, performance bound k_b and $-k_a$.

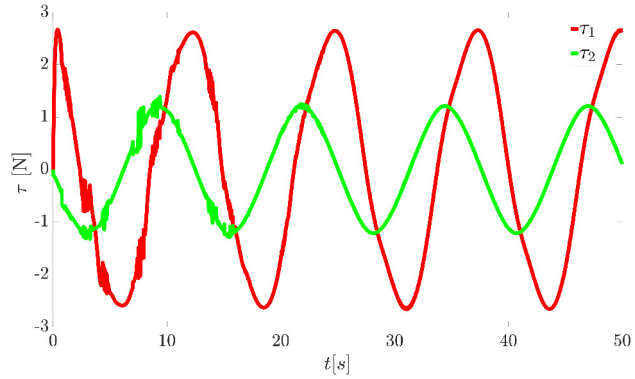

Fig. 6. Trajectories of the control τ under the proposed control strategy (38).

$\mathcal{P}_1 = \{-0.17 < e_{1_1} < 0.2\}$; For joint 2, the safety set is set as $\bar{\mathcal{S}}_2 = \{-2.17 < q_2 < 2.2\}$, the performance set is chosen as $\mathcal{P}_2 = \{-0.17 < e_{1_2} < 0.2\}$. To ensure that the 2-DoF robot manipulator track the desired trajectory while satisfying the above-illustrated safety and performance criteria, parameters are set as $k_c = [-1.17, -2.17]^\top$, $k_d = [1.2, 2.2]^\top$, $k_a = [0.17, 0.17]^\top$, and $k_b = [0.2, 0.2]^\top$. In order to ensure initial values lie in the corresponding safety and performance sets, we choose $x_1(0) = [0, 2]^\top$, and $x_2(0) = [0.5, 0]^\top$.

The parameter estimation update law (37) is adopted for the online estimation of the unknown coefficient vector θ^*. In Fig. 2, without incorporating external noises to satisfy the PE condition, the estimated parameters converge to the desired values. The corresponding parameter estimation errors are shown in Fig. 3, where they finally converge to a small neighborhood around 0, which means that an estimated model with high quality is gotten. However, even though a good tracking performance is achieved in [20], the norm of estimated weights does not converge.The proposed control law (38) is applied to the system and the simulation results are shown as follows. Fig. 4(a) and (b) shows that trajectories of q_1 and q_2 can follow their desired trajectories q_{1r} and q_{2r} precisely. The safety set $\bar{\mathcal{S}}_1$ for q_1 with upper bound k_{d_1} and lower bound k_{c_1}, and safety set $\bar{\mathcal{S}}_2$ for q_2 with upper bound k_{d_2} and lower bound k_{c_2} are never be violated during the tracking process.The tracking errors of the two joints are shown in Fig. 5. Tracking errors of the two joints finally converge to 0 and always lie in the required performance set $\mathcal{P}_1$ and $\mathcal{P}_2$, respectively. Note that in simulation, $\mathcal{P}_1$ and $\mathcal{P}_2$ share the same upper bound k_b and lower bound $-k_a$. The control trajectory is shown in Fig. 6, where τ oscillates when the estimated parameter vector is in the converging process. From the above analysis, it is concluded that the proposed parameter estimation update law (37) guarantees that the estimated parameters converge to their desired values. The control strategy given in (38) can drive the 2-DoF robot manipulator to track the reference trajectory precisely and satisfy the requirements of safety and performance together.

2) 2-DoF Robot Manipulator With Time-Varying Coefficient Vector: For the 2-DoF robot manipulator, to test the effectiveness of the proposed parameter estimation update law to counter time-varying uncertainties in terms of unknown payloads or friction parameters, the initial desired coefficient vector $\theta^* = [3.473, 0.196, 0.242, 5.3, 1.1]^\top$ is randomly reset as a new desired parameter vector $\theta_d^* = [4, 0.2, 0.3, 5.6, 0.8]^\top$ at time $t = 50s$. In practice, only parts of the coefficient vector will change due to environmental effects on the mass, length, or friction parameters, a hard disturbance is chosen here

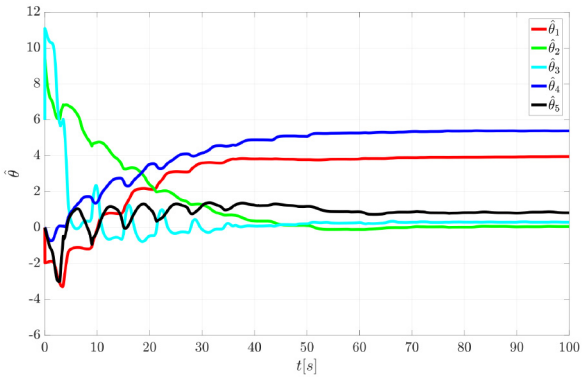

Fig. 7. Evolution of values for the estimated parameter vector $\hat{\theta}$, where an additional disturbance is incorporated at $t = 50$ s.

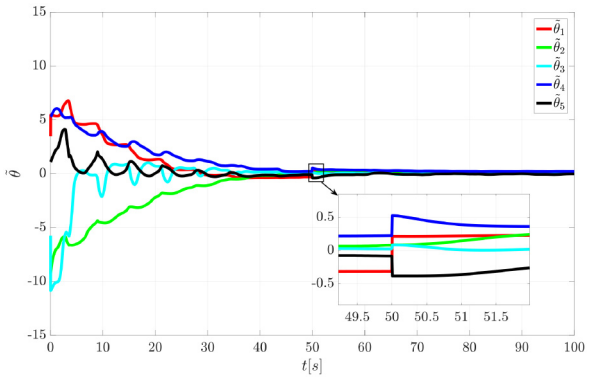

Fig. 8. Evolution of the parameter estimation error $\tilde{\theta}$, where an additional disturbance is incorporated at $t = 50$ s.

to show the superiority of the proposed method. Simulation results with a time-varying coefficient vector are shown from Figs. 7–11.

Fig. 7 shows trajectories of the estimated parameters, where an additional disturbance is incorporated at $t = 50$ s. As displayed in Fig. 7, in the first 50 s, the estimated parameters converge to the initial desired values of θ^*. When an additional disturbance is added at $t = 50$ s, the TF-CL method collects new data and enables the estimated parameters to finally converge to the new desired values of θ_d^*. As shown in Fig. 8, the trajectories of parameter estimation errors abruptly change at $t = 50s$ when an additional disturbance is added. Then, the parameter estimation errors still converge to a small neighborhood around zero. Fig. 9(a) and (b) demonstrates that under the proposed control strategy (38), two joints track their reference trajectories efficiently and will not violate their corresponding predefined safety sets $\bar{\mathcal{S}}_1$ and $\bar{\mathcal{S}}_2$ even when additional disturbance is added. The tracking errors of the two joints are shown in Fig. 10. It can be observed that the tracking errors e_{1_1} and e_{1_2} oscillate when the disturbance is added at time $t = 50$ s, then they finally converge to 0. The tracking errors always lie in the given performance set defined by lower bound $-k_a$ and upper bound k_b. The control trajectories given in Fig. 11 could provide additional information about the influence of the time-varying uncertainties on the control strategy. When the disturbance is added at $t = 50$ s, the magnitude of control τ_1 increases, and the magnitude of control τ_2 decreases in order to drive the robot to track the desired trajectory.

B. Experimental Validation on 3-DoF Robot Manipulator

Here, the proposed control strategy and the parameter estimation update law are applied to a 3-DoF robot manipulator to show their effectiveness. The 3-DoF robot manipulator (see Fig. 12), created by the Chair of Automatic Control Engineering (LSR), Technical University of Munich (TUM), is adopted in this article. The experiment configurations are as follows. The manipulator is actuated by 3 Maxon torque motors with a turn ratio of 1:100. The incremental encoders offer the joint position measurements with a resolution of 2000. The sensors and actuators are connected with the computer using a peripheral component interconnect (PCI) communication card. The executable algorithm is created by MATLAB 2017a in Ubuntu 14.04 LTS with the first-order Euler solver at the sampling rate of 1 kHz. More detailed information about the manipulator kinematics and dynamics can be found in [53] and [54].

Since the robot manipulator is confined in the horizontal plane, the gravity term is omitted here. Thus, the E–L equation of the 3-DoF robot manipulator follows:

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + F\dot{q} = \tau \tag{47}$$

where

$$q = \left[q_1, q_2, q_3\right]^\top \in \mathbb{R}^3, \quad \dot{q} = \left[\dot{q}_1, \dot{q}_2, \dot{q}_3\right]^\top \in \mathbb{R}^3$$
$$M(q) = \begin{bmatrix} m_{11} & m_{12} & m_{13} \\ m_{12} & m_{22} & m_{23} \\ m_{13} & m_{23} & m_{33} \end{bmatrix} \in \mathbb{R}^{3\times 3}$$
$$F = \begin{bmatrix} f_1 & 0 & 0 \\ 0 & f_2 & 0 \\ 0 & 0 & f_3 \end{bmatrix} \in \mathbb{R}^{3\times 3}$$

and $C(q,\dot{q})\dot{q} = [N_1, N_2, N_3]^\top \in \mathbb{R}^3$.

For brevity, denoting $c_{23} = \cos(q_2 + q_3)$, $s_3 = \sin q_3$, and $s_{23} = \sin(q_2 + q_3)$; For each element of the inertial matrix $M(q)$, $m_{11} = p_1 + p_2 c_{23} + p_3 c_2 + p_4 c_3$, $m_{12} = p_5 + p_6 c_{23} + p_7 c_2 + p_4 c_3$, $m_{13} = p_8 + p_6 c_{23} + p_9 c_3$, $m_{22} = p_5 + p_{10} c_3$, $m_{23} = p_8 + p_9 c_3$, and $m_{33} = p_8$; Besides, $N_1 = -p_2(s_{23}\dot{q}_1\dot{q}_3 + s_{23}\dot{q}_2\dot{q}_3 + s_{23}\dot{q}_1\dot{q}_2) - p_3 s_2 \dot{q}_1\dot{q}_2 - p_4(s_3\dot{q}_1\dot{q}_3 + s_3\dot{q}_2\dot{q}_3) - p_6(s_{23}\dot{q}_2^2 + s_{23}\dot{q}_3^2) - p_7 s_2 \dot{q}_2^2 - p_9 s_3 \dot{q}_3^2$, $N_2 = p_7 s_2 \dot{q}_1^2 + p_6 s_{23}\dot{q}_1^2 - p_9 s_3 \dot{q}_3^2 - p_4 s_3 \dot{q}_1\dot{q}_3 - p_4 s_3 \dot{q}_2\dot{q}_3$, and $N_3 = p_9(s_3\dot{q}_1^2 + s_3\dot{q}_2^2) + p_6 s_{23}\dot{q}_1^2 + p_4 s_3 \dot{q}_1\dot{q}_2$. For brevity, as for viscous friction values, $f_1 = f_2 = f_3 = f$ is assumed.

According to the above-illustrated model structure, the unknown coefficient vector $\theta^* \in \mathbb{R}^{11}$ can be written as

$$\theta^* = \left[p_1, p_2, p_3, p_4, p_5, p_6, p_7, p_8, p_9, p_{10}, f\right]^\top. \tag{48}$$

The corresponding filtered regressor matrices $Y_1(q,\dot{q})$, $Y_2(q,\dot{q}) \in \mathbb{R}^{3\times 11}$ are given as

$$Y_1(q,\dot{q}) = [Y_{11}, Y_{12}, Y_{13}]^\top \tag{49}$$

where Y_{11}, Y_{12}, $Y_{13} \in \mathbb{R}^{1\times 11}$ are in the following forms:

$$Y_{11} = \left[\dot{q}_1, c_{23}\dot{q}_1, c_2\dot{q}_1, c_3\dot{q}_1 + c_3\dot{q}_2, \dot{q}_2, c_{23}\dot{q}_2 + c_{23}\dot{q}_3, c_2\dot{q}_2, \dot{q}_3, c_3\dot{q}_3, 0, 0\right]$$
$$Y_{12} = \left[0, 0, 0, c_3\dot{q}_1, \dot{q}_1 + \dot{q}_2, c_{23}\dot{q}_1, c_2\dot{q}_1, \dot{q}_3, c_3\dot{q}_3, c_3\dot{q}_2, 0\right]$$

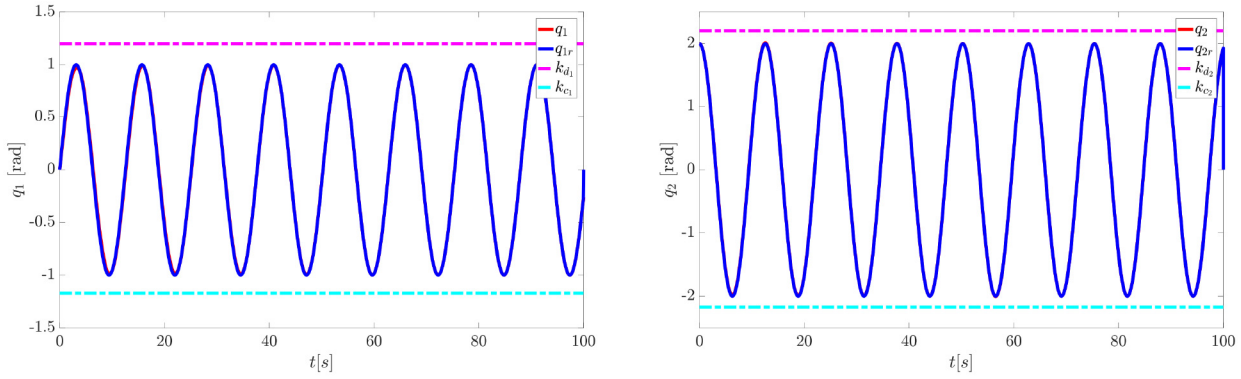

Fig. 9. Trajectories of q_1 and q_2 and their reference trajectories q_{1r} and q_{2r} where an additional disturbance is incorporated at $t = 50$ s. (a) Trajectory of q_1 and reference q_{1r}, safety bound k_{d_1} and k_{c_1}, where an additional disturbance is incorporated at $t = 50$ s. (b) Trajectory of q_2 and reference q_{2r}, safety bound k_{d_2} and k_{c_2}, where an additional disturbance is incorporated at $t = 50s$.

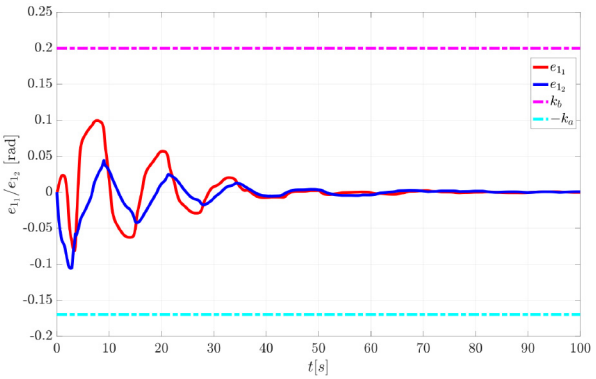

Fig. 10. Trajectories of the tracking errors e_{1_1} and e_{1_2}, performance bound k_b and $-k_a$, where an additional disturbance is incorporated at $t = 50$ s.

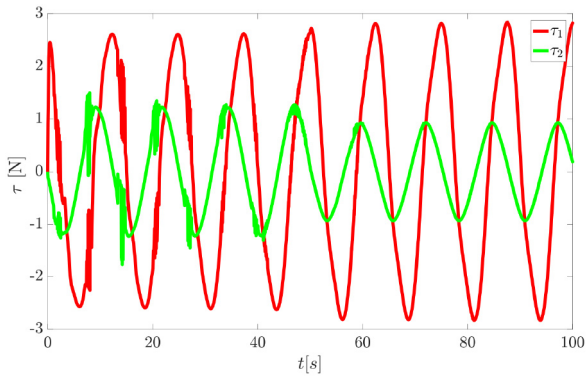

Fig. 11. Trajectories of the control τ, where an additional disturbance is incorporated at $t = 50$ s.

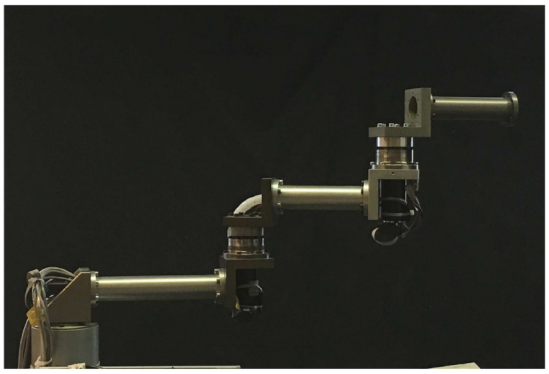

Fig. 12. 3-DoF Robot manipulator for experimental validation.

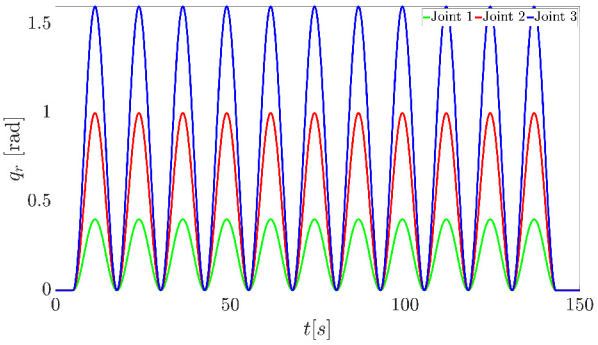

Fig. 13. Reference trajectories for three joints.

$$\text{and} \quad Y_{13} = \big[0, 0, 0, 0, 0, c_{23}\dot{q}_1, 0, \dot{q}_1 + \dot{q}_2 + \dot{q}_3, c_3\dot{q}_1 + c_3\dot{q}_2, 0, 0\big]$$

$$Y_2(q, \dot{q}) = [Y_{21}, Y_{22}, Y_{23}]^\top \tag{50}$$

where Y_{21}, Y_{22}, $Y_{23} \in \mathbb{R}^{1\times 11}$ follows:

$$Y_{21} = \big[0, s_{23}(\dot{q}_2 + \dot{q}_3)\dot{q}_1 - s_{23}\dot{q}_1\dot{q}_2 - s_{23}\dot{q}_1\dot{q}_3 - s_{23}\dot{q}_2\dot{q}_3, 0, 0, 0, s_{23}(\dot{q}_2 + \dot{q}_3)\dot{q}_2 + s_{23}(\dot{q}_2 + \dot{q}_3)\dot{q}_3 - s_{23}\dot{q}_2^2 - s_{23}\dot{q}_3^2, 0, 0, 0, 0, \dot{q}_1\big]$$

$$Y_{22} = \big[0, 0, 0, -s_3\dot{q}_2\dot{q}_3, 0, s_{23}(\dot{q}_2 + \dot{q}_3)\dot{q}_1 + s_{23}\dot{q}_1^2, s_2\dot{q}_1\dot{q}_2 + s_2\dot{q}_1^2, 0, 0, s_3\dot{q}_2\dot{q}_3, \dot{q}_2\big]$$

and

$$Y_{23} = \big[0, 0, 0, s_3\dot{q}_1\dot{q}_2, 0, s_{23}(\dot{q}_2 + \dot{q}_3)\dot{q}_1 + s_{23}\dot{q}_1^2, 0, 0, s_3\dot{q}_1\dot{q}_3 + s_3\dot{q}_2\dot{q}_3 + s_3\dot{q}_2^2 + s_3\dot{q}_1^2, 0, \dot{q}_3\big].$$

The regressor matrix $X(q, \dot{q}, \alpha, \dot{\alpha}) \in \mathbb{R}^{3\times 11}$ in (38) follows:

$$X(q, \dot{q}, \alpha, \dot{\alpha}) = [X_1, X_2, X_3]^\top \tag{51}$$

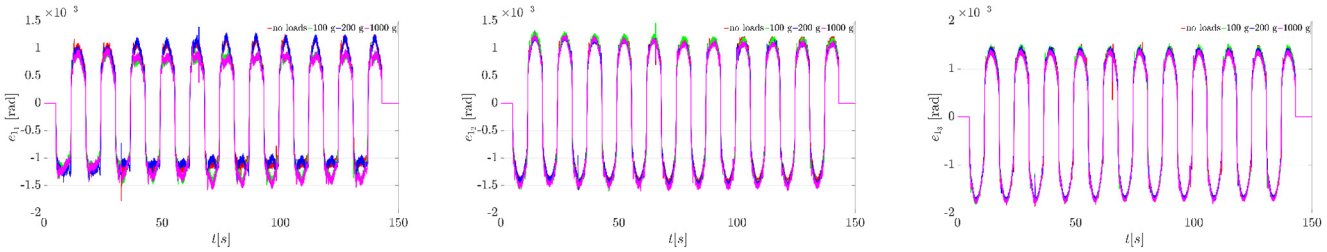

Fig. 14. Experimental results: trajectories of the tracking error e for three joints under different payloads. (a) Trajectories of the tracking error e_{1_1} under different payloads. (b) Trajectories of the tracking error e_{1_2} under different payloads. (c) Trajectories of the tracking error e_{1_3} under different payloads.

where $X_1, X_2, X_3 \in \mathbb{R}^{1\times 11}$ terms are defined as

$$\begin{aligned} X_1 = \big[&\dot{\alpha}_1, c_{23}\dot{\alpha}_1 - s_{23}\dot{q}_1\dot{q}_2 - s_{23}\dot{q}_1\dot{q}_3 - s_{23}\dot{q}_2\dot{q}_3 \\ &+ 0.5s_{23}(\dot{q}_2+\dot{q}_3)(\dot{q}_1-\alpha_1), c_2\dot{\alpha}_1 - s_2\dot{q}_1\dot{q}_2 \\ &+ 0.5s_2\dot{q}_2(\dot{q}_1-\alpha_1), c_3\dot{\alpha}_2 + c_3\dot{\alpha}_1 - s_3\dot{q}_1\dot{q}_3 \\ &- s_3\dot{q}_2\dot{q}_3 + 0.5s_3\dot{q}_3(\dot{q}_1-\alpha_1) + 0.5s_3\dot{q}_3(\dot{q}_2-\alpha_2) \\ &\dot{\alpha}_2, c_{23}\dot{\alpha}_2 + c_{23}\dot{\alpha}_3 - s_{23}\dot{q}_2^2 - s_{23}\dot{q}_3^2 + 0.5s_{23}(\dot{q}_2+\dot{q}_3) \\ &\times (\dot{q}_2-\alpha_2) + 0.5s_{23}(\dot{q}_2+\dot{q}_3)(\dot{q}_3-\alpha_3), c_2\dot{\alpha}_2 \\ &- s_2\dot{q}_2^2 + 0.5s_2\dot{q}_2(\dot{q}_2-\alpha_2), \dot{\alpha}_3, c_3\dot{\alpha}_3 - s_3\dot{q}_3^2 \\ &+ 0.5s_3\dot{q}_3(\dot{q}_3-\alpha_3), 0, \dot{q}_1\big] \\ X_2 = \big[&0,0,0, c_3\dot{\alpha}_1 - s_3\dot{q}_1\dot{q}_3 - s_3\dot{q}_2\dot{q}_3 + 0.5s_3\dot{q}_3(\dot{q}_1-\alpha_1) \\ &\dot{\alpha}_1+\dot{\alpha}_2, c_{23}\dot{\alpha}_1 + s_{23}\dot{q}_1^2 + 0.5s_{23}(\dot{q}_2+\dot{q}_3)(\dot{q}_1-\alpha_1) \\ &c_2\dot{\alpha}_1 + s_2\dot{q}_1^2 + 0.5s_2\dot{q}_2(\dot{q}_1-\alpha_1), \dot{\alpha}_3, c_3\dot{\alpha}_3 - s_3\dot{q}_3^2 \\ &+ 0.5s_3\dot{q}_3(\dot{q}_3-\alpha_3), c_3\dot{\alpha}_2 + 0.5s_3\dot{q}_3(\dot{q}_2-\alpha_2), \dot{q}_2\big] \end{aligned}$$

and

$$\begin{aligned} X_3 = \big[&0,0,0, s_3\dot{q}_1\dot{q}_2, 0, s_{23}\dot{q}_1^2 + c_{23}\dot{\alpha}_1 + 0.5s_{23}(\dot{q}_2+\dot{q}_3) \\ &(\dot{q}_1-\alpha_1), 0, \dot{\alpha}_1+\dot{\alpha}_2+\dot{\alpha}_3, c_3\dot{\alpha}_1 + c_3\dot{\alpha}_2 + s_3\dot{q}_1^2 \\ &+ s_3\dot{q}_2^2 + 0.5s_3\dot{q}_3(\dot{q}_1-\alpha_1) + 0.5s_3\dot{q}_3(\dot{q}_2-\alpha_2), 0, \dot{q}_3\big]. \end{aligned}$$

During the experiment, the 3-DoF robot manipulator is driven to track the desired sinusoidal trajectory $q_r \in \mathbb{R}^3$ that is designed as

$$q_r = \left(1 + \sin\left(\frac{t}{2} - \frac{\pi}{2}\right)\right)k_{\text{amp}}, 5 \le t \le 143 \tag{52}$$

where $k_{\text{amp}} = [0.2, 0.5, 0.8]^\top$ is the coefficient vector to distribute different amplitudes to each joint. The desired trajectories of the three joints are displayed in Fig. 13. Considering the required safety and performance issues, for joint 1, the safety set is set as $\bar{\mathcal{S}}_1 = \{-0.1 < q_1 < 0.52\}$, the performance set is chosen as $\mathcal{P}_1 = \{-0.1 < e_{1_1} < 0.12\}$; for joint 2, the safety set is designed as $\bar{\mathcal{S}}_2 = \{-0.1 < q_2 < 1.15\}$, the performance set follows $\mathcal{P}_2 = \{-0.1 < e_{1_2} < 0.15\}$. The safety set and performance set for joint 3 follow $\bar{\mathcal{S}}_3 = \{-0.15 < q_3 < 1.8\}$ and $\mathcal{P}_3 = \{-0.15 < e_{1_3} < 0.2\}$, respectively. To ensure that the 3-DoF robot manipulator tracks the desired trajectory while satisfying the above safety and performance criteria, parameters are set as $k_a = [0.1, 0.1, 0.15]^\top$, $k_b = [0.12, 0.15, 0.2]^\top$, $k_c = [-0.1, -0.1, -0.15]^\top$, and $k_d = [0.52, 1.15, 1.8]^\top$.

During the experiment, parameters for the TF-CL method are set as: $\Gamma = 0.06I_{11\times 1}$, $k_h = 0.4$, $k_t = 0.8$, $P = 15$, and $\varepsilon = 0.1$. Note that comparing to the simulation part, a smaller value of k_t is chosen here to suppress the influence of external noises existing in practical experiments. Time constant of the filter is set as $k = 0.001$. The initial values of the estimated parameters are set as $\hat{\theta}(0) = 0_{11\times 1}$. For the ABLF-based control law (38), control parameters are set as $k_1 = \text{diag}(20, 20, 20)$ and $k_2 = \text{diag}(25, 25, 25)$. The initial values are set as $x_1(0) = [0, 0, 0]^\top$ and $x_2(0) = [0, 0, 0]^\top$.

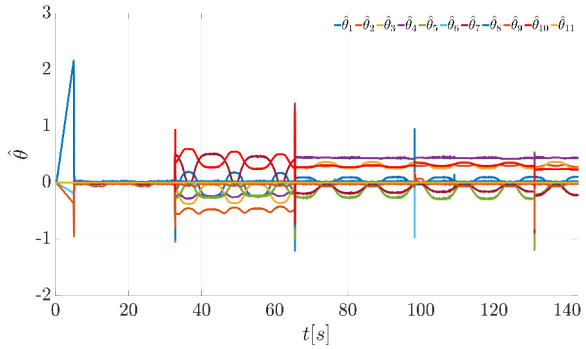

Fig. 15. Experimental results: Evolution of values for the estimated parameter vector $\hat{\theta}$ using the TF-CL method for without load case.

To verify the robustness of the proposed method, the experiment is conducted with different payloads under the above same parameter settings. The payloads are installed to the end effector of the manipulator. The tracking errors of the three joints with different payloads are displayed in Fig. 14(a)–(c), respectively. It can be observed that during the experiment, even under different payloads, the robot manipulator can track the desired trajectory well, tracking errors always lie in the required performance set $\mathcal{P}_i, i = 1, 2, 3$.For the robot manipulator without the payload case, the convergence result of the estimated parameters is shown in Fig. 15. As displayed in Fig. 15, in the beginning, the estimated parameters reach undesirable values with limited data. When rich enough data are recorded, the TF-CL technique effectively identifies the unknown parameters and parameter convergence is achieved. Note that the regular peaks of the estimated parameters are caused by encoder position drifting.

VII. Conclusion

This article presented a stable adaptive control strategy with guaranteed safety and performance based on ABLF, TF-CL, and backstepping methods. The TF-CL-based parameter

estimation update law guarantees that without incorporating external noises to satisfy the PE condition, the estimated parameters converge to the desired values fast. The information of joints acceleration is avoided by the torque filtering technique. Based on the estimated model, the proposed control strategy can drive the uncertain n-link robot manipulator to track the desired trajectory efficiently, while satisfying the requirements of safety and performance simultaneously. It is proven that the system output always remains in the predefined safety set, the tracking error is bounded by the performance set, and the parameter estimation error finally converges to 0 asymptotically. Both simulation and experimental results show the effectiveness of the proposed parameter estimation update law and the control strategy. To improve the generality and practicability of our method, the future work aims to extend the developed method to provide guaranteed performance and safety on full states even under the consideration of input saturation. Besides, the considered safety issue regarding the restricted operation range in this article allows us to integrate objectives of both safety and performance. The future work aims to extend the proposed method to tackle general safety concepts, e.g., collision avoidance with dangerous regions.

APPENDIX A
PROOF OF THEOREM 1

Let $V_{cl} : \mathbb{R}^m \to \mathbb{R}$ be a candidate continuously differential Lyapunov function defined as

$$V_{cl} = \frac{1}{2}\tilde{\theta}^\top \Gamma^{-1}\tilde{\theta}. \tag{53}$$

The bound of the Lyapunov function is

$$\frac{1}{2}\lambda_{\min}(\Gamma^{-1})\left\|\tilde{\theta}\right\|^2 \le V_{cl} \le \frac{1}{2}\lambda_{\max}(\Gamma^{-1})\left\|\tilde{\theta}\right\|^2. \tag{54}$$

Calculating time derivative of V_{cl} and substituting (22) into it yields

$$\begin{aligned}\dot{V}_{cl} &= \tilde{\theta}^\top \Gamma^{-1}\dot{\tilde{\theta}} = \tilde{\theta}^\top \Gamma^{-1}\dot{\hat{\theta}} \\ &= -k_t\tilde{\theta}^\top Y_f^\top e_f - \tilde{\theta}^\top \sum_{j=1}^{P} k_h Y_{f_j}^\top e_{f_j} \\ &= -k_t\tilde{\theta}^\top Y_f^\top Y_f\tilde{\theta} - \tilde{\theta}^\top \sum_{j=1}^{P} k_h Y_{f_j}^\top Y_{f_j}\tilde{\theta} \\ &\le -\tilde{\theta}^\top \sum_{j=1}^{P} k_h Y_{f_j}^\top Y_{f_j}\tilde{\theta} = -\tilde{\theta}^\top Q\tilde{\theta}\end{aligned} \tag{55}$$

where $Q = \sum_{j=1}^{P} k_h Y_{f_j}^\top Y_{f_j} \in \mathbb{R}^{m\times m}$. According to Assumption 2, Q is positive definite and $\lambda_{\min}(Q)$ is a positive constant. Thus, the following inequality holds:

$$\dot{V}_{cl} \le -\lambda_{\min}(Q)\left\|\tilde{\theta}\right\|^2. \tag{56}$$

It is concluded that the parameter estimation error will converge to zero asymptotically. This completes the proof.

APPENDIX B
PROOF OF THEOREM 2

Proof of 1): For the stability proof, let $Z = [e_1, e_2, \tilde{\theta}]^\top \in \mathbb{R}^{2n+m}$ and consider the following Lyapunov function:

$$V(Z) = V_{blf} + V_{cl}. \tag{57}$$

The time derivative of (57) based on (55) and (33) yields

$$\begin{aligned}\dot{V}(Z) &= \dot{V}_{blf} + \dot{V}_{cl} \\ &= -e_1^\top K_1 e_1 + L + e_2^\top\Big[\tau - C(x_1, x_2)x_2 - G(x_1) \\ &\qquad - Fx_2 - M(x_1)\dot{\alpha} + \frac{1}{2}\dot{M}(x_1)e_2\Big] \\ &\quad + \tilde{\theta}^\top\Gamma^{-1}\dot{\tilde{\theta}}.\end{aligned} \tag{58}$$

Substituting (35), (37) and (38) into (58) follows:

$$\begin{aligned}\dot{V}(Z) &= -e_1^\top K_1 e_1 + L + e_2^\top\left[X\hat{\theta} - k_2 e_2 - \left(e_2^\top\right)^+ L - X\theta^*\right] \\ &\quad + \tilde{\theta}^\top\Gamma^{-1}\dot{\hat{\theta}} \\ &= -e_1^\top K_1 e_1 - e_2^\top k_2 e_2 + e_2^\top X\tilde{\theta} + \tilde{\theta}^\top\Gamma^{-1} \\ &\quad \times\left[-\Gamma X^\top e_2 - \Gamma k_t Y_f^\top e_f - \sum_{j=1}^{P}\Gamma k_h Y_{f_j}^\top e_{f_j}\right] \\ &= -e_1^\top K_1 e_1 - e_2^\top k_2 e_2 - k_t\tilde{\theta}^\top Y_f^\top Y_f\tilde{\theta} - \tilde{\theta}^\top\sum_{j=1}^{P} k_h Y_{f_j}^\top \\ &\quad \times\left(Y_{f_j}\hat{\theta}_j - \tau_{f_j}\right) \\ &= -e_1^\top K_1 e_1 - e_2^\top k_2 e_2 - k_t\tilde{\theta}^\top Y_f^\top Y_f\tilde{\theta} - \tilde{\theta}^\top\sum_{j=1}^{P} k_h Y_{f_j}^\top \\ &\quad \times\left(Y_{f_j}\hat{\theta}_j - Y_{f_j}\theta^*\right) \\ &\le -e_1^\top K_1 e_1 - e_2^\top k_2 e_2 - \tilde{\theta}^\top\sum_{j=1}^{P} k_h Y_{f_j}^\top Y_{f_j}\tilde{\theta}.\end{aligned} \tag{59}$$

Let $N = \mathrm{diag}(K_1, k_2, Q) \in \mathbb{R}^{(2n+m)\times(2n+m)}$ with $Q = \sum_{j=1}^{P} k_h Y_{f_j}^\top Y_{f_j} \in \mathbb{R}^{m\times m}$, (59) can be rewritten as

$$\dot{V}(Z) \le -Z^\top NZ \le -\lambda_{\min}(N)\|Z\|^2 \tag{60}$$

where $\lambda_{\min}(N) = \min(\lambda_{\min}(K_1), \lambda_{\min}(k_2), \lambda_{\min}(Q))$. Finally, it is concluded that the tracking error e_1, the error e_2, and the parameter estimation error $\tilde{\theta}$ converge to zero asymptotically.

Proof of 2): Since $V(Z)$ is positive definite and $\dot{V}(Z) < 0$ according to (60), $V(Z) \le V(Z(0))$ establishes. From $V(Z) = V_1(e_1) + V_2(e_2) + V_{cl}(\tilde{\theta})$ and the fact that $V_2(e_2)$ and $V_{cl}(\tilde{\theta})$ are positive functions, it is concluded that $V_1(e_1) < V(Z(0))$, i.e., $V_1(e_1)$ is bounded. From characteristics of the ABLF (23), when $e_1 \to -k_a$ or $e_1 \to k_b$, $V_1(e_1) \to \infty$. The boundness of $V_1(e_1)$ means that $e_1 \ne -k_a$ or $e_1 \ne k_b$. Given that $-k_a \ll e_1(0) \ll k_b$, it is concluded that $-k_a \ll e_1(t) \ll k_b, \forall t > 0$, which means that the tracking error always lies in the required performance set (8). Besides, from the above analysis, we know that $V_1(e_1) < V(0)$. To get the bound of

e_1, first we take the ith element of e_1 as an example. For e_{1_i}, the following inequalities establish:

$$V(0) > \begin{cases} \frac{e_{1_i}^2}{2\left(k_{b_i}^2 - e_{1_i}^2\right)} & 0 < e_{1_i} < k_{b_i} \\ \frac{e_{1_i}^2}{2\left(k_{a_i}^2 - e_{1_i}^2\right)} & -k_{a_i} < e_{1_i} < 0. \end{cases} \tag{61}$$

It can be represented as the following equivalent form:

$$e_{1_i}^2 < \begin{cases} k_{b_i}^2 \frac{2V(0)}{1+2V(0)} & 0 < e_{1_i} < k_{b_i} \\ k_{a_i}^2 \frac{2V(0)}{1+2V(0)} & -k_{a_i} < e_{1_i} < 0. \end{cases} \tag{62}$$

From above it is concluded that for $e_{1_i} > 0$, $e_{1_i} < k_{b_i}\sqrt{[(2V(0))/(1+2V(0))]}$, and $e_{1_i} > -k_{a_i}\sqrt{[(2V(0))/(1+2V(0))]}$ when $e_{1_i} < 0$. Furthermore since $\sqrt{[(2V(0))/(1+2V(0))]} < 1$, $-k_{a_i} < -k_{a_i}\sqrt{[(2V(0))/(1+2V(0))]} < e_{1_i} < k_{b_i}\sqrt{[(2V(0))/(1+2V(0))]} < k_{b_i}$ establishes. Consider all elements of e_1 and the performance set $\mathcal{P}$ in (8), (39) establishes.

Consider the case of e_2, since $V_2(e_2) = 1/2e_2^\top M e_2 < V(0)$, $\|e_2\| \leq \sqrt{[(2V(0))/(\lambda_{\min}(M))]}$ establishes, i.e., e_2 remains in the set Ω_{e_2}.

Proof of 3): The output $y = x_1 = e_1 + y_d$. According to (39), $-\underline{U}_{e_1} \leq e_1 \leq \overline{U}_{e_1}$ establishes. We know that $-\underline{y}_d \leq y_d \leq \overline{y}_d$ from Assumption 1, it is easy to get that $-\underline{U}_{e_1} - \underline{y}_d \leq y \leq \overline{U}_{e_1} + \overline{y}_d$. Since $\underline{U}_{e_1} \ll k_a$ and $\overline{U}_{e_1} \ll k_b$, $-k_a - \underline{y}_d \ll -\underline{U}_{e_1} - \underline{y}_d \ll 0$ and $0 \ll \overline{U}_{e_1} + \overline{y}_d \ll k_b + \overline{y}_d$ establishes, i.e., $\Omega_y \in \bar{\mathcal{P}}$. Since $-k_a$ and k_b are chosen to satisfy $\bar{\mathcal{P}} \subseteq \bar{\mathcal{S}}$, $\Omega_y \in \bar{\mathcal{S}}$ also establishes, i.e., system outputs will not transgress the predefined safety set (7). This completes the proof.

REFERENCES

[1] D. P. Bertsekas, *Approximate Dynamic Programming*. Belmont, MA, USA: Athena Sci., 2012.

[2] F. Berkenkamp, R. Moriconi, A. P. Schoellig, and A. Krause, "Safe learning of regions of attraction for uncertain, nonlinear systems with Gaussian processes," in *Proc. IEEE 55th Conf. Decis. Control (CDC)*, Las Vegas, NV, USA, 2016, pp. 4661–4666.

[3] A. Bemporad, "Reference governor for constrained nonlinear systems," *IEEE Trans. Autom. Control*, vol. 43, no. 3, pp. 415–419, Mar. 1998.

[4] F. Blanchini, "Set invariance in control," *Automatica*, vol. 35, no. 11, pp. 1747–1767, 1999.

[5] A. D. Ames and M. Powell, "Towards the unification of locomotion and manipulation through control Lyapunov functions and quadratic programs," in *Control of Cyber-Physical Systems*. Heidelberg, Germany: Springer, 2013, pp. 219–240.

[6] P. Wieland and F. Allgöwer, "Constructive safety using control barrier functions," *IFAC Proc. Vol.*, vol. 40, no. 12, pp. 462–467, 2007.

[7] E. F. Camacho and C. B. Alba, *Model Predictive Control*. London, U.K.: Springer, 2013.

[8] H. J. Ferreau, H. G. Bock, and M. Diehl, "An online active set strategy to overcome the limitations of explicit MPC," *Int. J. Robust Nonlinear Control IFAC Affiliated J.*, vol. 18, no. 8, pp. 816–830, 2008.

[9] R. Kamalapurkar, P. Walters, and W. E. Dixon, "Model-based reinforcement learning for approximate optimal regulation," in *Control of Complex Systems*. Oxford, U.K.: Elsevier, 2016, pp. 247–273.

[10] R. Vinter, *Optimal Control*. Boston, MA, USA: Springer, 2010.

[11] F. L. Lewis, D. Vrabie, and V. L. Syrmos, *Optimal Control*. New York, NY, USA: Wiley, 2012.

[12] Y. Sui, A. Gotovos, J. W. Burdick, and A. Krause, "Safe exploration for optimization with Gaussian processes," in *Proc. 32nd Int. Conf. Mach. Learn. Res.*, vol. 37, 2015, pp. 997–1005.

[13] K. P. Tee, S. S. Ge, and E. H. Tay, "Barrier Lyapunov functions for the control of output-constrained nonlinear systems," *Automatica*, vol. 45, no. 4, pp. 918–927, 2009.

[14] H. K. Khalil and J. W. Grizzle, *Nonlinear Systems*, vol. 3. Upper Saddle River, NJ, USA: Prentice-Hall, 2002.

[15] W. Sun, Y. Liu, and H. Gao, "Constrained sampled-data ARC for a class of cascaded nonlinear systems with applications to motor-servo systems," *IEEE Trans. Ind. Informat.*, vol. 15, no. 2, pp. 766–776, Feb. 2019.

[16] K. B. Ngo, R. Mahony, and Z.-P. Jiang, "Integrator backstepping using barrier functions for systems with multiple state constraints," in *Proc. 44th IEEE Conf. Decis. Control*, Seville, Spain, 2005, pp. 8306–8312.

[17] K. P. Tee and S. S. Ge, "Control of nonlinear systems with full state constraint using a barrier Lyapunov function," in *Proc. 48th IEEE Conf. Decis. Control (CDC) Held Jointly 28th Chin. Control Conf.*, Shanghai, China, 2009, pp. 8618–8623.

[18] W. He, Z. Yin, and C. Sun, "Adaptive neural network control of a marine vessel with constraints using the asymmetric barrier Lyapunov function," *IEEE Trans. Cybern.*, vol. 47, no. 7, pp. 1641–1651, Jul. 2017.

[19] K. P. Tee, B. Ren, and S. S. Ge, "Control of nonlinear systems with time-varying output constraints," *Automatica*, vol. 47, no. 11, pp. 2511–2516, 2011.

[20] W. He, H. Huang, and S. S. Ge, "Adaptive neural network control of a robotic manipulator with time-varying output constraints," *IEEE Trans. Cybern.*, vol. 47, no. 10, pp. 3136–3147, Oct. 2017.

[21] S. S. Ge, C. C. Hang, T. H. Lee, and T. Zhang, *Stable Adaptive Neural Network Control*, vol. 13. New York, NY, USA: Springer, 2013.

[22] F. L. Lewis, D. M. Dawson, and C. T. Abdallah, *Robot Manipulator Control: Theory and Practice*. Boca Raton, FL, USA: CRC Press, 2003.

[23] J. Villadsen and M. L. Michelsen, *Solution of Differential Equation Models by Polynomial Approximation*, vol. 7. Englewood Cliffs, NJ, USA: Prentice-Hall, 1978.

[24] A. Zygmund, *Trigonometric Series*, vol. 1. Cambridge, U.K.: Cambridge Univ. Press, 2002.

[25] N. K. Bary, *A Treatise on Trigonometric Series*, vol. 1. New York, NY, USA: Elsevier, 2014.

[26] L. Schumaker, *Spline Functions: Basic Theory*. Cambridge, U.K.: Cambridge Univ. Press, 2007.

[27] L.-X. Wang and J. M. Mendel, "Fuzzy basis functions, universal approximation, and orthogonal least-squares learning," *IEEE Trans. Neural Netw.*, vol. 3, no. 5, pp. 807–814, Sep. 1992.

[28] S. N. Kumpati and P. Kannan, "Identification and control of dynamical systems using neural networks," *IEEE Trans. Neural Netw.*, vol. 1, no. 1, pp. 4–27, Mar. 1990.

[29] F. L. Lewis, K. Liu, and A. Yesildirek, "Neural net robot controller with guaranteed tracking performance," *IEEE Trans. Neural Netw.*, vol. 6, no. 3, pp. 703–715, May 1995.

[30] J. de Jesús Rubio, "SOFMLS: Online self-organizing fuzzy modified least-squares network," *IEEE Trans. Fuzzy Syst.*, vol. 17, no. 6, pp. 1296–1309, Dec. 2009.

[31] I. Elias *et al.*, "Genetic algorithm with radial basis mapping network for the electricity consumption modeling," *Appl. Sci.*, vol. 10, no. 12, p. 4239, 2020.

[32] B. Ren, S. S. Ge, K. P. Tee, and T. H. Lee, "Adaptive neural control for output feedback nonlinear systems using a barrier Lyapunov function," *IEEE Trans. Neural Netw.*, vol. 21, no. 8, pp. 1339–1345, Aug. 2010.

[33] W. He, Y. Chen, and Z. Yin, "Adaptive neural network control of an uncertain robot with full-state constraints," *IEEE Trans. Cybern.*, vol. 46, no. 3, pp. 620–629, Mar. 2016.

[34] B. Anderson, "Exponential stability of linear equations arising in adaptive identification," *IEEE Trans. Autom. Control*, vol. 22, no. 1, pp. 83–88, Feb. 1977.

[35] S. Boyd and S. S. Sastry, "Necessary and sufficient conditions for parameter convergence in adaptive control," *Automatica*, vol. 22, no. 6, pp. 629–639, 1986.

[36] K. J. Åström and B. Wittenmark, *Adaptive Control*. North Chelmsford, MA, USA: Courier Corp., 2013.

[37] J.-S. Lin and I. Kanellakopoulos, "Nonlinearities enhance parameter convergence in output-feedback systems," *IEEE Trans. Autom. Control*, vol. 43, no. 2, pp. 204–222, Feb. 1998.

[38] V. Adetola and M. Guay, "Finite-time parameter estimation in adaptive control of nonlinear systems," *IEEE Trans. Autom. Control*, vol. 53, no. 3, pp. 807–811, Apr. 2008.

[39] S. B. Roy and S. Bhasin, "Robustness analysis of initial excitation based adaptive control," in *Proc. IEEE 58th Conf. Decis. Control (CDC)*, Nice, France, 2019, pp. 7055–7062.

[40] R. Kamalapurkar, H. Dinh, S. Bhasin, and W. E. Dixon, "Approximate optimal trajectory tracking for continuous-time nonlinear systems," *Automatica*, vol. 51, pp. 40–48, Jan. 2015.

[41] S. Kersting and M. Buss, "Direct and indirect model reference adaptive control for multivariable piecewise affine systems," *IEEE Trans. Autom. Control*, vol. 62, no. 11, pp. 5634–5649, Nov. 2017.
[42] G. Chowdhary and E. Johnson, "Concurrent learning for convergence in adaptive control without persistency of excitation," in *Proc. 49th IEEE Conf. Decis. Control (CDC)*, Atlanta, GA, USA, 2010, pp. 3674–3679.
[43] R. Kamalapurkar, "Simultaneous state and parameter estimation for second-order nonlinear systems," in *Proc. IEEE 56th Annu. Conf. Decis. Control (CDC)*, Melbourne, VIC, Australia, 2017, pp. 2164–2169.
[44] H. Modares, F. L. Lewis, and M.-B. Naghibi-Sistani, "Adaptive optimal control of unknown constrained-input systems using policy iteration and neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 24, no. 10, pp. 1513–1525, Oct. 2013.
[45] A. D. Ames, X. Xu, J. W. Grizzle, and P. Tabuada, "Control barrier function based quadratic programs for safety critical systems," *IEEE Trans. Autom. Control*, vol. 62, no. 8, pp. 3861–3876, Aug. 2017.
[46] H. Sadeghian, L. Villani, M. Keshmiri, and B. Siciliano, "Task-space control of robot manipulators with null-space compliance," *IEEE Trans. Robot.*, vol. 30, no. 2, pp. 493–506, Apr. 2014.
[47] M. Saveriano and D. Lee, "Learning barrier functions for constrained motion planning with dynamical systems," 2020. [Online]. Available: arXiv:2003.11500.
[48] C. P. Bechlioulis and G. A. Rovithakis, "Robust adaptive control of feedback linearizable MIMO nonlinear systems with prescribed performance," *IEEE Trans. Autom. Control*, vol. 53, no. 9, pp. 2090–2099, Oct. 2008.
[49] I. M. Mitchell, "Comparing forward and backward reachability as tools for safety analysis," in *Proc. Int. Workshop Hybrid Syst. Comput. Control*, 2007, pp. 428–443.
[50] G. Welch and G. Bishop, "An introduction to the Kalman filter," Dept. Comput. Sci., Univ. North Carolina, Chapel Hill, NC, USA, Rep. TR 95-041, 1995.
[51] G. Tao, *Adaptive Control Design and Analysis*, vol. 37. Hoboken, NJ, USA: Wiley, 2003.
[52] A. Parikh, R. Kamalapurkar, and W. E. Dixon, "Integral concurrent learning: Adaptive control with parameter convergence without pe or state derivatives," 2015. [Online]. Available: arXiv:1512.03464.
[53] Z. Zhang, M. Leibold, and D. Wollherr, "Integral sliding-mode observer-based disturbance estimation for Euler–Lagrangian systems," *IEEE Trans. Control Syst. Technol.*, vol. 28, no. 6, pp. 2377–2389, Nov. 2020.
[54] R. Hayat, "Model-free robust-adaptive controller design and identification for robot manipulators," Ph.D. dissertation, Dept. Elect. Comput. Eng., Technische Universität München, Munich, Germany, 2019.

Cong Li received the M.Sc. degree in control science and engineering from Xi'an Jiaotong University, China, in 2018. He is currently pursuing the Ph.D. degree in automatic control with the Chair of Automatic Control Engineering, Technical University of Munich, Munich, Germany.

He is also a Research Associate with the Chair of Automatic Control Engineering, Technical University of Munich. His research interests include reinforcement learning, robust control, constraint optimization, and robotics.

Fangzhou Liu (Member, IEEE) received the M.Sc. degree in control theory and engineering from the Harbin Institute of Technology, Harbin, China, in 2014, and the Doktor-Ingenieur degree in electrical engineering from the Technical University of Munich, Munich, Germany, in 2019.

He is currently a Lecturer and a Research Fellow with the Chair of Automatic Control Engineering, Technical University of Munich. His current research interests include networked control systems, modeling, analysis, and control on social networks, concurrent learning, and their applications.

Dr. Liu has received the Dimitris N. Chorafas Prize and the Promotionspreis der Fakultät für Elektrotechnik und Informationstechnik for his Ph.D. thesis.

Yongchao Wang received the M.Sc. degree in control science and engineering from Xidian University, Xi'an, China, in 2016. He is currently pursing the Ph.D. degree in automatic control with the Chair of Automatic Control Engineering, Technical University of Munich, Munich, Germany.

He is also a Research Associate with the Chair of Automatic Control Engineering, Technical University of Munich. His research interests include adaptive control, robust nonlinear control, optimal control, backstepping, and robotics.

Martin Buss (Fellow, IEEE) received the Diploma Engineering degree in electrical engineering from the Technische Universität Darmstadt, Darmstadt, Germany, in 1990, and the Doctor of Engineering degree in electrical engineering from the University of Tokyo, Tokyo, Japan, in 1994.

In 1988, he was a Research Student with the Science University of Tokyo, Tokyo, for one year. From 1994 to 1995, he was a Postdoctoral Researcher with the Department of Systems Engineering, The Australian National University, Canberra, ACT, Australia. From 1995 to 2000, he was a Senior Research Assistant and a Lecturer with the Chair of Automatic Control Engineering, Department of Electrical Engineering and Information Technology, Technical University of Munich, Munich, Germany. From 2000 to 2003, he was a Full Professor, the Head of the Control Systems Group, and the Deputy Director of the Faculty IV, Electrical Engineering and Computer Science, Institute of Energy and Automation Technology, Technical University of Berlin, Berlin, Germany. Since 2003, he has been a Full Professor (Chair) with the Chair of Automatic Control Engineering, Faculty of Electrical Engineering and Information Technology, Technical University of Munich, where he has been with the Medical Faculty since 2008. His research interests include automatic control, mechatronics, multimodal human system interfaces, optimization, nonlinear, and hybrid discrete-continuous systems.

Prof. Buss has been awarded the ERC Advanced Grant SHRINE.