Subsystem-Based Control with Modularity for Strict-Feedback Form Nonlinear Systems

Janne Koivumäki[1], Jukka-Pekka Humaloja[2], Lassi Paunonen[2], Wen-Hong Zhu[3] and Jouni Mattila[1]

***Abstract*—This study proposes an adaptive *subsystem-based control* (SBC) for systematic and straightforward nonlinear control of *n*th-order strict-feedback form (SFF) systems. By decomposing the SFF system to subsystems, a generic term (namely *stability connector*) can be created to address dynamic interactions between the subsystems. This 1) enables modular control design with global asymptotic stability, 2) such that the control design and its stability analysis can be performed locally at a subsystem level, 3) while avoiding an excessive growth of the control design complexity when the system order *n* increases. The above properties make the proposed method suitable especially for high-order systems. We also design a smooth projection function for the system parametric uncertainties. The efficiency of the method is demonstrated in simulations with a nonlinear 5th-order system.**

***Index Terms*—Nonlinear control, model-based control, adaptive control, global asymptotic stability, modular control.**

I. Introduction

NONLINEAR model-based control aims to design a specific feedforward (FF) compensation term based on the system inverse dynamics to generate the control output(s) from the system states and desired input signals [1]. If the FF compensation can exactly capture the inverse of the plant dynamics for all frequencies, an infinite control bandwidth with zero tracking error becomes theoretically possible [2], [3]. While early control methods, e.g., *feedback linearization* [4], aimed to cancel (or linearize) the system nonlinearities, *adaptive backstepping* [5] became a significant breakthrough in nonlinear systems control by incorporating the nonlinearities towards ideal FF compensation with *global asymptotic stability*.

This study proposes globally asymptotically stable adaptive *subsystem-based control* (SBC) for *n*th-order strict-feedback form (SFF) systems. The proposed method has built-in modularity and it avoids excessive growth of the control design complexity when the system order *n* increases (an issue reported for backstepping-based methods in several studies [6]–[14]). *Dynamic surface control* (DSC) [6], [7], *adaptive* DSC [8] and *neural network* (NN) *adaptive DSC* [9] are previously developed as an alternative to backstepping to avoid the reported "explosion of complexity" with semi-global stability. They are based on *multiple sliding surface* control [15], [16] (a method similar to backstepping) using a series of low-pass filters [7].

The research was funded by Academy of Finland grants 335569 (J. Mattila) and 310489 (L. Paunonen); and the Wihuri Foundation (J.-P. Humaloja).

[1]J. Koivumäki and J. Mattila are with Faculty of Engineering and Natural Sciences, Tampere University, Finland. e-mail: firstname.surname@tuni.fi

[2]J.-P. Humaloja and L. Paunonen are with Mathematics, Faculty of Information Technology and Communication Sciences, Tampere University, Finland. e-mail: firstname.surname@tuni.fi

[3]W.-H. Zhu is with the Canadian Space Agency, Canada. e-mail: Wen-Hong.Zhu@canada.ca

More recently, an *adaptive DSC* with nonlinear filtering [10] and *NN adaptive DSC* [11] achieved semi-global asymptotic convergence, improving the bounded output trajectory tracking of the previous DSC methods (see [11]). In [12], a *predefined-time DSC* (non-adaptive method) achieved global stability. Also backstepping-based *command filtered methods* for *NN adaptive control* [13] and *finite-time adaptive control* [14] are proposed to overcome the "explosion of complexity", however, without asymptotic convergence. Our method involves either filtering or NNs, and is globally asymptotically stable. While the studies in [6]–[14] used a 3rd-order system in numerical validations, we provide the results with a 5th-order system.

The proposed method originates from *virtual decomposition control* (VDC) [3], [17] that is developed for controlling complex robotic systems. *Modularity* is one of the key aspects in addressing complexity in advanced control realizations [18], [19, Sec. IV]. In VDC, robotic systems are virtually decomposed into *modular subsystems* (rigid links and joints) such that both control design and stability analysis can be performed locally at the subsystem (SS) level to guarantee overall global asymptotic stability. In particular, VDC introduced *virtual power flows* (VPFs) [3, Def. 2.16] to define dynamic interactions between the adjacent SSs such that the VPFs cancel each others out when the SSs are connected. However, when applied beyond robotics, the interactions between SSs will no longer be described by VPFs [20]. Some early ideas for the proposed method originate from the application-oriented paper in [20]. In addition, some ideological similarities can be seen to [21] that designs strictly passive interaction dynamics for adjacent SSs in SFF; see Remark 4.4 for more details.

As the main contribution, the proposed method generalizes the "subsystem-based control philosophy" in [3], [20] for the *n*th-order SFF systems. After defining a generic form for SSs, we design a specific *stability connector* (a generic spill-over term in SS stability analysis in Def. 4.1) to address dynamic interactions between the adjacent SSs. We show that every SS with a "stability preventing" connector is compensated by the subsequent SS with a corresponding "stabilizing" connector. Similarly to VDC, we formulate a generic definition for *virtual stability*[1] such that when every SS is virtually stable, the overall system becomes automatically globally asymptotically stable. Instead of using Lebesque L_2/L_∞ integrable functions as in [3], [20], [21], we base the results on Lyapunov functions. The proposed method is modular in the sense that control laws for every SS can be designed with a single generic-form equation;

[1]In terms of Lyapunov functions, definition of *virtual stability* (see Def. 4.2 in Section IV) includes quadratic terms for asymptotic convergence added with stability connector(s) for compensating/stabilizing dynamics of adjacent SSs.

see Remark 3.2. As part of the control, we design a smooth projection function for the system parametric uncertainties.

Next, Section II introduces the control problem. Section III formulates the proposed method. Section IV provides in-depth analysis on the control design and its stability. Section V provides numerical validation. Section VI concludes the study.

II. The Control Problem

Consider the following nth-order SFF system

$$\begin{cases} \theta_{11}\dot{x}_1(t) = f_1(x_1,t) + g_1(x_1,t)x_2(t) & (1)\\ \theta_{i1}\dot{x}_i(t) = f_i(\boldsymbol{x}_i,t) + g_i(\boldsymbol{x}_i,t)x_{i+1}(t),\ \forall i \in \{2,...,n-1\} & (2)\\ \theta_{n1}\dot{x}_n(t) = f_n(\boldsymbol{x}_n,t) + g_n(\boldsymbol{x}_n,t)u(t) & (3)\end{cases}$$

where $\boldsymbol{x}_k = [x_1(t), x_2(t), \cdots, x_k(t)]$, $\forall k \in \{1,\cdots,n\}$, $u(t)$ is the system input, $f_k(\boldsymbol{x}_k,t)$, $\forall k \in \{1,\cdots,n\}$, can be written as

$$f_k(\boldsymbol{x}_k,t) = \theta_{k2}\gamma_{k2}(\boldsymbol{x}_k,t) + \theta_{k3}\gamma_{k3}(\boldsymbol{x}_k,t) + \cdots + \theta_{kj}\gamma_{kj}(\boldsymbol{x}_k,t) \quad (4)$$

and $\theta_{k1}, \theta_{k2}, \cdots, \theta_{kj} > 0$ in (1)–(4) are constant parameters. We often omit the notation t for brevity. Similarly to backstepping, we assume that $g_k(\boldsymbol{x}_k)$ and $f_k(\boldsymbol{x}_k)$ (i.e, $\gamma_{k\zeta}(\boldsymbol{x}_k)$, $\forall \zeta \in \{2,\ldots,j\}$) are sufficiently smooth and $g_k(\boldsymbol{x}_k,t) \neq 0$ on $\mathbb{R}^k \times [0,\infty)$.

Throughout the paper, we use n to denote the system overall order, while it also denotes the last SS (or its element) in (3). We use $i \in \{2,\cdots,n-1\}$ to denote a SS (or its element) in the middle of the SFF sequence; see (2). We use k to denote an arbitrary decomposed SS (or its element), such that generic and modular form for SS_k (the kth SS) in (1)–(3) is given by

$$\theta_{k1}\dot{x}_k = f_k(\boldsymbol{x}_k) + g_k(\boldsymbol{x}_k)x_{k+1},\ \forall k \in \{1,...,n\} \quad (5)$$

where we denote $x_{n+1} = u$.

Let $x_{1\text{d}} \in C^{n-1}(0,\infty;\mathbb{R})$ be a desired trajectory for $x_1(t)$ such that $x_{1\text{d}}^{(n)}$ exists almost everywhere; here $C^{n-1}(0,\infty;\mathbb{R})$ denotes the space of $n-1$ times continuously differentiable real-valued functions over the interval $[0,\infty)$. In the sequel, we will use the shorthand notation C^k for the space of k times continuously differentiable functions over their interval of definition. Next, we aim to design a control for (1)–(3), such that $e_1(t) = x_{1\text{d}}(t) - x_1(t)$ globally asymptotically converges to zero when $t > 0$.

III. The Proposed Control Method

In Section III-A, we first design the baseline SBC by assuming the plant parameters θ_{kj} in (1)–(4) known $\forall k$, $\forall j$. Then, Section III-B proposes a projection function $\mathscr{P}_k$ for parametric uncertainties, such that SBC can be updated to the proposed *adaptive* SBC in Section III-C. The control design philosophy behind the proposed method is analyzed later in Section IV.

A. Subsystem-Based Control

Let the system parameters θ_{kj} in (1)–(4) be known $\forall k, \forall j$. The SBC for the SFF system in (1)–(3) can be designed as

$$\begin{cases} g_1(x_1)x_{2\text{d}} = \theta_{11}\dot{x}_{1\text{d}} - f_1(x_1) + \lambda_1 e_1 \\ \qquad\qquad = \mathbf{Y}_1\boldsymbol{\theta}_1 + \lambda_1 e_1 & (6)\\ g_i(\boldsymbol{x}_i)x_{(i+1)\text{d}} = \theta_{i1}\dot{x}_{i\text{d}} - f_i(\boldsymbol{x}_i) + \delta_{i-1}g_{i-1}(\boldsymbol{x}_{i-1})e_{i-1} + \lambda_i e_i \\ \qquad\qquad = \mathbf{Y}_i\boldsymbol{\theta}_i + \delta_{i-1}g_{i-1}(\boldsymbol{x}_{i-1})e_{i-1} + \lambda_i e_i & (7)\\ g_n(\boldsymbol{x}_n)u = \theta_{n1}\dot{x}_{n\text{d}} - f_n(\boldsymbol{x}_n) + \delta_{n-1}g_{n-1}(\boldsymbol{x}_{n-1})e_{n-1} + \lambda_n e_n \\ \qquad\qquad = \mathbf{Y}_n\boldsymbol{\theta}_n + \delta_{n-1}g_{n-1}(\boldsymbol{x}_{n-1})e_{n-1} + \lambda_n e_n & (8)\end{cases}$$

where $\lambda_k e_k = \lambda_k(x_{k\text{d}} - x_k)$ is the *local feedback* (FB) *term* with $\lambda_k > 0$; $\delta_{k-1}g_{k-1}(\boldsymbol{x}_{k-1})e_{k-1}$ is the *stabilizing FB term* for the previous subsystem with $\delta_{k-1} > 0$; $f_k(\boldsymbol{x}_k)$ is defined in (4); and in the model-based FF compensation term $\mathbf{Y}_k\boldsymbol{\theta}_k$, the regressor $\mathbf{Y}_k$ and the parameter vector $\boldsymbol{\theta}_k$ are defined as

$$\mathbf{Y}_k := \left[\dot{x}_{k\text{d}},\ -\gamma_{k2}(\boldsymbol{x}_k),\ -\gamma_{k3}(\boldsymbol{x}_k), \cdots,\ -\gamma_{kj}(\boldsymbol{x}_k)\right] \in \mathbb{R}^{1\times j} \quad (9)$$

$$\boldsymbol{\theta}_k := \left[\theta_{k1}, \theta_{k2}, \theta_{k3}, \cdots, \theta_{kj}\right]^T \in \mathbb{R}^j. \quad (10)$$

The control design in (6)–(8) is modular. This can be shown similarly as addressed later in Remark 3.2.

B. The Proposed Smooth Projection Function

As Fig. 1 shows, a parameter estimate $\widehat{\boldsymbol{\theta}}_k$ (that is an argument of $x_{(k+1)\text{d}}$) should be a function in C^{n-k} when stepping through the remaining time derivatives $\frac{d}{dt}$ in the chain. This can be guaranteed with the projection function in Def. 3.1.

Definition 3.1: Let $k \in \{1,...,n\}$, $a,b,c > 0$ satisfying $c+b > b \geqslant a > a-c > 0$, $\text{p} \in C^{n-k}$, and $\sigma, \rho > 0$ be fixed. Then, $\mathscr{P}_k(t)$ is defined to be the solution of

$$\dot{\mathscr{P}}_k(t) = \rho\left[\text{p}(t) + \sigma\kappa(\mathscr{P}_k)\right], \qquad \mathscr{P}_k(0) = \mathscr{P}_{k0} \quad (11)$$

for some $\mathscr{P}_{k0} \in \mathbb{R}$. Here κ has the form

$$\kappa(\mathscr{P}_k) = \begin{cases} b - \mathscr{P}_k & \text{if} \quad \mathscr{P}_k \geqslant b + c \\ (b - \mathscr{P}_k)S_b(\mathscr{P}_k) & \text{if} \quad b < \mathscr{P}_k < b + c \\ 0 & \text{if} \quad a \leqslant \mathscr{P}_k \leqslant b \\ (a - \mathscr{P}_k)S_a(\mathscr{P}_k) & \text{if} \quad a - c < \mathscr{P}_k < a \\ a - \mathscr{P}_k & \text{if} \quad \mathscr{P}_k \leqslant a - c \end{cases}$$

and the strictly decreasing $S_a : (a-c, a) \to (0,1)$ and strictly increasing $S_b : (b, b+c) \to (0,1)$ are such that $\kappa \in C^{n-k}$.

Appendix A provides formulas for the switching functions S_a and S_b, and an analysis of the projection function $\mathscr{P}_k$.

Remark 3.1: We note that when $n-k \leqslant 1$ holds for SS_k, $\mathscr{P} \in C^1$ in [3], [22] can be used instead of $\mathscr{P}_k$ satisfying (11). When $n-k \leqslant 2$ holds, $\mathscr{P}_2 \in C^2$ in [3], [22] can be used.

C. Adaptive Subsystem-Based Control

Let the system in (1)–(4) be subject to parametric uncertainties, i.e., θ_{kj} is unknown $\forall k, \forall j$. The control in Section III-A can be updated to the proposed *adaptive* SBC as

$$\begin{cases} g_1(x_1)x_{2\text{d}} = \mathbf{Y}_1\widehat{\boldsymbol{\theta}}_1 + \lambda_1 e_1 & (12)\\ g_i(\boldsymbol{x}_i)x_{(i+1)\text{d}} = \mathbf{Y}_i\widehat{\boldsymbol{\theta}}_i + \delta_{i-1}g_{i-1}(\boldsymbol{x}_{i-1})e_{i-1} + \lambda_i e_i & (13)\\ g_n(\boldsymbol{x}_n)u = \mathbf{Y}_n\widehat{\boldsymbol{\theta}}_n + \delta_{n-1}g_{n-1}(\boldsymbol{x}_{n-1})e_{n-1} + \lambda_n e_n & (14)\end{cases}$$

where $\mathbf{Y}_k\widehat{\boldsymbol{\theta}}_k$ is the adaptive model-based FF compensation, $\mathbf{Y}_k$ is defined in (9) and $\widehat{\boldsymbol{\theta}}_k \in \mathbb{R}^j$ is an estimate of $\boldsymbol{\theta}_k$ in (10). The estimated parameters in $\widehat{\boldsymbol{\theta}}_k$ need to be updated. We define

$$\mathbf{p}_k := e_k\mathbf{Y}_k^T \quad (15)$$

such that the ζth element of $\widehat{\boldsymbol{\theta}}_k$ in (12)–(14) can be updated by using the projection function $\mathscr{P}_k$ in Definition 3.1 as

$$\widehat{\theta}_{k\zeta} = \mathscr{P}_k(t), \qquad \forall \zeta \in \{1,...,j\} \quad (16)$$

with parameters $\text{p}_{k\zeta}, \rho_{k\zeta}, \sigma_{k\zeta}, \text{a} = \underline{\theta}_{k\zeta}, \text{b} = \overline{\theta}_{k\zeta}$ and $c_{k\zeta}$, where $\widehat{\theta}_{k\zeta}$ is the ζth element of $\widehat{\boldsymbol{\theta}}_k$; $\text{p}_{k\zeta}$ is the ζth element of $\mathbf{p}_k$ in

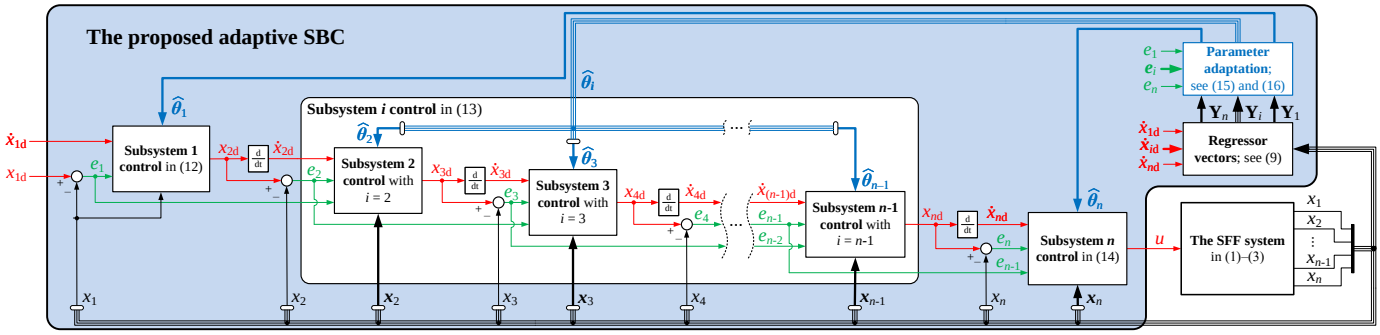

Fig. 1. Diagram of the proposed adaptive SBC (highlighted in light blue). The desired variables (and the control output u) are shown in red, the feedback signals are in green, the adaptive control is in blue, and the system output states are in black. The bold lines are vectors and the thin lines are scalar variables.

(15); $\rho_{k\zeta} > 0$ and $\sigma_{k\zeta} > 0$ are the parameter update gains; $\underline{\theta}_{k\zeta}$ and $\overline{\theta}_{k\zeta}$ are the lower and the upper bounds of $\theta_{k\zeta}$; and $c_{k\zeta}$ defines the activation interval beyond the bounds.

Fig. 1 shows the diagram of the proposed method.

Remark 3.2: Similarly to SS_k dynamics in (5), SS_k control in (12)–(14) can be reproduced with a single modular-structured equation $g_k(\boldsymbol{x}_k)x_{(k+1)\text{d}} = \mathbf{Y}_k\widehat{\boldsymbol{\theta}}_k + \delta_{k-1}g_{k-1}(\boldsymbol{x}_{k-1})e_{k-1} + \lambda_k e_k$ such that $\delta_0 g_0(\boldsymbol{x}_0)e_0 = 0$ and $x_{(n+1)\text{d}} = u$. This modularity allows that changing SS_k dynamics or adding/removing SSs, do not alter the control laws in the remaining SSs.

IV. Stability Analysis

Next, we provide an in-depth analysis on the adaptive SBC in Section III-C. Respective analysis can be performed for the SBC in Section III-A using $\boldsymbol{\theta}_k - \boldsymbol{\theta}_k = 0$ instead of $\boldsymbol{\theta}_k - \widehat{\boldsymbol{\theta}}_k$.

Motivated by a key concept in virtual stability analysis—a *virtual power flow* [3, Sect. 2.9.2]—we introduce a related notion of a *stability connector* as follows:

Definition 4.1: For the system (1)–(3) with the control (12)–(14), the *stability connector* s_k is defined as

$$s_k = \Delta_k g_k(t,\boldsymbol{x}_k)e_k e_{k+1}$$

where SS-related term $\Delta_k = 1$, if $k = 1$, and $\Delta_k = \frac{1}{\delta_1 \cdots \delta_{k-1}}$, if $k > 1$, and $\delta_1, \delta_2, \cdots, \delta_{k-1} > 0$ are feedback gains from Section III.

Next, in Lemmas 4.1–4.3 we provide auxiliary results for the convergence analysis in Theorem 4.1. Motivated by the concept of *virtual stability* [3, Sect. 2.9], the auxiliary analysis is carried out for the individual subsystem error dynamics e_k and the corresponding parameter estimation errors $\boldsymbol{\theta}_k - \widehat{\boldsymbol{\theta}}_k$.

Subtracting (1) from (12), adding $\theta_{11}\dot{x}_{1\text{d}} - \theta_{11}\dot{x}_{1\text{d}} = 0$, using (4), (9) and (10), and rearranging the terms, we get the following error dynamics for SS_1

$$\theta_{11}\dot{e}_1 = -\lambda_1 e_1 + g_1(x_1)e_2 + \mathbf{Y}_1(\boldsymbol{\theta}_1 - \widehat{\boldsymbol{\theta}}_1). \tag{17}$$

Lemma 4.1: Considering SS_1 error dynamics in (17), and $\boldsymbol{\theta}_1 - \widehat{\boldsymbol{\theta}}_1$ governed by (10), (15) and (16), the derivative of the quadratic function

$$v_1 = \frac{1}{2}\left(\theta_{11}e_1^2 + \sum_{\zeta=1}^{j}\frac{(\theta_{1\zeta} - \widehat{\theta}_{1\zeta})^2}{\rho_{1\zeta}}\right) \tag{18}$$

along the trajectories of the error dynamics satisfies

$$\dot{v}_1 \leqslant -\lambda_1 e_1^2 + s_1 \tag{19}$$

where s_1 is the stability connector from Def. 4.1.

Proof: See Appendix B. ∎

Remark 4.1: In Lemma 4.1, term e_2 in (17) is treated as an external input that causes s_1 to appear in (19) (see Appendix B) that will be canceled out based on the result of the next lemma. The dynamics of e_2 as well as the subsequent subsystems error dynamics are accounted for in the next two lemmas.

Subtracting (2) from (13), adding $\theta_{i1}\dot{x}_{i\text{d}} - \theta_{i1}\dot{x}_{i\text{d}} = 0$ using (4), (9) and (10), and rearranging the terms, we get the following error dynamics for SS_i, $\forall i \in \{2, \ldots, n-1\}$,

$$\begin{aligned}\theta_{i1}\dot{e}_i = &-\lambda_i e_i - \delta_{i-1}g_{i-1}(\boldsymbol{x}_{i-1})e_{i-1} + g_i(\boldsymbol{x}_i)e_{i+1}\\ &+ \mathbf{Y}_i(\boldsymbol{\theta}_i - \widehat{\boldsymbol{\theta}}_i).\end{aligned} \tag{20}$$

Lemma 4.2: Considering SS_i error dynamics in (20), and $\boldsymbol{\theta}_i - \widehat{\boldsymbol{\theta}}_i$ governed by (10), (15) and (16), the derivative of the quadratic function

$$v_i = \frac{1}{2(\delta_1 \cdots \delta_{i-1})}\left(\theta_{i1}e_i^2 + \sum_{\zeta=1}^{j}\frac{(\theta_{i\zeta} - \widehat{\theta}_{i\zeta})^2}{\rho_{i\zeta}}\right) \tag{21}$$

along the trajectories of the error dynamics satisfies

$$\dot{v}_i \leqslant -\frac{\lambda_i}{\delta_1 \cdots \delta_{i-1}}e_i^2 - s_{i-1} + s_i \tag{22}$$

where s_{i-1} and s_i are the stability connectors from Def. 4.1.

Proof: See Appendix B. ∎

Remark 4.2: Similarly to Lemma 4.1, e_{i+1} in (20) is treated as an external input that causes s_i to appear in (22). The stabilizing FB term $\delta_{i-1}g_{i-1}(\boldsymbol{x}_{i-1})e_{i-1}$ in (20) creates another stability connector $-s_{i-1}$ to appear in (22) (see Appendix B) that will cancel out s_{i-1} from the previous SS. The last connector s_{n-1} will be canceled out based on the result of the next lemma, after which we are in the position to present the convergence result for the overall error dynamics.

Subtracting (3) from (14), adding $\theta_{n1}\dot{x}_{n\text{d}} - \theta_{n1}\dot{x}_{n\text{d}} = 0$ using (4), (9) and (10), and rearranging the terms, we get the following error dynamics for SS_n

$$\theta_{n1}\dot{e}_n = -\lambda_n e_n - \delta_{n-1}g_{n-1}(\boldsymbol{x}_{n-1})e_{n-1} + \mathbf{Y}_n(\boldsymbol{\theta}_n - \widehat{\boldsymbol{\theta}}_n). \tag{23}$$

Lemma 4.3: Considering SS_n error dynamics in (23), and $\boldsymbol{\theta}_n - \widehat{\boldsymbol{\theta}}_n$ governed by (10), (15) and (16), the derivative of the quadratic function

$$v_n = \frac{1}{2(\delta_1 \cdots \delta_{n-1})}\left(\theta_{n1}e_n^2 + \sum_{\zeta=1}^{j}\frac{(\theta_{n\zeta} - \widehat{\theta}_{n\zeta})^2}{\rho_{n\zeta}}\right) \tag{24}$$

along the trajectories of the error dynamics satisfies

$$\dot{v}_n \leqslant -\frac{\lambda_n}{\delta_1 \cdots \delta_{n-1}} e_n^2 - s_{n-1} \tag{25}$$

where s_{n-1} is the stability connector from Def. 4.1.

Proof: See Appendix B. ∎

We will now construct a Lyapunov candidate for the overall error dynamics as the sum of the quadratic functions from Lemmas 4.1–4.3. Based on the properties derived in the lemmas, we obtain that the error dynamics will remain bounded, and moreover, that the control errors converge globally asymptotically to zero. The result is given in the following theorem.

Theorem 4.1: Consider the error dynamics $\boldsymbol{e} = [e_1, \ldots, e_n]^T$ and the parameter estimation error $\boldsymbol{\theta}_k - \widehat{\boldsymbol{\theta}}_k$, $\forall k \in \{1, 2, \ldots, n\}$, that are governed in Lemmas 4.1–4.3. For arbitrary initial conditions, $\boldsymbol{\theta}_k - \widehat{\boldsymbol{\theta}}_k$ remains bounded and $e_k(t) \to 0$ globally as $t \to \infty$ for all $k \in \{1, 2, \ldots, n\}$.

Proof: Using (18), (21) and (24), we choose a Lyapunov candidate function for the overall error dynamics as

$$\begin{aligned} v_{\text{tot}} &= v_1 + \sum_{i=2}^{n-1} v_i + v_n \\ &= \frac{1}{2}\boldsymbol{e}^T \mathbf{A} \boldsymbol{e} + \sum_{k=1}^{n} \frac{1}{2(\delta_1 \cdots \delta_{k-1})} \sum_{\zeta=1}^{j} \frac{(\theta_{k\zeta} - \widehat{\theta}_{k\zeta})^2}{\rho_{k\zeta}} \end{aligned}$$

where $\mathbf{A} = diag\left(\theta_{11}, \frac{\theta_{21}}{\delta_1}, \frac{\theta_{31}}{\delta_1 \delta_2}, \cdots, \frac{\theta_{n1}}{\delta_1 \cdots \delta_{n-1}}\right) \in \mathbb{R}^{n \times n}$ is positive definite. Then, it follows from (19), (22) and (25) that

$$\begin{aligned} \dot{v}_{\text{tot}} &= \dot{v}_1 + \sum_{i=2}^{n-1} \dot{v}_i + \dot{v}_n \\ &\leqslant -\lambda_1 e_1^2 + s_1 - \sum_{i=2}^{n-1} \left[\frac{\lambda_i}{\delta_1 \cdots \delta_{i-1}} e_i^2 - s_{i-1} + s_i \right] \\ &\quad - \frac{\lambda_n}{\delta_1 \cdots \delta_{n-1}} e_n^2 - s_{n-1} \\ &= -\lambda_1 e_1^2 - \sum_{i=2}^{n-1} \frac{\lambda_1}{\delta_1 \cdots \delta_{i-1}} e_i^2 - \frac{\lambda_n}{\delta_1 \cdots \delta_{n-1}} e_n^2 + \sum_{k=1}^{n-1} (s_k - s_k) \\ &= -\boldsymbol{e}^T \mathbf{B} \boldsymbol{e} \end{aligned}$$

where $\mathbf{B} = diag\left(\lambda_1, \frac{\lambda_2}{\delta_1}, \frac{\lambda_3}{\delta_1 \delta_2}, \cdots, \frac{\lambda_n}{\delta_1 \cdots \delta_{n-1}}\right) \in \mathbb{R}^{n \times n}$ is positive definite and every stability connector s_k is canceled by its negative counterpart $-s_k$, $\forall k \in \{1, 2, ..., n-1\}$. By [23, Thm. 8.4] both the control errors and the parameter estimation errors are bounded, and $\boldsymbol{e}(t)^T \mathbf{B} \boldsymbol{e}(t) \to 0$ globally as $t \to \infty$, which by the positive-definiteness of $\mathbf{B}$ is equivalent to $\boldsymbol{e}(t) \to \mathbf{0}$ as $t \to \infty$, i.e., $e_k(t) \to 0$, $\forall k \in \{1, 2, \ldots, n\}$ as $t \to \infty$. ∎

Finally, motivated by the original concept of *virtual stability* [3, Sect. 2.9], Def. 4.2 generalizes the results in Lemmas 4.1–4.3 for generic tools of *virtual stability* to design SS_k control.

Definition 4.2: The kth subsystem, $\forall k \in \{1, ..., n\}$, in (1)–(3), combined with its respective control in (12)–(16), is said to be *virtually stable* if the derivative of a quadratic function $v_k = \alpha_k e_k^2 + (\boldsymbol{\theta}_k - \widehat{\boldsymbol{\theta}}_k)^T \boldsymbol{\Gamma}_k (\boldsymbol{\theta}_k - \widehat{\boldsymbol{\theta}}_k)$ along the trajectories of the error dynamics satisfies $\dot{v}_k \leqslant -\beta_k e_k^2 - s_{k-1} + s_k$ for some $\alpha_k, \beta_k > 0$ and positive-definite $\boldsymbol{\Gamma}_k \in \mathbb{R}^{k \times k}$, where s_{k-1} and s_k are the stability connectors by Def. 4.1 such that $s_0 = 0$ and $s_n = 0$.

We note that alternative control laws, e.g., for the parameter adaptation, can be incorporated/designed for Def. 4.2, as long as the stability connectors s_{k-1} and s_k exist and the stability can be guaranteed as in the proof of Theorem 4.1. This is similarly as we have demonstrated for local control design modifications in [24], where a local observer design was incorporated to VDC using VPFs; see Appendix C for VDC and VPFs.

Remark 4.3: Similarly to *adaptive backstepping* [5], $x_{(k+1)\text{d}}$ in (12) and (13) acts as a fictitious control from SS_k to SS_{k+1}, $\forall k \in \{1, ..., n-1\}$, while the actual control effort u is eventually obtained in (14) after recursively stepping through every SS. However, instead of designing the overall control u that eventually stabilizes the entire system "as a whole", we proposed generic and *modular* tools (see Remark 3.2, Def. 4.1, Thm. 4.1 and Def. 4.2) that **1**) automatically stabilize the adjacent SSs (for global asymptotic stability), and **2**) prevent an excessive growth of the control design complexity. The method also allows that **3**) SS_k control can be modified (without affecting the control laws in the remaining SSs), as long as s_{k-1} and s_k in Def. 4.2 exist for the adjacent SSs and stability can be guaranteed as shown in the proof of Theorem 4.1.

Remark 4.4: While the *dissipative adaptive control* in [21] designs strictly passive interaction dynamics for adjacent SSs, we designed in (12)–(14) the stabilizing FB term $\delta_{k-1} g_{k-1}(\boldsymbol{x}_{k-1}) e_{k-1}$, $\forall k \in \{2, ..., n\}$, to produce *stability connector* s_{k-1}. This provides that every SS is automatically stabilized by its adjacent SS such that passivity between SSs does not need to be considered. We based the results on Lyapunov functions instead of Lebesque L_2/L_∞ integrable functions.

Comparisons to VDC [3] can be found in Appendix C.

V. Numerical Validation

The previous methods [6]–[14] demonstrated the results without the "explosion of complexity" with a 3rd-order SFF plant. For this purpose, we use the following 5th-order SFF plant (the plant from [8], [9] updated with SS_3 and SS_4)

$$\begin{cases} a_1 \dot{x}_1 = b_1 x_1^3 + x_2 \\ a_2 \dot{x}_2 = b_2 (x_1^2 + x_2^2) + x_3 \\ a_3 \dot{x}_3 = b_3 \sin^3(x_1) \sin(x_2) x_3 - x_4 \\ a_4 \dot{x}_4 = b_4 (x_1^2 x_4 - x_2^2 x_3) - [(\cos^4(x_1) \cos^2(x_2) + 2] x_5 \\ a_5 \dot{x}_5 = u. \end{cases} \tag{26}$$

Using (12)–(14), we design the control for the plant in (26) as

$$\begin{cases} g_1(x_1) x_{2\text{d}} = \mathbf{Y}_1 \widehat{\boldsymbol{\theta}}_1 + \lambda_1 e_1 \\ g_2(\boldsymbol{x}_2) x_{3\text{d}} = \mathbf{Y}_2 \widehat{\boldsymbol{\theta}}_2 + \lambda_2 e_2 + \delta_1 g_1(x_1) e_1 \\ g_3(\boldsymbol{x}_3) x_{4\text{d}} = \mathbf{Y}_3 \widehat{\boldsymbol{\theta}}_3 + \lambda_3 e_3 + \delta_2 g_2(\boldsymbol{x}_2) e_2 \\ g_4(\boldsymbol{x}_4) x_{5\text{d}} = \mathbf{Y}_4 \widehat{\boldsymbol{\theta}}_4 + \lambda_4 e_4 + \delta_3 g_3(\boldsymbol{x}_3) e_3 \\ g_5(\boldsymbol{x}_5) u = \mathrm{Y}_5 \widehat{\theta}_5 + \lambda_5 e_5 + \delta_4 g_4(\boldsymbol{x}_4) e_4 \end{cases} \tag{27}$$

where $g_1(x_1) = g_2(\boldsymbol{x}_2) = g_5(\boldsymbol{x}_5) = 1$, $g_3(\boldsymbol{x}_3) = -1$, $g_4(\boldsymbol{x}_4) = -(\cos^4(x_1)\cos^2(x_2) + 2)$; $\mathbf{Y}_1 = [\dot{x}_{1\text{d}} \;\; -x_1^3]$, $\mathbf{Y}_2 = [\dot{x}_{2\text{d}} \;\; -(x_1^2 + x_2^2)]$, $\mathbf{Y}_3 = [\dot{x}_{3\text{d}} \;\; -\sin^3(x_1)\sin(x_2)x_3]$, $\mathbf{Y}_4 = [\dot{x}_{4\text{d}} \;\; -(x_1^2 x_4 - x_2^2 x_3)]$, $\mathrm{Y}_5 = \dot{x}_{5\text{d}}$; and ideally $\boldsymbol{\theta}_1 = [\theta_{11} \;\; \theta_{12}]^T = [a_1 \;\; b_1]^T$, $\boldsymbol{\theta}_2 = [\theta_{21} \;\; \theta_{22}]^T = [a_2 \;\; b_2]^T$, $\boldsymbol{\theta}_3 = [\theta_{31} \;\; \theta_{32}]^T = [a_3 \;\; b_3]^T$, $\boldsymbol{\theta}_4 = [\theta_{41} \;\; \theta_{42}]^T = [a_4 \;\; b_4]^T$, $\theta_5 = \theta_{51} = a_5$ hold for the parameters.

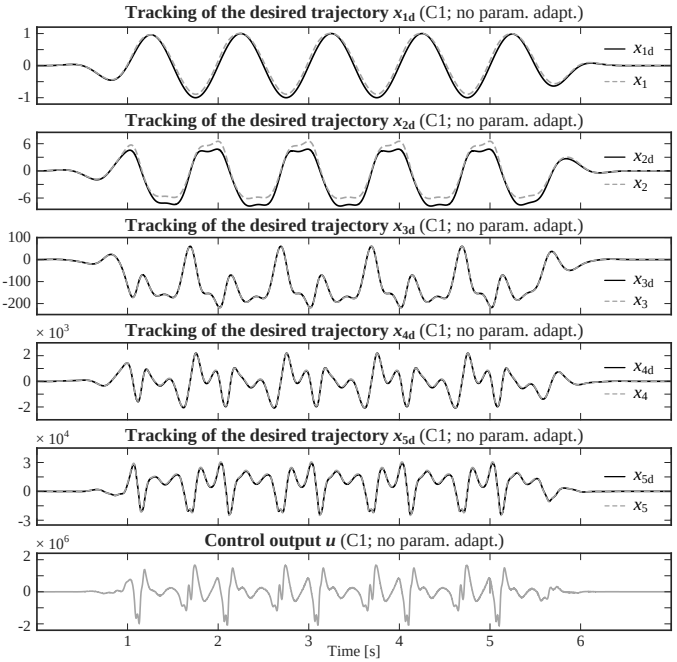

Fig. 2. Control performance in C1 with inaccurate constant parameters (no parameter adaptation). The desired trajectories are shown in black and their controlled variables in gray (plots 1–5). The last plot shows the output signal u.

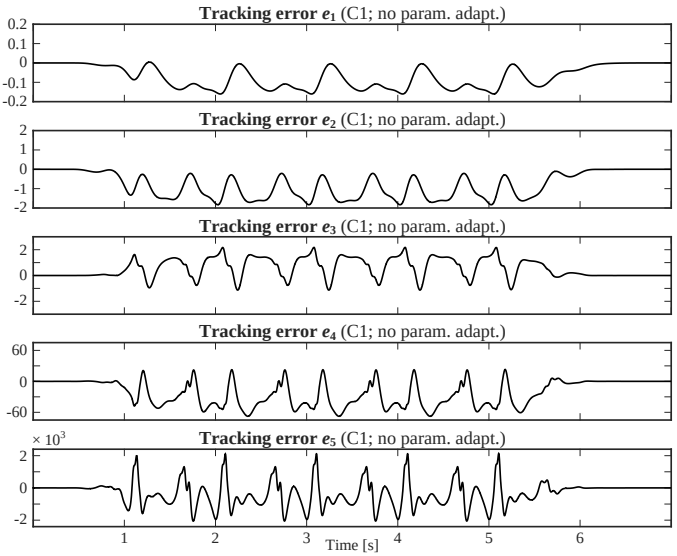

Fig. 3. Tracking errors e_k, $\forall k \in \{1,...,5\}$, in C1 (no parameter adaptation).

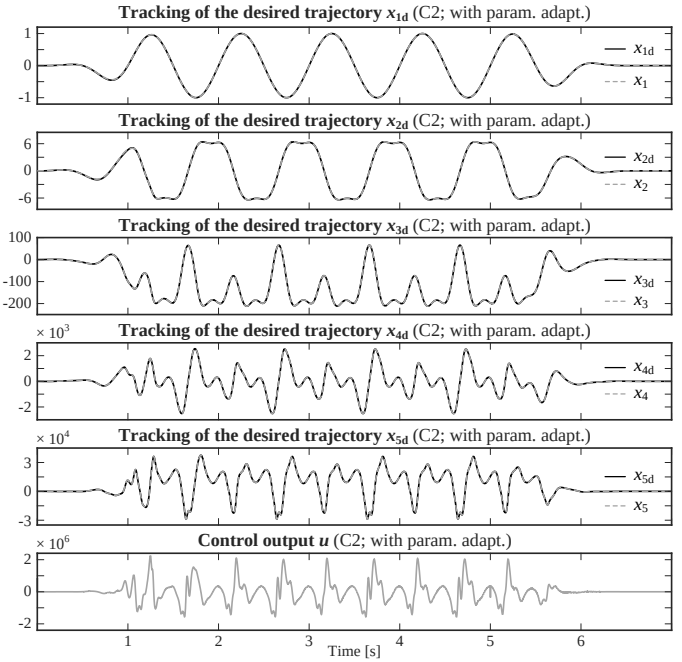

Fig. 4. Control performance in C2 with adaptive control. The desired trajectories are shown in black and their controlled variables in gray (plots 1–5). The last plot shows the output signal u.

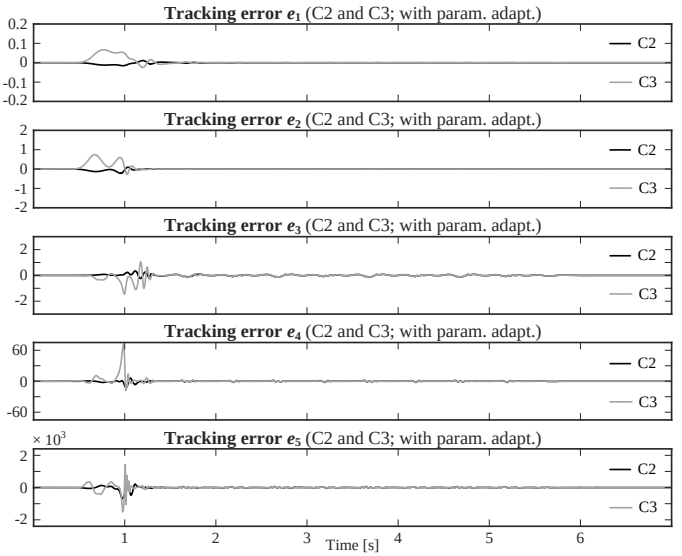

Fig. 5. Tracking errors e_k, $\forall k \in \{1,...,5\}$, in C2 and C3 (adaptive control).

To study the global asymptotic convergence suggested by Theorem 4.1, the following sufficiently smooth reference trajectory $x_{1d}(t)$ is used

$$x_{1d}(t) = \begin{cases} \sin(2\pi t)\tanh(t^3), & \text{if } 0 \leqslant t \leqslant 5 \\ \sin(2\pi t)\tanh(t^3)[1-\tanh((t-5)^3)], & \text{if } t > 5. \end{cases}$$

For simplicity, $a_1, a_2, a_3, a_4 = 1$ are used in (26) and considered known for the control in (27). The plant parameters b_1,b_2,b_3,b_4 = 5 and a_5 = 2 are considered unknown for the experiments with adaptive control in (27). The FB gains were loosely tuned to λ_1 = 10, λ_2 = 20, λ_3 = 40, λ_4 = 80, λ_5 = 160, δ_1 = 10, δ_2 = 10, δ_3 = 100 and δ_4 = 1000. A fixed-step discrete solver with the sample time of 0.1 ms was used in the simulations. The following three test cases are studied:

C1: The baseline SBC (in Sec. III-A) is employed in (27) with inaccurate constant FF parameters θ_{12},θ_{32} = 6, θ_{22},θ_{42} = 4 and θ_{51} = 2.5 (while the respective plant parameters are b_1,b_2,b_3,b_4 = 5 and a_5 = 2). C1 is used as a baseline for comparisons to the proposed adaptive SBC later in C2 and C3. Figs. 2 and 3 show C1 results.

C2: The adaptive SBC (in Sec. III-C) is enabled in (27) with initial parameter estimates $\widehat{\theta}_{12}(0)$,$\widehat{\theta}_{32}(0)$ = 6, $\widehat{\theta}_{22}(0)$, $\widehat{\theta}_{42}(0)$ = 4 and $\widehat{\theta}_{51}(0)$ = 2.5. Figs. 4–6 show C2 results.

C3: The adaptive SBC (in Sec. III-C) is enabled in (27) with initial parameter estimates $\widehat{\theta}_{12}(0)$,$\widehat{\theta}_{32}(0)$ = 0.1, $\widehat{\theta}_{22}(0)$, $\widehat{\theta}_{42}(0)$ = 9.9 and $\widehat{\theta}_{51}(0)$ = 0.5 that are outside the projection function $\mathscr{P}_k$ bounds. Figs. 5 and 6 show C3 results.

In C2 and C3, the parameter update gains were set to $\rho_{12} = 2000$, $\rho_{22} = 4$, $\rho_{32} = 2$, $\rho_{42} = 0.001$, $\rho_{51} = 3e^{-8}$, $\sigma_{k2} = 1000/\rho_{k2}$, $\forall k \in \{1,...,4\}$, and $\sigma_{51} = 1000/\rho_{51}$; the parameter bounds were set to $\overline{\theta}_{12},\overline{\theta}_{22},\overline{\theta}_{32},\overline{\theta}_{42} = 9$, $\overline{\theta}_{51} = 3$ and $\underline{\theta}_{12},\underline{\theta}_{22},\underline{\theta}_{32},\underline{\theta}_{42},\underline{\theta}_{51} = 1$; and the activation intervals beyond the bounds were set to $c_{12},c_{22},c_{32},c_{42} = 0.5$ and $c_{51} = 0.25$.

Figs. 2 and 3 show the results in C1 with inaccurate FF parameters (i.e., $\theta_{k2} \neq b_k$, $\forall k \in \{1,...,4\}$, and $\theta_{51} \neq a_5$). In Fig. 2, plots 1–5 shows the desired trajectory x_{kd}, $\forall k \in \{1,...,5\}$, in black and its controlled state x_k in gray. The last plot shows the control output u. Fig. 3 shows the detailed tracking errors. The maximum absolute errors are $|e_1|_{\max} = 0.160$, $|e_2|_{\max} = 1.831$, $|e_3|_{\max} = 2.170$, $|e_4|_{\max} = 67.67$ and $|e_5|_{\max} = 2136$. Due to the parametric uncertainty, noticeable tracking errors occur in the transition phases.

Figs. 4–6 show the main results of the study with the proposed adaptive SBC in (27). Fig. 4 shows the tracking results in C2 where the initial parameter estimate values are defined in accordance to C1, i.e., $\widehat{\theta}_{12}(0),\widehat{\theta}_{32}(0) = 6$, $\widehat{\theta}_{22}(0),\widehat{\theta}_{42}(0) = 4$ and $\widehat{\theta}_{51}(0) = 2.5$. As the black lines in Fig. 5 show, the tracking errors are significantly decreased in relation to C1, with the max. absolute errors $|e_1|_{\max} = 0.015$, $|e_2|_{\max} = 0.218$, $|e_3|_{\max} = 0.341$, $|e_4|_{\max} = 12.36$ and $|e_5|_{\max} = 684.6$. As stated by the theory, global asymptotic convergence is achieved. Fig. 6 shows the parameter estimates $\widehat{\theta}_{12}$, $\widehat{\theta}_{22}$, $\widehat{\theta}_{32}$, $\widehat{\theta}_{42}$ and $\widehat{\theta}_{51}$ in black, showing that the projection function $\mathscr{P}_k$ actively pushes the parameter values toward their real plant value.

Finally, in C3, the initial parameter values are set outside the projection function $\mathscr{P}_k$ bounds such that $\widehat{\theta}_{12}(0),\widehat{\theta}_{32}(0) = 0.1$, $\widehat{\theta}_{22}(0),\widehat{\theta}_{42}(0) = 9.9$, $\widehat{\theta}_{51}(0) = 0.5$. Figs. 5 and 6 show the results in gray. Despite a significant inaccuracy in the initial parameter values, the projection function $\mathscr{P}_k$ actively pushes the parameter values toward their real plant value (see Fig. 6). The max. absolute tracking errors are $|e_1|_{\max} = 0.067$, $|e_2|_{\max} = 0.744$, $|e_3|_{\max} = 1.452$, $|e_4|_{\max} = 71.73$ and $|e_5|_{\max} = 1494$ (see Fig. 5). After 1.5 s, the control behavior in C3 becomes virtually identical to C2.

VI. Conclusions

This study proposed theoretical foundations for an *adaptive subsystem-based control* for controlling nth-order SFF systems with parametric uncertainties. As an alternative for backstepping, we provided systematic and straightforward tools for globally asymptotically stable control while avoiding a growth of the control design complexity when the system order n increases. The proposed method is modular in the sense that the control for every SS can be designed with a single generic-form equation such that changing SS dynamics or removing/adding SSs do not affect to the control laws in the remaining SSs. For the method, we reformulated the original concept of *virtual stability* in [3, Def. 2.17] and proposed a specific *stability connector* to address dynamic interactions between the adjacent SSs. These features enable that both the control design and its stability analysis can be performed locally at a SS level (as opposed to the whole system). We proposed also a smooth projection function $\mathscr{P}_k$ for the system parametric uncertainties. Theoretical developments on global asymptotic convergence (in Theorem 4.1) were verified in numerical simulations with a nonlinear 5th-order SFF system. We left topics, e.g., a signal noise attenuation and semi-SFF systems with unknown dynamics, for a subject of future studies.

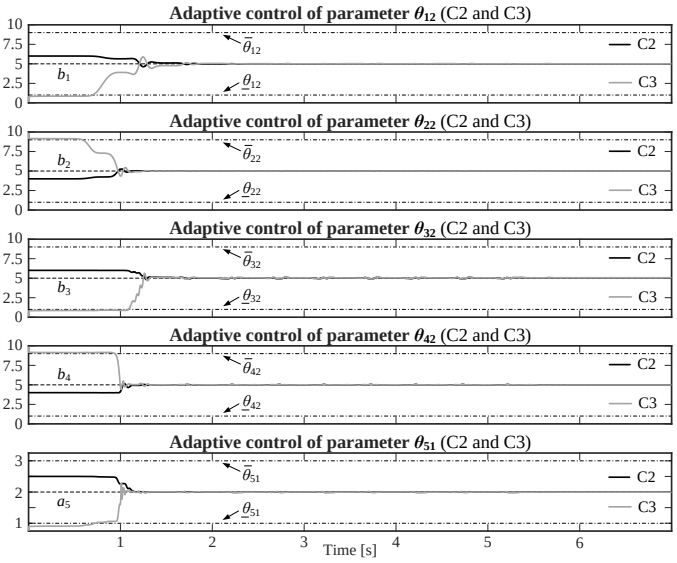

Fig. 6. Adapted parameters $\widehat{\theta}_{12}(t)$, $\widehat{\theta}_{22}(t)$, $\widehat{\theta}_{32}(t)$, $\widehat{\theta}_{42}(t)$ and $\widehat{\theta}_{51}(t)$. The results in C2 are given in black, while the result in C3 are given in gray. The true plant parameters $b_1,b_2,b_3,b_4 = 5$ and $a_5 = 2$ are shown in dashed line. The upper bounds ($\overline{\theta}$) and lower bounds ($\underline{\theta}$) are shown in dashed-dot line.

Appendix A
The Projection Function $\mathscr{P}_k$

In Def. 3.1, parameters a and b define the lower and upper bounds for $\mathscr{P}_k$ such that $b \geqslant \mathscr{P}_c \geqslant a > 0$, where $\mathscr{P}_c$ is the actual constant parameter value for $\mathscr{P}_k$. Within the bounds, $\dot{\mathscr{P}}_k$ is driven by $\rho \mathrm{p}(t)$ and the behavior of $\mathscr{P}_k$ is equal to $\mathscr{P} \in C^1$ and $\mathscr{P}_2 \in C^2$ in [3], [22]. Outside the bounds, a corrective term $\sigma\kappa(\mathscr{P}_k)$ is designed to bring $\mathscr{P}_k$ back toward the bounds. The parameter c defines activation intervals $(a-c,a)$ and $(b,b+c)$ for the switching functions S_a and S_b. Fig. 7 shows how the projection function in Def. 3.1 is driven within the segments defined by a, b and c, and illustrates the designed $\kappa(\mathscr{P}_k)$.

Let $k \in \{1,...,n\}$ be fixed and assume $|\mathrm{p}^{(n-k)}(t)| < +\infty$. To guarantee the existence of $\mathscr{P}_k^{(n-k)}$, i.e., $\kappa \in C^{n-k}$, the strictly decreasing $S_a \in C^{n-k}(a-c,a;(0,1))$ and strictly increasing $S_b \in C^{n-k}(b,b+c;(0,1))$ are required to satisfy:

$$\begin{gathered}\lim_{x\to(a-c)^+} S_a(x) = 1, \quad \lim_{x\to(a-c)^+} S_a^{(j)}(x) = 0, \\ \lim_{x\to a^-} S_a(x) = 0 \quad \text{and} \quad \lim_{x\to a^-} S_a^{(j)}(x) = 0,\end{gathered} \tag{28}$$

$$\begin{gathered}\lim_{x\to(b+c)^-} S_b(x) = 1, \quad \lim_{x\to(b+c)^-} S_b^{(j)}(x) = 0, \\ \lim_{x\to b^+} S_b(x) = 0 \quad \text{and} \quad \lim_{x\to b^+} S_b^{(j)}(x) = 0,\end{gathered} \tag{29}$$

$\forall j \in \{1,...,n-k\}$. These assumptions guarantee that κ in Def. 3.1 satisfies $\kappa \in C^{n-k}$ (see Fig. 7). Def. A.1 provides a strictly decreasing function $S_a \in C^{n-k}$ and a strictly increasing function $S_b \in C^{n-k}$, satisfying (28) and (29), respectively, for any $n-k$.

Definition A.1: $S_a : (a-c, a) \to (0,1)$ is a *smooth and strictly decreasing* switching function defined as

$$S_a(\mathscr{P}_k) := \frac{1}{2}\left[1 - \tanh\left(\frac{1}{a-c-\mathscr{P}_k} + \frac{1}{a-\mathscr{P}_k}\right)\right]$$

and $S_b : (b, b+c) \to (0,1)$ is a *smooth and strictly increasing* switching function defined as

$$S_b(\mathscr{P}_k) := \frac{1}{2}\left[1 + \tanh\left(\frac{1}{b-\mathscr{P}_k} + \frac{1}{b+c-\mathscr{P}_k}\right)\right].$$

The projection function in (11) has the following property.

Lemma A.1: For any constant $\mathscr{P}_c$ with $a \leqslant \mathscr{P}_c \leqslant b$ we have

$$(\mathscr{P}_c - \mathscr{P}_k)\left(\mathrm{p}(t) - \frac{1}{\rho}\dot{\mathscr{P}}_k\right) \leqslant -\sigma\kappa^2(\mathscr{P}_k) \leqslant 0, \ \forall \mathscr{P}_k \in \mathbb{R}. \tag{30}$$

Proof: The proof of Lemma A.1 follows a similar procedure as the proof of Lemma 2.10 in [3]. First, substituting (11) into (30) we get

$$(\mathscr{P}_c - \mathscr{P}_k)\left(\mathrm{p}(t) - \frac{1}{\rho}\dot{\mathscr{P}}_k\right) = -\sigma(\mathscr{P}_c - \mathscr{P}_k)\kappa(\mathscr{P}_k). \tag{31}$$

Below, we will show that $-\sigma(\mathscr{P}_c - \mathscr{P}_k)\kappa(\mathscr{P}_k) \leqslant -\sigma\kappa(\mathscr{P}_k)^2 \leqslant 0$ holds for all $\mathscr{P}_k \in \mathbb{R}$ in (31), eventually completing the proof of Lemma A.1.

Since $a \leqslant \mathscr{P}_c \leqslant b$, we have

$$\mathscr{P}_c - \mathscr{P}_k \geqslant a - \mathscr{P}_k \tag{32}$$

$$\mathscr{P}_k - \mathscr{P}_c \geqslant \mathscr{P}_k - b. \tag{33}$$

When $\mathscr{P}_k \leqslant a-c$, Def. 3.1 yields $\kappa(\mathscr{P}_k) = (a - \mathscr{P}_k) > 0$. Using (32), we get

$$\begin{aligned} -\sigma(\mathscr{P}_c - \mathscr{P}_k)\kappa(\mathscr{P}_k) &\leqslant -\sigma(a - \mathscr{P}_k)\kappa(\mathscr{P}_k) \\ &= -\sigma\kappa^2(\mathscr{P}_k) \leqslant 0. \end{aligned} \tag{34}$$

When $a-c < \mathscr{P}_k < a$, Def. 3.1 yields $\kappa(\mathscr{P}_k) = (a - \mathscr{P}_k)S_a(\mathscr{P}_k) > 0$ and $S_a(\mathscr{P}_k) \in (0,1)$. Using (32), we get

$$\begin{aligned} -\sigma(\mathscr{P}_c - \mathscr{P}_k)\kappa(\mathscr{P}_k) &\leqslant -\sigma(a - \mathscr{P}_k)\kappa(\mathscr{P}_k) \\ &< -\sigma(a - \mathscr{P}_k)S_a(\mathscr{P}_k)\kappa(\mathscr{P}_k) \\ &= -\sigma\kappa^2(\mathscr{P}_k) \leqslant 0. \end{aligned} \tag{35}$$

When $a \leqslant \mathscr{P}_k \leqslant b$, $\kappa(\mathscr{P}_k) = 0$ and we get

$$-\sigma(\mathscr{P}_c - \mathscr{P}_k)\kappa(\mathscr{P}_k) = 0. \tag{36}$$

When $b < \mathscr{P}_k < b+c$, Def. 3.1 yields $\kappa(\mathscr{P}_k) = (b - \mathscr{P}_k)S_b(\mathscr{P}_k) < 0$ and $S_b(\mathscr{P}_k) \in (0,1)$. Using (33), we get

$$\begin{aligned} -\sigma(\mathscr{P}_c - \mathscr{P}_k)\kappa(\mathscr{P}_k) &= \sigma(\mathscr{P}_k - \mathscr{P}_c)\kappa(\mathscr{P}_k) \\ &\leqslant \sigma(\mathscr{P}_k - b)\kappa(\mathscr{P}_k) \\ &< \sigma(\mathscr{P}_k - b)S_b(\mathscr{P}_k)\kappa(\mathscr{P}_k) \\ &= -\sigma(b - \mathscr{P}_k)S_b(\mathscr{P}_k)\kappa(\mathscr{P}_k) \\ &= -\sigma\kappa^2(\mathscr{P}_k) \leqslant 0. \end{aligned} \tag{37}$$

When $\mathscr{P}_k \geqslant b+c$, Def. 3.1 yields $\kappa(\mathscr{P}_k) = (b - \mathscr{P}_k) < 0$. Using (33), we get

$$\begin{aligned} -\sigma(\mathscr{P}_c - \mathscr{P}_k)\kappa(\mathscr{P}_k) &= \sigma(\mathscr{P}_k - \mathscr{P}_c)\kappa(\mathscr{P}_k) \\ &\leqslant \sigma(\mathscr{P}_k - b)\kappa(\mathscr{P}_k) \\ &= -\sigma(b - \mathscr{P}_k)\kappa(\mathscr{P}_k) \\ &= -\sigma\kappa^2(\mathscr{P}_k) \leqslant 0. \end{aligned} \tag{38}$$

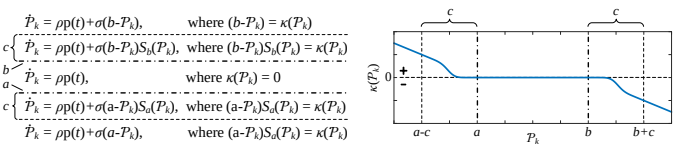

Fig. 7. Left: a diagram describing how the proposed projection function in Def. 3.1 is driven within the piecewise defined segments in $\kappa(\mathscr{P}_k)$. Right: a behavior of $\kappa(\mathscr{P}_k)$ in Def. 3.1 with the designed S_a and S_b in Def. A.1.

Finally, (34)–(38) together with (31) complete the proof. ■

APPENDIX B
PROOFS OF LEMMAS 4.1, 4.2, AND 4.3

Proof of Lemma 4.1: Let us first define the two properties in (39) and (40). It follows directly from Lemma A.1 that

$$\sum_{\zeta=1}^{j}(\theta_{1\zeta} - \widehat{\theta}_{1\zeta})\left(\mathrm{p}_{1\zeta} - \frac{\dot{\widehat{\theta}}_{1\zeta}}{\rho_{1\zeta}}\right) \leqslant 0. \tag{39}$$

In addition, Definition 4.1 yields

$$g_1(x_1)e_1e_2 = s_1. \tag{40}$$

Using (17), (39) and (40), the derivative of the quadratic function v_1 in (18) can be written as

$$\begin{aligned} \dot{v}_1 &= e_1\theta_{11}\dot{e}_1 - \sum_{\zeta=1}^{j}(\theta_{1\zeta} - \widehat{\theta}_{1\zeta})\frac{\dot{\widehat{\theta}}_{1\zeta}}{\rho_{1\zeta}} \\ &= -\lambda_1 e_1^2 + g_1(x_1)e_1e_2 + e_1\mathbf{Y}_1(\boldsymbol{\theta}_1 - \widehat{\boldsymbol{\theta}}_1) \\ &\quad - \sum_{\zeta=1}^{j}(\theta_{1\zeta} - \widehat{\theta}_{1\zeta})\frac{\dot{\widehat{\theta}}_{1\zeta}}{\rho_{1\zeta}} \\ &= -\lambda_1 e_1^2 + s_1 + \sum_{\zeta=1}^{j}(\theta_{1\zeta} - \widehat{\theta}_{1\zeta})\left(\mathrm{p}_{1\zeta} - \frac{\dot{\widehat{\theta}}_{1\zeta}}{\rho_{1\zeta}}\right) \\ &\leqslant -\lambda_1 e_1^2 + s_1 \end{aligned}$$

which completes the proof of Lemma 4.1. ■

Proof of Lemma 4.2: As in the proof of Lemma 4.1, using (20), Def. 4.1 and Lemma A.1, the derivative of the quadratic function v_i in (21), $\forall i \in \{2, \ldots, n-1\}$, can be written as

$$\begin{aligned} \dot{v}_i &= e_i\frac{\theta_{i1}}{\delta_1\cdots\delta_{i-1}}\dot{e}_i - \frac{1}{\delta_1\cdots\delta_{i-1}}\sum_{\zeta=1}^{j}(\theta_{i\zeta} - \widehat{\theta}_{i\zeta})\frac{\dot{\widehat{\theta}}_{i\zeta}}{\rho_{i\zeta}} \\ &= e_i\frac{1}{\delta_1\cdots\delta_{i-1}}\Big[g_i(\boldsymbol{x}_i)e_{i+1} - \delta_{i-1}g_{i-1}(\boldsymbol{x}_{i-1})e_{i-1} - \lambda_i e_i \\ &\quad + \mathbf{Y}_i(\boldsymbol{\theta}_i - \widehat{\boldsymbol{\theta}}_i)\Big] - \frac{1}{\delta_1\cdots\delta_{i-1}}\sum_{\zeta=1}^{j}(\theta_{i\zeta} - \widehat{\theta}_{i\zeta})\frac{\dot{\widehat{\theta}}_{i\zeta}}{\rho_{i\zeta}} \\ &= -\frac{\lambda_i}{\delta_1\cdots\delta_{i-1}}e_i^2 - \frac{1}{\delta_1\cdots\delta_{i-2}}g_{i-1}(\boldsymbol{x}_{i-1})e_{i-1}e_i \\ &\quad + \frac{1}{\delta_1\cdots\delta_{i-1}}g_i(\boldsymbol{x}_i)e_ie_{i+1} \\ &\quad + \frac{1}{\delta_1\cdots\delta_{i-1}}\sum_{\zeta=1}^{j}(\theta_{i\zeta} - \widehat{\theta}_{i\zeta})\left(\mathrm{p}_{i\zeta} - \frac{\dot{\widehat{\theta}}_{i\zeta}}{\rho_{i\zeta}}\right) \\ &\leqslant -\frac{\lambda_i}{\delta_1\cdots\delta_{i-1}}e_i^2 - s_{i-1} + s_i \end{aligned}$$

which completes the proof of Lemma 4.2. ■

Proof of Lemma 4.3: As in the proof of Lemma 4.1, using (23), Def. 4.1 and Lemma A.1, the derivative of the quadratic function v_n in (24) can be written as

$$\begin{aligned}\dot{v}_n &= e_n\frac{\theta_{n1}}{\delta_1\cdots\delta_{n-1}}\dot{e}_n - \frac{1}{\delta_1\cdots\delta_{n-1}}\sum_{\zeta=1}^{j}(\theta_{n\zeta}-\widehat{\theta}_{n\zeta})\frac{\dot{\widehat{\theta}}_{n\zeta}}{\rho_{n\zeta}}\\ &= e_n\frac{1}{\delta_1\cdots\delta_{n-1}}\Big[-\lambda_n e_n-\delta_{n-1}g_{n-1}(\boldsymbol{x}_{n-1})e_{n-1}\\ &\quad+\mathbf{Y}_n(\boldsymbol{\theta}_n-\widehat{\boldsymbol{\theta}}_n)\Big]-\frac{1}{\delta_1\cdots\delta_{n-1}}\sum_{\zeta=1}^{j}(\theta_{n\zeta}-\widehat{\theta}_{n\zeta})\frac{\dot{\widehat{\theta}}_{n\zeta}}{\rho_{n\zeta}}\\ &= -\frac{\lambda_n}{\delta_1\cdots\delta_{n-1}}e_n^2-\frac{1}{\delta_1\cdots\delta_{n-2}}g_{n-1}(\boldsymbol{x}_{n-1})e_{n-1}e_n\\ &\quad+\frac{1}{\delta_1\cdots\delta_{n-1}}\sum_{\zeta=1}^{j}(\theta_{n\zeta}-\widehat{\theta}_{n\zeta})\left(\mathrm{p}_{n\zeta}-\frac{\dot{\widehat{\theta}}_{n\zeta}}{\rho_{n\zeta}}\right)\\ &\leqslant -\frac{\lambda_n}{\delta_1\cdots\delta_{n-1}}e_n^2-s_{n-1}\end{aligned}$$

which completes the proof of Lemma 4.3. ■

APPENDIX C
COMPARISONS TO VDC [3]

In VDC, a robotic system (composed of rigid links connected by joints) is virtually decomposed into subsystems by placing conceptual *virtual cutting points* (VCPs); see [3, Def. 2.13] or [25, Def. 2]. VCPs allow that *rigid links* (described with 2nd-order ordinary differential equations) can be conceptually decomposed to separate parts where six-dimensional force/moment vector can be exerted from one part to another. Then, *virtual power flows* (VPFs) (see [3, Def. 2.16] or [25, Def. 4]) are used to address the dynamic interactions between the decomposed parts. When $\{\mathbf{A}\}$ is a three-dimensional orthogonal coordinate system locating at the VCP, a VPF $p_{\mathbf{A}}$ at $\{\mathbf{A}\}$ is defined as the inner product of the linear/angular velocity vector error $({}^{\mathbf{A}}V_{\mathrm{r}}-{}^{\mathbf{A}}V)\in\mathbb{R}^6$ and the force/moment vector error $({}^{\mathbf{A}}F_{\mathrm{r}}-{}^{\mathbf{A}}F)\in\mathbb{R}^6$, i.e., $p_{\mathbf{A}}=({}^{\mathbf{A}}V_{\mathrm{r}}-{}^{\mathbf{A}}V)^T({}^{\mathbf{A}}F_{\mathrm{r}}-{}^{\mathbf{A}}F)$.

In the original concept of *virtual stability* (see [3, Def. 2.17] or [25, Def. 5]), VPFs provide a stabilizing connection interface to the adjacent subsystems such that global asymptotic stability follows automatically when every subsystem is virtually stable. Similarly to VPFs, we designed stability connector s_k (in Def. 4.1) such that we were able to formulate the concept of virtual stability (in Def. 4.2) for SFF systems. Instead of using Lebesque L_2/L_∞ integrable functions, we based the results on the Lyapunov functions.

VDC introduced 1st- and 2nd-order differentiable projection functions $\mathscr{P}\in C^1$ and $\mathscr{P}_2\in C^2$ (see [3], [22]), satisfying the needs for robotic control purposes. The present study updated $\mathscr{P}_2\in C^2$ to the piecewise-continuous *(n-k)*th-order differentiable projection function $\mathscr{P}_k\in C^{n-k}$, $\forall k\in\{1,...,n\}$, targeted for *n*th-order SFF systems.

REFERENCES

[1] J.-J. E. Slotine and W. Li, *Applied nonlinear control*, Prentice-Hall Englewood Cliffs, NJ, 1991.

[2] S. Skogestad and I. Postlethwaite, *Multivariable feedback control: analysis and design*, vol. 2, New York: Wiley, 2007.

[3] W.-H. Zhu, *Virtual Decomposition Control - Toward Hyper Degrees of Freedom Robots*, Springer-Verlag, 2010.

[4] L. Hunt, R. Su, and G. Meyer, "Global transformations of nonlinear systems," *IEEE Trans. Autom. Control*, vol. 28, no. 1, pp. 24–31, 1983.

[5] M. Krstić, I. Kanellakopoulos, and P. Kokotović, *Nonlinear and Adaptive Control Design*, John Wiley & Sons, Inc., 1995.

[6] D. Swaroop, J. K. Hedrick, P. P. Yip, and J. C. Gerdes, "Dynamic surface control for a class of nonlinear systems," *IEEE Trans. Autom. Control*, vol. 45, no. 10, pp. 1893–1899, 2000.

[7] B. Song and J. K. Hedrick, *Dynamic surface control of uncertain nonlinear systems: an LMI approach*, Springer Science & Business Media, 2011.

[8] P. P. Yip and J. K. Hedrick, "Adaptive dynamic surface control: a simplified algorithm for adaptive backstepping control of nonlinear systems," *Int. J. Control*, vol. 71, no. 5, pp. 959–979, 1998.

[9] D. Wang and J. Huang, "Neural network-based adaptive dynamic surface control for a class of uncertain nonlinear systems in strict-feedback form," *IEEE Trans. Neural Netw.*, vol. 16, no. 1, pp. 195–202, 2005.

[10] Y.-H. Liu, "Adaptive dynamic surface asymptotic tracking for a class of uncertain nonlinear systems," *Int. J. Robust Nonlinear Control*, vol. 28, no. 4, pp. 1233–1245, 2018.

[11] J. Song, M. Yan, and P. Yang, "Neural adaptive dynamic surface asymptotic tracking control for a class of uncertain nonlinear system," *Circuits, Systems, and Signal Processing*, vol. 40, no. 4, pp. 1673–1698, 2021.

[12] B. Liu, M. Hou, C. Wu, W. Wang, Z. Wu, and B. Huang, "Predefined-time backstepping control for a nonlinear strict-feedback system," *Int. J. Robust Nonlinear Control*, vol. 31, no. 8, pp. 3354–3372, 2021.

[13] G. Zhu, J. Du, and Y. Kao, "Command filtered robust adaptive NN control for a class of uncertain strict-feedback nonlinear systems under input saturation," *J. Franklin Inst.*, vol. 355, no. 15, pp. 7548–7569, 2018.

[14] J. Ding and W. Zhang, "Finite-time adaptive control for nonlinear systems with uncertain parameters based on the command filters," *Int. J. Adapt. Control Signal Process.*, vol. 35, no. 9, pp. 1754–1767, 2021.

[15] J.-J. Slotine and J. K. Hedrick, "Robust input-output feedback linearization," *Int. J. control*, vol. 57, no. 5, pp. 1133–1139, 1993.

[16] M. Won and J. K. Hedrick, "Multiple-surface sliding control of a class of uncertain nonlinear systemsm," *Int. J. Control*, vol. 64, no. 4, pp. 693–706, 1996.

[17] W.-H. Zhu, Y.-G. Xi, Z.-J. Zhang, Z. Bien, and J. De Schutter, "Virtual decomposition based control for generalized high dimensional robotic systems with complicated structure," *IEEE Trans. Robot. Autom.*, vol. 13, no. 3, pp. 411–436, 1997.

[18] S. Mastellone and A. van Delft, "The impact of control research on industrial innovation: What would it take to make it happen?," *Control Engineering Practice*, vol. 111, pp. 104737, 2021.

[19] J. Mattila, J. Koivumäki, D. Caldwell, and C. Semini, "A survey on control of hydraulic robotic manipulators with projection to future trends," *IEEE/ASME Trans. Mechatronics*, vol. 22, no. 2, pp. 669–680, 2017.

[20] J. Koivumäki and J. Mattila, "Adaptive and nonlinear control of discharge pressure for variable displacement axial piston pumps," *J. Dyn. Sys., Meas., Control*, vol. 139, no. 10, pp. 101008, 2017.

[21] P. Seiler and A. Alleyne, "Dissipative adaptive control for strict feedback form systems," *Eur. J. Control*, vol. 8, no. 5, pp. 435–444, 2002.

[22] W.-H. Zhu and J. De Schutter, "Adaptive control of mixed rigid/flexible joint robot manipulators based on virtual decomposition," *IEEE Trans. Robot. Autom.*, vol. 15, no. 2, pp. 310–317, 1999.

[23] H.K. Khalil, *Nonlinear systems*, Prentice Hall, 3rd edition, 2002.

[24] J.-P. Humaloja, J. Koivumaki, L. Paunonen, and J. Mattila, "Decentralized observer design for virtual decomposition control," *IEEE Trans. Autom. Control*, vol. 67, no. 5, pp. 2529–2536, 2022.

[25] J. Koivumäki and J. Mattila, "Stability-guaranteed force-sensorless contact force/motion control of heavy-duty hydraulic manipulators," *IEEE Trans. Robot.*, vol. 31, no. 4, pp. 918–935, 2015.