

RESEARCH ARTICLE

Fractional order inspired iterative adaptive control

Bence Varga^{1,2} , József K. Tar^{1,3,4}  and Richárd Horváth² 

¹Doctoral School of Applied Informatics and Applied Mathematics, Obuda University, Budapest, Hungary, ²Banki Donat Faculty of Mechanical and Safety Engineering, Obuda University, Budapest, Hungary, ³John von Neumann Faculty of Informatics, Obuda University, Budapest, Hungary, and ⁴Antal Bejczy Center of Intelligent Robotics, Obuda University, Budapest, Hungary

Corresponding author: Bence Varga; Email: varga.bence@bgk.uni-obuda.hu

Received: 15 November 2022; **Revised:** 29 October 2023; **Accepted:** 31 October 2023;

First published online: 15 December 2023

Keywords: control of robotic systems; mechatronic systems; robot dynamics; automation; design

Abstract

Although several studies have revealed that fractional order controllers usually outperform conventional integer-order control solutions, fractional order controllers are not yet widely applied in industrial applications due to their complex mathematical background. In this paper, further improvements of a simple weighted sum feedback design are introduced that imitates the behavior of a fractional order controller but is free from its various formal restrictions. The proposed control solution has the main characteristics of a fractional order controller, such as finite memory length, excellent transient response with no overshoot and robust behavior, but it is placed into a much simpler mathematical framework. In the current paper, a simple derivative term was incorporated in the design which made the controller's output more stable by completely eliminating output chattering. The proposed control method was developed for a general second-order system. It was tested in a fixed point iteration-based adaptive control scenario, through simulations using a robotic example and on experimental basis as well, utilizing a simple one-degree-of-freedom electromechanical system. The presented experiments are the first systematic investigations of the fixed point iteration-based adaptive control method.

1. Introduction

Fractional calculus is a generalization of differential calculus invented by Isaac Newton and Gottfried Leibniz, which includes non-integer order differential and integral. The origin of fractional calculus dates back to the late 17th century, since in 1695 L'Hospital wrote a letter to Leibniz, in which the following famous question arose – “*What if n be $1/2$?*” (this question referred to a differential of $1/2$ th order) [1]. Over the years, many famous mathematicians contributed to the development of fractional calculus including Euler, Lagrange, Laplace, Fourier, etc. [1, 2], also various definitions were developed for fractional order operators, which can be used in different research areas [3, 4]. The most commonly used definitions are the Riemann-Liouville, the Grünwald-Letnikov and the Caputo versions [5]. In the past few decades, fractional order calculus gained a great deal of attention both from scientific and industrial community. Extensive research was carried out in different application areas, showing that it has ample applications in biology [6], modeling distributed systems [7], viscoelastic phenomena [8], systems with delay [9], industrial applications [10], etc. It is considered at the deepest level of formulating physical principles [11]. Discrete-time realization of the concept was considered for example in ref. [12]. Regarding control applications, the superiority of fractional order control solutions was emphasized in refs. [13, 14], foreshadowing the widespread of Fractional Order Proportional-Integral-Derivative (FOPID) controllers in industrial applications. Perhaps one of the most known features of FOPID controllers is the excellent startup response. In ref. [15], a Fractional Order Proportional Integral (FOPI) and a conventional PI (Proportional Integral) control solution were tested on an electromechanical drive

system in which the FOPI controller outperformed the PI controller, by almost fully eliminating the initial overshoot and the oscillation of the system, both in position and speed control. Various other studies showed similar results using fractional order feedback terms in their design, in robotics and in other process control applications (e.g.: refs. [16–18]). Besides excellent startup response of fractional order controllers (FOCs), some studies also showed that they can work more energy efficiently [19], which makes this control approach more suitable for “off grid” applications (i.e., mobile robots, drones, etc.).

Fractional order feedback can be formulated in various manners. Perhaps the most intuitive solution is the generalization of a PID controller, introduced by Podlubny in ref. [20] as

$$u(t) = K_p e(t) + K_i \mathcal{I}_t^\lambda e(t) + K_d \mathcal{D}_t^\mu e(t), \tag{1}$$

where \mathcal{I}_t^λ is a general non-integer order integral operator of order $\lambda \in \mathbb{R}$, and in similar manner, \mathcal{D}_t^μ is a general non-integer order differential operator of order $\mu \in \mathbb{R}$ and $e(t)$ is the error. Equation (1) is often called the output equation of a $PI^\lambda D^\mu$ controller, which yields a simple PID controller with K_p proportional, K_i integral and K_d derivative gain, for $\lambda, \mu = 1$ parameter settings. For FOPID controllers, theoretically λ, μ might have any values, however, due to the physical interpretations, it is usually chosen from $[0, 2]$ interval [5], providing additional tuning options to the user to more efficiently shape the system response, that way stabilizing the controlled system and improve its robustness. Using the Laplace transform of fractional derivative and integral with zero initial conditions [21], the parallel form of transfer function can be obtained,

$$C(s) = K_p + \frac{K_i}{s^\lambda} + K_d s^\mu. \tag{2}$$

It is well known that fractional order systems are infinite dimensional so, in a control application, infinite number of elements should be considered from error signal’s past (infinite memory). That way, digital implementation of FOCs usually involves some kind of approximation of fractional operators, for example, Oustaloup’s recursive filter method [22] (or its improved version [23]) is often used [24–27]. The essence of this solution is the approximation of $s^q, q \in \mathbb{R}, q \in [-1, 1]$ fractional terms in a given frequency range $[\omega_l, \omega_h]$ as,

$$H(s) = s^q = K \prod_{k=-N}^N \frac{s + \tilde{\omega}_k}{s + \omega_k}, \tag{3}$$

where the $\tilde{\omega}_k$ and ω_k are consecutive zero and pole frequencies, which can be recursively calculated $\tilde{\omega}_k = \omega_l \frac{\omega_h}{\omega_l} \frac{k+N+0.5(1-q)}{2N+1}, \omega_k = \omega_l \frac{\omega_h}{\omega_l} \frac{k+N+0.5(1-q)}{2N-1}, K = \omega_k^q$ is the filter gain and the order of the recursive filter is $n = 2N + 1$.

Oustaloup’s recursive filter provides very good approximations of the fractional order elements, however, it is limited to the $[\omega_l, \omega_h]$ frequency range. To overcome this issue, a time domain approximation was introduced in ref. [28] by Machado *et al.* for motion control applications. This solution is a simple n -term truncated series approximation of the Grünwald-Letnikov definition of fractional derivatives, which provides a simple digital implementation and robust behavior, however, it requires very low sampling times and high number of terms for good approximation.

These two examples nicely exhibit the implementation complexity of FOCs compared to that of the traditional PID controllers, since Eq. (1) can be implemented in a single line of code for $\lambda = \mu = 1$ integer order case. This induced our research in a simple weighted sum feedback design (some preliminary results can be found in ref. [29]), which has properties similar to that of a FOC, such as short-term finite memory, robust behavior and excellent transient response with no overshoots, but completely independent from the mathematical framework of fractional order control that way providing a simple alternative to FOCs in motion control applications. The proposed weighted sum feedback design can be implemented in a fixed-point iteration-based (FPI-based) control scenario, that way a robust adaptive control solution can be achieved with good error convergence. The control method can be used in various applications, for example, controlling industrial robots (Section 5) or in life sciences. The application of FPI-based control in robotics has been widely investigated and it was shown that it can be combined

with various classical methods, for example, it can improve the stability of a Slotine-Li Adaptive Robot Controller [30] or improve the parameter identification process in case of Adaptive Inverse Dynamics Controller [31]. The adaptive deformation, which is the essence of FPI-based control, can guarantee a strictly prescribed behavior for the components of $e(t)$, which is highly desirable for robot control applications, where no overshoot in the joint coordinates is also necessary for safe operation. The main contributions of this paper are:

- In this paper, further improvement of a simple weighted sum feedback design is introduced, which eliminates the design issues of the originally proposed solution [29]. This improved version generates a nice continuous output signal without output chattering, and it is easy to implement in a simple microcontroller.
- The proposed control solution is tested and compared to a PID control solution on experimental basis on a simple electromechanical system using FPI-based adaptive control scenario. It can be an important milestone in the research of FPI control since although this control solution was introduced in 2009 [32], no systematic experimental results have been presented on this topic so far.

The paper is structured in the following manner. In Section 2, some implementation details of a fractional order computed torque controller are introduced, which served as an inspiration for our feedback design. Section 3 was divided into two parts. In the first part, some details were introduced about the originally proposed weighted sum design, and in the second part, the newly proposed and improved weighted sum design is introduced. In Section 4, a detailed description of FPI-based control is presented. Section 5 contains some simulation results for a robotic system since this paper serves as continuation of ref. [29]. The proposed control solution was also verified on experimental basis in Section 6. Finally, in Section 7, the conclusions are summarized.

2. Fractional order computed torque control

In control applications, to achieve asymptotic error convergence, most commonly PID-type error feedback is used, which was developed at the beginning of the 20th century [33, 34]. For complex non-linear systems such as industrial robots, the mathematically simplest control solution is the Computed Torque Control (CTC) [35], which directly utilizes the available dynamic model of the system. However, later it was revealed that CTC controllers – as well as other model-based control solutions – were burdened by modeling imprecisions [36] which makes precise trajectory tracking rather difficult. CTC is a special implementation of PID control for trajectory-tracking application. If $q^N(t) \in \mathbb{R}^n$ denotes the nominal trajectory (in joint space), which should be precisely tracked by the robot and $e_{int}(t) := \int_{t_0}^t e(\xi) d\xi$ is the integral of the trajectory tracking error ($e(t) = q^{Nom}(t) - q^{Real}(t)$), then a kinematic prescription can be formulated as

$$\left(\Lambda + \frac{d}{dt}\right)^3 e_{int}(t) \equiv 0. \quad (4)$$

Evidently, the general solution of (4) can be given as the linear combination of certain basis functions that span the linear space of the solutions as $e_{int}(t) = \sum_{\ell=0}^2 a_\ell (t - t_0)^\ell \exp(-\Lambda(t - t_0))$. In that solution the initial conditions are determined by the parameters $\{a_0, a_1, a_2\}$, and the exponential terms of a_0 , a_1 and a_2 coefficients are mapped to zero by the corresponding operators $(\Lambda + \frac{d}{dt})$, $(\Lambda + \frac{d}{dt})^2$ and finally $(\Lambda + \frac{d}{dt})^3$. That yields that the “desired value” of second derivative of the joint coordinates ($\ddot{q}^{Des}(t) \in \mathbb{R}^n$) can be obtained as

$$\ddot{q}^{Des}(t) = \ddot{q}^N(t) + \Lambda^3 e_{int}(t) + 3\Lambda^2 e(t) + 3\Lambda \dot{e}(t). \quad (5)$$

In Eq. (5), $\mathbb{R} \ni \Lambda > 0$ is the “error decay time constant” and $\dot{e}(t)$ is the derivative error. Similar considerations can be done for the $(\Lambda + \frac{d}{dt})^2 e(t) \equiv 0$ (PD controller) and for the $(\Lambda + \frac{d}{dt}) e(t) \equiv 0$ (the P

controller for first-order systems), too. The obtained values of $\ddot{q}^{Des}(t)$ can be directly applied in the available dynamic model of our system in order to calculate the necessary control force ($Q(t) \in \mathbb{R}^n$), which is then applied to the system. However, such solutions can generate significant overshoots or undershoots due to the integer order calculus used in their design. They are very sensitive to modeling errors, too. The PID feedback gains can be identified in (5) as $K_p = 3\Lambda^2, K_d = 3\Lambda, K_i = \Lambda^3$. These gains can be calculated independently in more sophisticated manner, too, using the Lyapunov-equation [37, 38], however, usually this requires continuous tuning of the parameters.

The control performance can be increased by introducing the second integral of the tracking error [39], but this may result in more significant overshoots and undershoots of the system,

$$e_{int2}(t) := \int_{t_0}^t e_{int}(\xi) d\xi \text{ with } \left(\Lambda + \frac{d}{dt} \right)^4 e_{int2}(t) \equiv 0. \tag{6}$$

In similar manner, multiple integral terms can be incorporated into the kinematic prescription, which leads to one of the simplest definitions of fractional order feedback [40]. The multiple integral terms can be expressed using ‘‘Riemann-Liouville n -fold Integral’’ formula,

$$F_{n+1}(t) = \int_{t_0}^t F_n(\xi) d\xi, F_n(t) = \frac{1}{(n-1)!} \int_{t_0}^t f(\xi)(t-\xi)^{n-1} d\xi. \tag{7}$$

In Eq. (7), the solution can be extended to real and complex numbers if the $\frac{1}{(n-1)!}$ term is replaced with Euler’s Gamma function ($\Gamma(s) = (s-1)!, s \in \mathbb{C}$), which can be calculated for complex numbers as well. The Riemann-Liouville Fractional Integral and Derivative of order $\mathbb{C} \ni \alpha > 0$ are defined as the generalization of the Riemann-Liouville n -fold integral as,

$$\begin{aligned} \mathcal{I}_a^\alpha f(t) &\stackrel{def}{=} \frac{1}{\Gamma(\alpha)} \int_a^t f(\xi)(t-\xi)^{\alpha-1} d\xi, (s-1)! \equiv \Gamma(s), \\ \mathcal{D}_a^\alpha f(t) &\stackrel{def}{=} \frac{1}{\Gamma(1-\alpha)} \frac{d}{dt} \int_a^t f(\xi)(t-\xi)^{-\alpha} d\xi, \Gamma(s) \stackrel{def}{=} \int_0^\infty \xi^{s-1} e^{-\xi} d\xi. \end{aligned} \tag{8}$$

Upon careful revision of Eq. (8), some numerical issues can be revealed, since in the definition of Riemann-Liouville Fractional Derivative, the integrand is singular at its upper boundary ($\xi = t$). To tackle this problem, a simple solution was introduced in refs. [41, 42], in which the numerical issues were avoided by applying an integer order integral of the fractionally derivated term. The idea is that consider the derivative term in (5) as the integral of the second derivative ($\dot{e}(t) = \dot{e}(t_0) + \int_{t_0}^t \ddot{e}(\xi) d\xi$), in which the derivative can be replaced with a fractional order $\alpha \in [0, 1]$ derivative as $\dot{e}(t) = \int_{t_0}^t [\mathcal{D}_{t_0}^\alpha \dot{e}(\xi)] d\xi$. That way a simple fractional order PID-type kinematic design can be achieved as,

$$\ddot{q}^{Des} = \ddot{q}^N(t) + \Lambda^3 e_{int}(t) + 3\Lambda^2 e(t) + 3\Lambda_2 \int_{t_0}^t [\mathcal{D}_{t_0}^\alpha \dot{e}(\xi)] d\xi. \tag{9}$$

Due to the integer order integral applied on the fractional derivative of $\dot{e}(t)$, the numerically problematic integral in (8) can be eliminated. The integral in the last term of (9) can be numerically approximated, due to the ‘‘forgetting nature’’ of the integral term in the definition equation of fractional derivative, which allows the application of finite memory length (‘‘old’’ data on the trajectory tracking error makes only negligible contribution – ‘‘short memory principle’’ [43]). The interval $[(t_0 \leq) t - T, t]$ was considered as a grid of points $\{t - K\delta t = t - T, \dots, t - 2\delta t, t - \delta t, t\}$, and it was assumed that $\dot{e}(t)$ did not vary considerably within the intervals. So $\int_{t_i}^{t_i+\delta t} \dot{e}(\xi)(t-\xi)^{-\alpha} d\xi \approx \dot{e}(t_i + \delta t) \int_{t_i}^{t_i+\delta t} (t-\xi)^{-\alpha} d\xi$ was applied in which for the integral simple closed-form formula was available. The closed-form integral term can be easily calculated for each $[t_i, t_i + \delta t]$ interval, which yields C_i coefficients in (10), that way a ‘‘weighted sum type’’ feedback can be achieved as,

$$\sum_{\ell=0}^{\ell_{max}} \dot{e}(t_i - \ell\delta t) C_\ell, C_0 > C_1 > \dots > C_{\ell_{max}} > 0. \tag{10}$$

In the above equation, the “forgetting nature” of the solution can be revealed by C_l coefficients, also it becomes apparent that this solution is free from the singularity issue mentioned previously (when $\xi = t$) by the fact that $C_0 \gg C_1 > 0$. The parameters T and α can be arbitrarily set by the user. The latter describes the “speed of forgetting the past,” and on the other hand, T characterizes the “memory length” of this solution. If a small value is selected for T , a similar solution can be achieved as in (5).

In both refs. [41] and [42], it was shown that this type of fractional feedback can yield better transient response than conventional PID control solution. If α , Λ_1 and Λ_2 parameters are set correctly, the initial overshoot of the system can be reduced or sometimes completely evaded.

3. Fractional order inspired weighted sum-type feedback

One of the key features of the solution proposed in (9) is the finite memory length incorporated in its design due to fractional feedback term, which makes the trajectory tracking of a particular control system in the initial phase much better. That invokes the question “*What if a similar principle is applied without the mathematical restrictions of fractional order calculus?*”. An answer to this question was proposed in ref. [29], which is a simple forgetting weighted sum-type solution characterized in the following subsection, including some of its advantages and disadvantages as well.

3.1. Preliminary research

Consider an order 2 system in which, on the basis of the dynamic model, $\ddot{q}(t)$ can be set by the controller. Suppose that the trajectory tracking error is exponentially driven to 0 in monotonic manner, which makes it possible to apply the following time approximation,

$$\dot{e}(t) = -\Lambda e(t), \text{ that numerically is} \tag{11a}$$

$$e(t_{i+1}) = e(t_i)(1 - \delta t \Lambda). \tag{11b}$$

This approximation has obviously no memory effect, so the following modification was applied (the following feedback design will be denoted as Weighted Sum or WS feedback in the rest of the paper):

$$e(t_{i+1}) = e(t_i) (1 - \delta t \Lambda C_0) - \sum_{\ell=1}^{H-1} \delta t \Lambda C_\ell e(t_{i-\ell}) = a_0 e(t_i) - \sum_{\ell=1}^{H-1} a_\ell e(t_{i-\ell}), \tag{12}$$

in which the discrete memory length is $H - 1$. In the above equation, the C_l coefficients serve a similar purpose as in Eq. (10), describing the “speed of forgetting the past”. Equation (12) can be applied over a grid of points $\{t - l\delta t = t - T, \dots, t - 2\delta t, t - \delta t, t\}$, where the coefficient can be calculated using an arbitrarily chosen function, $C_\ell \propto f(t)$, so that $1 \approx a_0 < 1$, and for $\ell \in \{1, \dots, H - 1\} 0 \approx a_\ell > 0$ for small δt and reasonable $\Lambda > 0$. As long as in a t_i time instant, the calculated value of the desired trajectory tracking error is $e^{Des}(t_{i+1})$, the desired value of joint coordinates (sticking to a robotic control application) can be calculated as $q^{Des}(t_{i+1}) = q^N(t_{i+1}) - e^{Des}(t_{i+1})$, which leads to:

$$\ddot{q}(t_{i-1})^{Des} \approx (q^{Des}(t_{i+1}) - 2q(t_i) + q(t_{i-1}))/\delta t^2, \tag{13}$$

using forward differences for differentiation. The convergence of sequence (Eq. (9)) can be described in matrix form, for example, for a particular memory length ($H = 4$)

$$\begin{bmatrix} e(t_{i+1}) \\ e(t_i) \\ e(t_{i-1}) \\ e(t_{i-2}) \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} e(t_i) \\ e(t_{i-1}) \\ e(t_{i-2}) \\ e(t_{i-3}) \end{bmatrix}, \tag{14}$$

where matrix A contains all coefficients of each error term as $a_{11} = a_0 = 1 - \delta t \Lambda C_0$, $a_{12} = -a_1 = -\delta t \Lambda C_1$, etc., and it can be easily transformed into the Jordan canonical form through similarity transformation. From the structure of each Jordan Blocks $(\lambda I + \Delta)$ in the main diagonal, it immediately follows that

$$(\lambda I + \Delta)^n = \sum_{\ell=0}^{m-1} \frac{n!}{\ell!(n-\ell)!} \lambda^{n-\ell} \Delta^\ell \rightarrow 0 \text{ as } n \rightarrow \infty \text{ if } |\lambda| < 1. \tag{15}$$

where $\lambda \in \mathbb{C}$ is an eigenvalue of the matrix in (14), and Δ is a *nilpotent upper triangular matrix* of size $m \times m$ with the property of $\Delta^m = 0$. The eigenvalues are the roots of the characteristic polynomial ($\det(\lambda I - A)$), which can be simply formulated as $p_{(H)}(\lambda) = \lambda p_{(H-1)} - a_{1H}$, for example, for memory length $H = 4$

$$p_{(4)}(\lambda) = \begin{vmatrix} \lambda - a_{11} & -a_{12} & -a_{13} & -a_{14} \\ -1 & \lambda & 0 & 0 \\ 0 & -1 & \lambda & 0 \\ 0 & 0 & -1 & \lambda \end{vmatrix} = (\lambda - a_{11})\lambda^3 - a_{12}\lambda^2 - a_{13}\lambda - a_{14}, \tag{16}$$

$p_{(4)}(\lambda) = \lambda p_{(3)}(\lambda) - a_{14}$, or in similar manner $p_{(5)}(\lambda) = \lambda p_{(4)}(\lambda) - a_{15}$, etc., that is easy to see when the determinant of (16) obtained through expansion according the last column, since most of the minors of matrix A are canceled due to the 0 elements. For $\delta t \rightarrow 0$ limit, the roots of the polynomial can be easily obtained since $a_{11} \rightarrow 1$, and for $a_{1k} \rightarrow 0$ for all $k > 0$, for example, $p_{(5)}(\lambda) = (\lambda - 1)\lambda^4$ has the roots $\lambda_1 = 1$, and with greater multiplicity $\lambda_2 = 0$. For a small but non-zero δt , this polynomial varies continuously and its roots remain in the vicinity of 1 and 0, so based on the convergence requirement given in Eq. (15), the only concern is the root near 1, when $\delta t > 0$. For the estimation of its modification the form, $\lambda_1: 1 \rightarrow 1 + \mu \delta t$ can be considered in the first order of δt while the other terms are modified as $a_{11}: 1 \rightarrow 1 - \Lambda C_0 \delta t$, and $a_{12}: 0 \rightarrow -\Lambda C_1 \delta t$, etc. Consequently $(1 - a_{11})$ varies as $0: \rightarrow 1 + \mu \delta t - (1 - \Lambda C_0 \delta t) = (\mu + \Lambda C_0) \delta t$ that has first order in δt . Therefore, only the order 0 terms must be taken into account in λ^3 , in λ^2 and in λ^1 as 1. For instance, for the equation $p_{(4)}(1 + \mu \delta t) = 0$, the approximation $\mu + \Lambda C_0 + \Lambda C_1 + \Lambda C_2 + \Lambda C_3 = 0$ is obtained leading to $\lambda_1: 1 \rightarrow 1 - \delta t(\Lambda C_0 + \Lambda C_1 + \Lambda C_2 + \Lambda C_3) = 1 - \delta t \Lambda < 1$ that guarantees convergence for small δt and reasonable $\Lambda > 0$ values [29].

The proposed solution PUMA method was successfully applied in an FPI-based adaptive control scene for controlling a 3-Dof PUMA-type robot arm (some simulation results are presented in Section 5 for comparison purposes). The simulations showed very nice behavior of the controlled system in the early stages of the control. The monotonic decrease of the trajectory tracking error was observed with no overshoots, on the other hand, the computational demand increased (which is a general problem for fractional order control). Later upon further investigation of the control solution, the following issues were revealed:

- When the proposed solution was applied without adaptive control, a significant chattering was observed in the control force (which was nicely smoothed when FPI-based adaptive control was used).
- Although chattering was sufficiently reduced in our preliminary simulations with the use of adaptive deformation, significant speed fluctuation was observed when applying the proposed control method in a motor control application. This can be caused by the time delays of the system and the higher cycle time of controller, which made the adaptive controller slightly less efficient than in the simulations, that way the chattering could not be smoothed anymore.
- The solution became divergent when noise filtering was applied.

To resolve these issues a revised control solution will be proposed in the next section.

3.2. Higher order implementation of lower order control task

By considering the discrete-time approximation of the problem with time-resolution δt , the following statement can be done for (11a):

$$e^{Des}(t + \delta t) \approx (1 - \Lambda \delta t)e(t). \tag{17}$$

Assuming that in a given $t - \delta t$ control cycle, the initial conditions are $e(t - \delta t)$ (trajectory tracking error in the current control cycle) $\dot{e}(t - \delta t)$ (derivative of trajectory tracking error) that determines $e(t)$, too. Using forward differences according to (13), the acceleration, which should be applied to the system assuming that $\ddot{q}(t)$ can be set instantly, can be calculated as (a similar solution was applied in ref. [44]):

$$\ddot{e}^{Des}(t - \delta t) \approx \frac{e^{Des}(t + \delta t) - 2e(t) + e(t - \delta t)}{\delta t^2}, \tag{18}$$

$$\dot{e}^{Des}(t - \delta t) \approx \frac{(1 - \Lambda \delta t)e(t) - 2e(t) + e(t - \delta t)}{\delta t^2}, \tag{19}$$

$$\ddot{e}^{Des}(t - \delta t) \approx \frac{-\Lambda \delta t e(t) - e(t) + e(t - \delta t)}{\delta t^2}, \tag{20}$$

$$\ddot{e}^{Des}(t - \delta t) \approx \frac{-\Lambda e(t)}{\delta t} - \frac{\dot{e}(t - \delta t)}{\delta t}, \tag{21}$$

$$\ddot{e}^{Des}(t - \delta t) \approx -\Lambda \frac{e(t) - e(t - \delta t)}{\delta t} - \Lambda \frac{e(t - \delta t)}{\delta t} - \frac{\dot{e}(t - \delta t)}{\delta t}, \tag{22}$$

$$\ddot{e}^{Des}(t - \delta t) \approx -\Lambda \dot{e}(t - \delta t) - \frac{\Lambda}{\delta t} e(t - \delta t) - \frac{\dot{e}(t - \delta t)}{\delta t}, \tag{23}$$

$$\ddot{e}^{Des}(t - \delta t) \approx -\Lambda \dot{e}(t - \delta t) - \frac{1}{\delta t} (\dot{e}(t - \delta t) + \Lambda e(t - \delta t)), \text{ where} \tag{24}$$

$$\dot{e}(t - \delta t) \approx \frac{e(t) - e(t - \delta t)}{\delta t}. \tag{25}$$

The freshly set value of $\ddot{q}^{Des}(t - \delta t)$ can be calculated as:

$$\ddot{q}^{Des}(t - \delta t) \approx q^N(t - \delta t) + \Lambda \dot{e}(t - \delta t) + \frac{1}{\delta t} (\dot{e}(t - \delta t) + \Lambda e(t - \delta t)). \tag{26}$$

If the dynamic model of the controlled system is available, the necessary force can be computed as

$$Q(t - \delta t) = F(q(t - \delta t), \dot{q}(t - \delta t), \ddot{q}^{Des}(t - \delta t)), \tag{27}$$

this control strategy can be realized for a second-order system consistently.

In a more sophisticated manner, the solution can be formulated using differential equations since the second derivative of the tracking error can be introduced as $\ddot{e}(t) = -\Lambda \dot{e}(t)$, however, this would result in information loss since the value of e remains undetermined to the tune of a constant value. Instead, the following idea can be chosen: consider the difference of $\dot{e}(t)$ and $-\Lambda e(t)$ and drive it asymptotically to zero by the strategy

$$\left(\lambda + \frac{d}{dt} \right) (\dot{e}(t) + \Lambda e(t)) \equiv 0 \text{ resulting } \ddot{e}(t) = -\Lambda \dot{e}(t) - \lambda (\dot{e}(t) + \Lambda e(t)). \tag{28}$$

Evidently, if $\lambda \gg \Lambda$, $(\dot{e}(t) + \Lambda e(t))$ very quickly has to converge to zero, that is, the original first-order strategy is soon realized. Furthermore, it can be observed that (28) essentially corresponds to (24) if $\lambda = \frac{1}{\delta t}$. In a digital controller, the $\Lambda \ll \frac{1}{\delta t}$ condition used to be generally valid.

Introducing an $x \stackrel{\text{def}}{=} [e \ \dot{e}]^T$ state variable and its time derivative $\dot{x} = [\dot{e} \ \ddot{e}]^T$, the Eq. (28) can be rewritten in matrix form

$$\dot{x} = \begin{bmatrix} \dot{e} \\ \ddot{e} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\Lambda\lambda & -(\Lambda + \lambda) \end{bmatrix} \begin{bmatrix} e \\ \dot{e} \end{bmatrix} = Ax. \tag{29}$$

The characteristic polynomial of matrix A that is $\det(\xi I - A) = \xi^2 + \xi(\Lambda + \lambda) + \Lambda\lambda$ and its roots are $\xi_1 = -\Lambda$ and $\xi_2 = -\lambda$. Ensuring that the real part of the eigenvalues are negative ($\Re(\xi_1) < 0$ and $\Re(\xi_2) < 0$), so $\Lambda, \lambda > 0$ a stable solution can be obtained.

Obviously neither Eq. (28) nor (27) (which is a discrete-time approximation) have memory effect. However, utilizing (12) and its derivative, the following solution can be obtained,

$$\ddot{q}^{Des}(t) = \ddot{q}^N(t) + (\lambda + \Lambda)\dot{e}(t) + \Lambda\lambda e(t), \text{ so} \tag{30a}$$

$$\ddot{q}^{Des}(t) = \ddot{q}^N(t) + (\lambda + \Lambda)\dot{e}(t_i) + \Lambda\lambda e(t_i) - (\lambda + \Lambda) \sum_{\ell=0}^{H-1} C_\ell \Lambda \delta t \dot{e}(t_{i-\ell}) - \Lambda\lambda \sum_{\ell=0}^{H-1} C_\ell \Lambda \delta t e(t_{i-\ell}), \tag{30b}$$

which can be implemented in a simple embedded system. This feedback solution will be denoted as WSPD or Weighted Sum PD feedback in the rest of the paper since it is the augmentation of Eq. (12) with a derivative term.

4. Fixed point iteration-based adaptive control

The first efforts to develop adaptive control solutions were made in the 1950s when the design of autopilot solutions for high-performance aircraft was started [45]. These machines work in various operating conditions (various speeds and altitudes, different weather circumstances, other external disturbances, etc.), which means that the system dynamics are almost constantly changing from controller’s point of view. This could not be handled by an ordinary constant-gain feedback controller. Despite the enthusiasm in the early stages of the research, due to lack of knowledge in stability analysis (although the foundations were already given by Lyapunov at that time [38]) and catastrophic failure in a flight test [46], the research in adaptive control came almost to a halt by the end of the decade [47]. However, the research in adaptive control came to a new life in the next decade when various mathematical concepts were established [48] and the Lyapunov theory was successfully applied in the design [49]. Since then the Lyapunov function-based design has become superior in adaptive control (few application examples, e.g., refs. [50, 51]).

In the last few decades, adaptive backstepping control got a great deal of attention from the scientific community [52–54]. This control method is built on a recursive solution, based on Lyapunov’s approach, and it can ensure the stability of the system. It was shown in a recent article that backstepping control can be combined with fixed point iteration-based adaptive deformation [55] as well, that way a robust adaptive controller can be obtained. The essence of this solution is that the design starts from a known stable state which is separated by the largest number of integration from control input and backs out step by step, stabilizing each subsystem until final external control input is achieved. Although backstepping control is an effective non-linear control solution, it requires some tedious design work from the control engineer and the problem complexity drastically increases for higher-order systems.

In the recent years, artificial intelligence techniques came into prominence in adaptive control, such as neural network (NN) [56] and fuzzy logic system [57]. These techniques are mainly used to address unknown system dynamics (e.g., friction, dead-zone, etc.) through parameter estimation where some kind of friction model is utilized, although in ref. [56], a new learning algorithm was introduced for simultaneous motion control and parameter estimation. In NN-based control, a general problem is that small residual approximation errors can affect the control performance and also usually these techniques involve slow learning resulting in long transient stage. The transient response of the controlled system can be improved in various manner. In ref. [58], prescribed performance controller (PPC) was introduced

and experimentally verified, where asymptotic error convergence was achieved through RISE-based (robust integral of sign of error) control design. In this paper, a smooth friction model [59] was also used in combination with an echo state neural network to learn the system dynamics and compensate for unknown non-linear model components. However, PPCs could be burdened by singularity due to the inverse error transform function in their design. On the other hand, the asymptotic tracking and prescribed transient behavior can be also guaranteed using funnel control [60, 61], which is simple to implement and does not require precise parameter identification.

Fixed point iteration-based adaptive control was introduced in 2009 [32] as an alternative option for motion control applications. It can be regarded as primitive case of machine learning and it can ensure fast convergence without precise dynamic model of the controlled system. Its introduction was motivated by certain important features of the Lyapunov function-based design as follows:

- The Lyapunov function can be regarded as the square of a special norm in which the metric tensor's components depend on the feedback gains. In this technique, only the monotonic decrease of this norm can be guaranteed while the individual components of which this norm is built up can increase and decrease, too. In certain application fields as for example, life sciences, the monotonic decrease of the individual components may have significance.
- In many application fields, it is impossible to directly measure the components of which the Lyapunov function is built up. Since they are organic part of the metric in use, often complicated state estimation techniques have to be implemented to make the Lyapunov function-based design applicable.

The suggested new approach was exempt from such difficulties: it directly aimed at the decrease of a given component without needing the measurement or estimation of each component of the state variable. It was extended to multivariable systems, too, and operated on the basis of Banach's fixed point theorem. It needed the observation only of the variable of interest on the following basis. In the case of a control task of relative order $n \in \mathbb{N}$, the $q^{(n)}$ derivative of the generalized coordinate can be immediately set by the control force, while the lower order derivatives evolve relatively slowly. Normally, by the use of the Lie derivatives it is possible to calculate the functional relationship between the control force and the appropriate derivative, however, this computation is often very complicated, time consuming, and in the possession of only an approximate system model, often is useless. Instead, a simple affine model in the form $Q(t) = Aq^{(n)}(t) + B$ can be used as initial approximation, since the other slowly varying components approximately can be considered as constants. In many single variable systems, the sign of A can be assumed to be constant positive or negative. The output of the kinematic design can generate a $q^{(n)Des}$ derivative so that if it would be realized the tracking error should asymptotically converge to zero. If the available mathematical model of the system would be precise this scenario could be realized. However, in the case of an approximate model, the model itself together with the controlled system bring about a response function that maps the desired input to the observable output as $q^{(n)Real}(t) = f(q^{(n)Des}(t)) \neq q^{(n)Des}(t)$. To realize the desired input, according to Fig. 1, the input of the approximate model should be deformed so that $q^{(n)Des}(t) = f(q^{(n)Def}(t))$, in which $q^{(n)Def}(t)$ is an adaptively deformed input. The necessary deformation can be iteratively found if it is known that the response function is approximately direction keeping, that is, if for an infinitesimally small Δx , $\Delta f := f(x + \Delta x) - f(x)$ the scalar product $\Delta x^T \Delta f > 0$. (This concept can be regarded as the generalization of the single variable monotonic increasing function.) Consider a small $\alpha \in \mathbb{R}$ number and the sequence $\{x_{i+1} = x_i + \alpha(g - f(x_i))\}$ where g is some goal value. For a differentiable response function it can be estimated that

$$f(x_{i+1}) - g \approx f(x_i) + \alpha \left. \frac{\partial f}{\partial x} \right|_{x_i} (g - f(x_i)) - g$$

$$f(x_{i+1}) - g \approx \left[I - \alpha \left. \frac{\partial f}{\partial x} \right|_{x_i} \right] (f(x_i) - g) ,$$
(31)

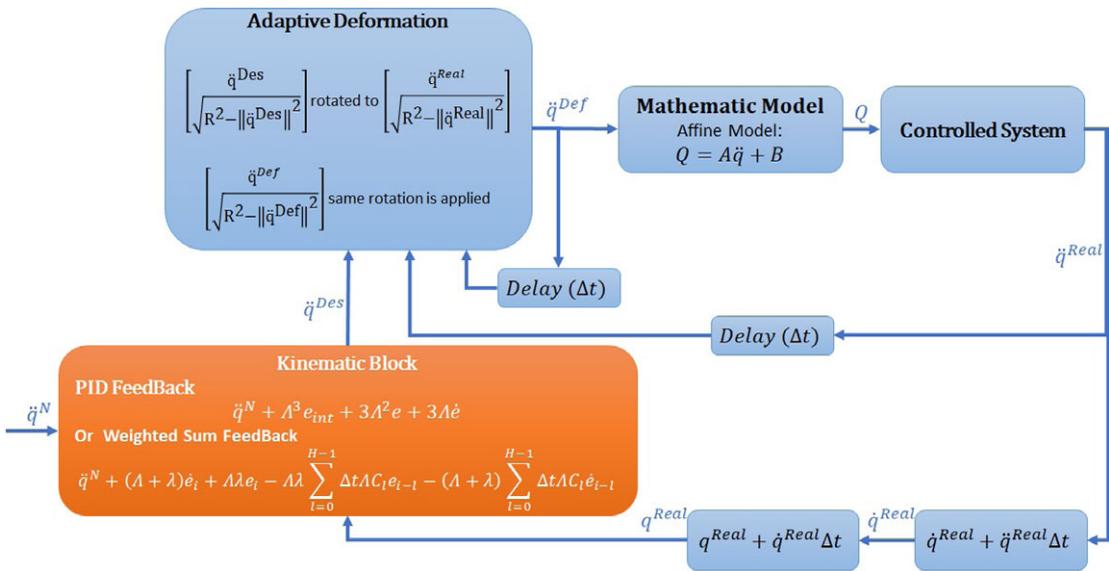


Figure 1. Fixed point iteration-based adaptive control schematics using a particular adaptive deformation method (the Euler integration in the bottom of the figure refers to the fact that the exact integration is done by the “physical operation” of the controlled system); the iterative sequence can be initiated with $\ddot{q}^{Def}(1) = \ddot{q}^{Des}(1)$.

that, if the matrix $I - \alpha \left. \frac{\partial f}{\partial x} \right|_{x_i}$ decreases the norm of the vectors iteratively will result in $x_i \rightarrow x_*, f(x_*) = g$. For an arbitrary array w , it can be written that

$$\left\| \left[I - \alpha \left. \frac{\partial f}{\partial x} \right|_{x_i} \right] w \right\|^2 = w^T w - \alpha w^T \left[\left(\left. \frac{\partial f}{\partial x} \right|_{x_i} \right)^T + \left. \frac{\partial f}{\partial x} \right|_{x_i} \right] w + \alpha^2 w^T \left(\left. \frac{\partial f}{\partial x} \right|_{x_i} \right)^T \left. \frac{\partial f}{\partial x} \right|_{x_i} w, \quad (32)$$

in which for a finite partial derivative and a very small $\alpha > 0$ the third term can be neglected, and for an approximate direction-keeping function, the second term is negative, that is the distance between the actual response and the goal can be step by step decreased. In the case of robots, for instance, the existence of the positive definite inertia matrix can guarantee the approximate direction-keeping nature of the response function. Figure 1 describes how this method is inserted into the control scheme of the Computed Torque Control for robots (i.e., for second-order systems) so that during one digital control step only one step of the above adaptive iteration can be done, while the goal value $g(t) := \ddot{q}^{Des}(t)$ slowly drifts in time. According to (32), the speed of convergence of this iteration depends on the value of α , and in the lack of information on the exact model parameters, the method experimentally can be installed and investigated.

Equation (32) indicates that the iteration can turn into a divergent one if the parameter α is too big, while the convergence becomes too slow if it is too small. In both cases, the quality of the adaptive control is corrupted. Over the years, various transformation functions were proposed for use in the block “Adaptive Deformation” (e.g., refs. [62–64]) and the solution was applied in a Model Reference Adaptive Control scenario as well [65]. For making it more easy to find an appropriate α , the linear iteration outlined in (31) was replaced by a non-linear one in ref. [66] in the following simple manner: by choosing a big value $0 < R \in \mathbb{R}$ so that $\|\ddot{q}^{Real} \in \mathbb{R}^n\|, \|\ddot{q}^{Des} \in \mathbb{R}^n\|, \|\ddot{q}^{Def} \in \mathbb{R}^n\| \ll R$, these vectors were so augmented by the use of a “complementary, physically not interpreted dimension” that they obtained the common “augmented norm” R . Consequently, the $\mathcal{A} := [\ddot{q}^{Des}; D^{Des}]$, $\mathcal{B} := [\ddot{q}^{Real}; D^{Real}]$ and $\mathcal{C} := [\ddot{q}^{Def}; D^{Def}] \in \mathbb{R}^{n+1}$ vectors can be rotated into each other in \mathbb{R}^{n+1} , and it is easy to construct the orthogonal matrix that rotates $\mathcal{B}(i)$ to $v^{Des}(i + 1)$ so that the orthogonal subspace of these vectors remains

invariant. By interpolating the angle of rotation by a factor $\lambda_a \in (0, 1)$, the physically interpreted projections of the vectors will not be completely identical: they will only approach each other. The next deformed value $C(i + 1)$ will be created from $C(i)$ with this rotation of limited angle. To sum up, the parameter of the adaptive rotation will be λ_a instead of α in (31). In the practice, it was found that it is more easy to set λ_a than α .

Algorithm 1. Adaptive Deformation using Abstract Rotations [67].

```

Data:  $\epsilon, R, \lambda_a, \ddot{q}_{max}$  scalar values // from the user
Data:  $\ddot{q}^{Def}(t - \delta t), \ddot{q}^{Des}(t), \ddot{q}^S(t - \delta t)$  arrays with  $n$  element // from measurements
Result:  $\ddot{q}^{Def}(t)$ 
1 begin
2 // Ensure that  $\|\mathcal{A}\| = \|\mathcal{B}\| = \|C\|$ 
3  $\mathcal{A} \leftarrow \left[ \ddot{q}^{Des}(t - \delta t) \quad \sqrt{R^2 - \|\ddot{q}^{Des}(t)\|^2} \right]^T$ 
4  $\mathcal{B} \leftarrow \left[ \ddot{q}^S(t - \delta t) \quad \sqrt{R^2 - \|\ddot{q}^S(t - \delta t)\|^2} \right]^T$ 
5  $C \leftarrow \left[ \ddot{q}^{Def}(t - \delta t) \quad \sqrt{R^2 - \|\ddot{q}^{Def}(t - \delta t)\|^2} \right]^T$ 
6  $\mathcal{A}_{\perp\mathcal{B}} \leftarrow \mathcal{A} - \frac{\mathcal{B}^T \mathcal{A}}{\mathcal{B}^T \mathcal{B}} \mathcal{B}$ 
7  $e_{\perp\mathcal{B}} \leftarrow \frac{\mathcal{A}_{\perp\mathcal{B}}}{\epsilon + \|\mathcal{A}_{\perp\mathcal{B}}\|}$  // small  $\epsilon$  value is used to avoid singularity
8  $e_{\mathcal{B}} \leftarrow \frac{\mathcal{B}}{\|\mathcal{B}\|}$ 
9  $\varphi \leftarrow \arcsin\left(\frac{\|\mathcal{A}_{\perp\mathcal{B}}\|}{R}\right)$ 
10  $G \leftarrow e_{\perp\mathcal{B}} e_{\mathcal{B}}^T - e_{\mathcal{B}} e_{\perp\mathcal{B}}^T$ 
11  $O \leftarrow I + G \cdot \sin(\lambda_a \varphi) + G^2 \cdot (1 - \cos(\lambda_a \varphi))$  //  $I$  is a unit array
12  $Transformed \leftarrow OC$ 
13 // Limit the output of the adaptive deformation block
14  $\ddot{q}^{Def}(t) \leftarrow \ddot{q}_{max} \tanh \frac{[Transformed[0] \quad Transformed[1] \quad \dots \quad Transformed[n-1]]}{\ddot{q}_{max}}$ 
15 end

```

The solution can be regarded as a primitive case of machine learning, as in each control cycle, the control force is calculated based on the observation of the behavior of the system in the previous control cycle, although this information is not used for amending the imperfect dynamic model parameters (in ref. [67] some efforts were made to improve the dynamic model as well, by combining the FPI controller with Particle Swarm Optimization). Also, it can be noted that this approach can be regarded as a special version of the “Data Driven Modeling Technology” (e.g., refs. [68, 69]) in which the usual regression-based approach is replaced with the adaptive deformation.

This approach has the following interesting properties:

- Since it applies Euler integration, the precision of the computations with the PID-type feedback terms is improved by the refinement of the discrete-time-resolution in the usual manner.
- However, if a time interval of given duration is tackled with a finer time resolution, with the increase of the number of the digital control cycles, the adaptive iteration obtains a possibility for the realization of more adaptive iterative steps during unit time, so it is expected that the method’s precision increases with this refinement.

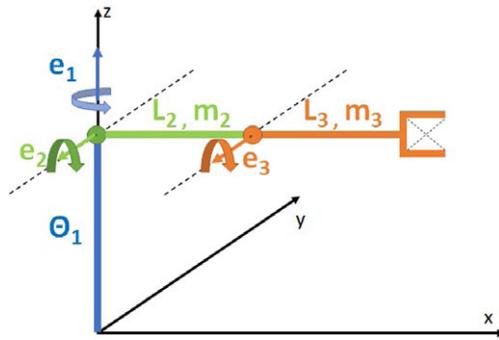


Figure 2. Simplified model of a 3-DoF Puma-type robot arm.

- By feeding back higher-order derivative, its noise sensitivity may be stronger than that of the non-adaptive case: the application of stronger noise filtering may decrease the quality of precision due to its delaying effects. This property is a common feature of the Acceleration Feedback Controllers (e.g., ref. [70]).

5. Simulation-based analysis

As a continuation of our previous research in replacing a fractional order controller with a simple weighted sum feedback design (first efforts were introduced in ref. [29]), further simulation-based results are shown in this section, on a simple model of a 3 Degree of Freedom (DoF) PUMA-type robot arm. The simplified schematic of the robotic arm is shown in Fig. 2. In this section, a PID (Eq. (5)) and the originally proposed weighted sum (WS) feedback (Eq. (12)) are compared to the improved version of the weighted sum feedback (WSPD) (Eq. (30)) proposed in this paper, through a trajectory tracking application in a FPI-based adaptive control scenario.

5.1. Control design for simulations

For the trajectory tracking applications, the nominal trajectory (q^N) was generated in joint space as a sinusoidal function with increasing amplitude. Our simulations were made according to the following considerations:

- **Mathematical Model:** In order to imitate modeling deficiencies, an “approximate” and an “exact” model parameter set is introduced (see in Table I). The necessary control force (Q) was calculated using the approximate parameter set and the system response, which would be measured in a real application, was calculated using the exact model parameters. The dynamic equations of the investigated system ($C_x = \cos q_x, S_x = \sin q_x, C_{xy} = \cos (q_x + q_y), S_{xy} = \sin (q_x + q_y)$):

$$\begin{aligned}
 Q_1 = & \left(\Theta_1 + \frac{1}{4}m_2L_2^2C_2^2 + \frac{1}{4}m_3L_3^2C_{23}^2 + m_3L_2^2C_2^2 + \frac{1}{2}m_3L_2L_3C_{23}C_2 \right) \ddot{q}_1 + \\
 & + \left(-\frac{1}{2}m_2L_2^2C_2S_2\dot{q}_2 - \frac{1}{2}m_3L_3^2C_{23}S_{23}(\dot{q}_2 + \dot{q}_3) - 2m_3L_2^2C_2S_2\dot{q}_2 \right) \dot{q}_1 + \\
 & + \left(-\frac{1}{2}m_3L_2L_3S_{23}C_2(\dot{q}_2 + \dot{q}_3) - \frac{1}{2}m_3L_2L_3C_{23}S_2\dot{q}_2 \right) \dot{q}_1, \tag{33a}
 \end{aligned}$$

Table I. *Modelling parameters.*

Parameter designation	Approximate model	Exact model
Momentum of inertia for Link 1 (Θ_1) (kg m ²)	60	50
Link 2 length (L_2) (m)	1.5	1
Link 3 length (L_3) (m)	2.5	2
Mass 2 length (m_2) (kg)	25	10
Mass 3 length (m_3) (kg)	25	20
Gravitational acceleration (g) ($\frac{m}{s^2}$)	10	9.81

Table II. *Kinematic parameter setting for simulations.*

Parameter designation	PID	WS	WSPD
Error decay constant (Λ) (s ⁻¹)	12	12	12
Derivative error decay constant (λ) (s ⁻¹)	–	–	25
Memory length (H)	–	200	200
Forgetting sharpness (p)	–	2	2

$$\begin{aligned}
 Q_2 = & \left(\frac{1}{4}m_2L_2^2 + \frac{1}{4}m_3L_3^2 + m_3L_2^2 + \frac{1}{2}m_3L_3L_2C_3 \right) \ddot{q}_2 - \frac{1}{2}m_3L_3L_2S_3\dot{q}_3\dot{q}_2 + \\
 & + \left(\frac{1}{4}m_3L_3^2 + \frac{1}{4}m_3L_3L_2C_3 \right) \ddot{q}_3 - \frac{1}{4}m_3L_3L_2S_3\dot{q}_3^2 + \frac{1}{4}m_2L_2^2C_2S_2\dot{q}_1^2 + \\
 & + \left(\frac{1}{4}m_3L_3^2C_{23}S_{23} + m_3L_2^2C_2S_2 + \frac{1}{4}m_3L_2L_3S_{23}C_2 + \frac{1}{4}m_3L_2L_3C_{23}S_2 \right) \dot{q}_1^2 + \\
 & + \frac{1}{2}m_2L_2gC_2 + m_3gL_2C_2 + \frac{1}{2}m_3L_3gC_{23}, \tag{33b}
 \end{aligned}$$

$$\begin{aligned}
 Q_3 = & \left[\frac{1}{4}m_3L_3^2 + \frac{1}{4}m_3L_3L_2C_3 \right] \ddot{q}_2 + \frac{1}{4}m_3L_3^2\ddot{q}_3 + \frac{1}{4}m_3L_3^2C_{23}S_{23}\dot{q}_1^2 + \\
 & + \frac{1}{4}m_3L_3L_2S_{23}C_2\dot{q}_1^2 + \frac{1}{4}m_3L_3L_2S_3\dot{q}_2^2 + \frac{1}{2}m_3gL_3C_{23}. \tag{33c}
 \end{aligned}$$

- Adaptive Deformation: All three compared feedback solutions are implemented in a Fixed Point Iteration-based control scenario, in which convergent iteration is calculated using Abstract Rotations [66]. The essence of this solution is given in Section 4. The adaptive parameter $\lambda_a = 0.06$ and the initial condition is $\ddot{q}^{Des}(1) = \ddot{q}^{Def}(1)$.
- Kinematic Block: The parameter settings for all three kinematic feedback designs are summarized in Table II. The parameter settings were kept the same throughout our simulations. For the weighted sum-type feedback solutions, the forgetting nature of the solution was characterized as

$$C_\ell = ((\epsilon + \ell)\delta t)^{-p}, \tag{34}$$

where $\epsilon = 0.1$ and $\delta t = 0.001$ s, which is the control cycle time, finally p is the forgetting sharpness which can arbitrarily set.

Simulations were made in Julia programming language on an HP 250 G6 computer under Windows10 operating system.

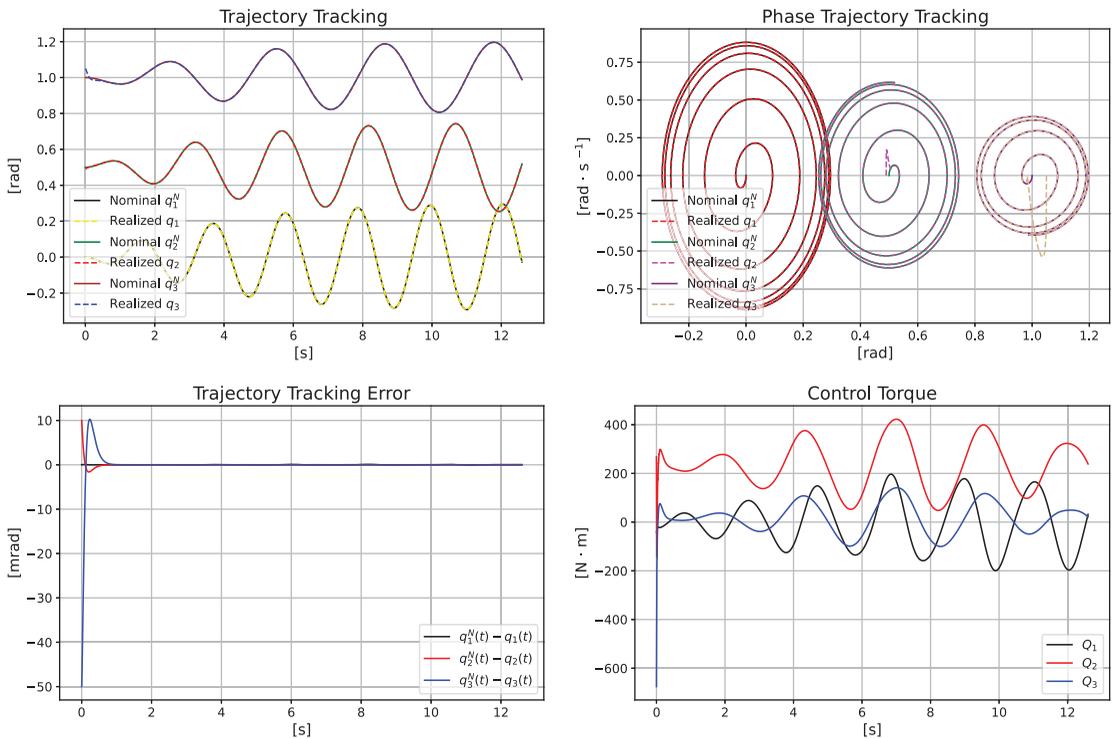


Figure 3. Simulation results for fixed point iteration-based adaptive control with PID feedback ($\Lambda = 12 \text{ s}^{-1}$, $\lambda_a = 0.06$).

5.2. Simulation results

In the first set of simulations (Figs. 3–5), the three kinematic feedback designs were compared in an FPI-based adaptive control scenario. Figure 3 gives us a good basis for comparison, as in this case, the kinematic block was implemented using PID-type feedback. This solution generated a considerable overshoot at the beginning of the control, however later the modeling errors were nicely compensated through the adaptive deformation which yields a very low (less than 0.5 mrad) trajectory tracking error in the steady state (after the initial error is compensated). Figure 4 shows the simulation results for a weighted sum-type feedback in kinematic design. In this simulation, the initial overshoot of the system was completely eliminated and even lower trajectory tracking error was achieved in the steady state. However, a very drastic control action can be observed in the initial stage of the control. On the other hand, in Fig. 5, the excessively high control forces in the initial stage of the control were avoided and still very nice asymptotic error convergence was achieved.

In the second set of experiments, no adaptive deformation was applied but all kinematic parameters were kept the same as before. This simulation exhibits the advantage of the newly proposed weighted sum PD-type feedback, as in Fig. 6, using a simple weighted sum solution, a serious chattering can be observed in the control forces which could not be implemented on a real system. This effect was nicely smoothed by the adaptive deformation so the chattering effect did not appear in the previous results. On the other hand, the proposed solution produced suitable control forces without the adaptive deformation and nice tracking was achieved obviously with higher tracking errors due to modeling imprecisions (Fig. 7).

In the last simulation, which is presented in Fig. 8, the proposed weighted sum PD-type feedback solution was tested with “noisy signal measurements”. In the simulation, the feedback terms/signals were loaded with Gaussian noise ($\sigma = 10^{-6}$ rad) and a 3rd-order low-pass filter was applied as,

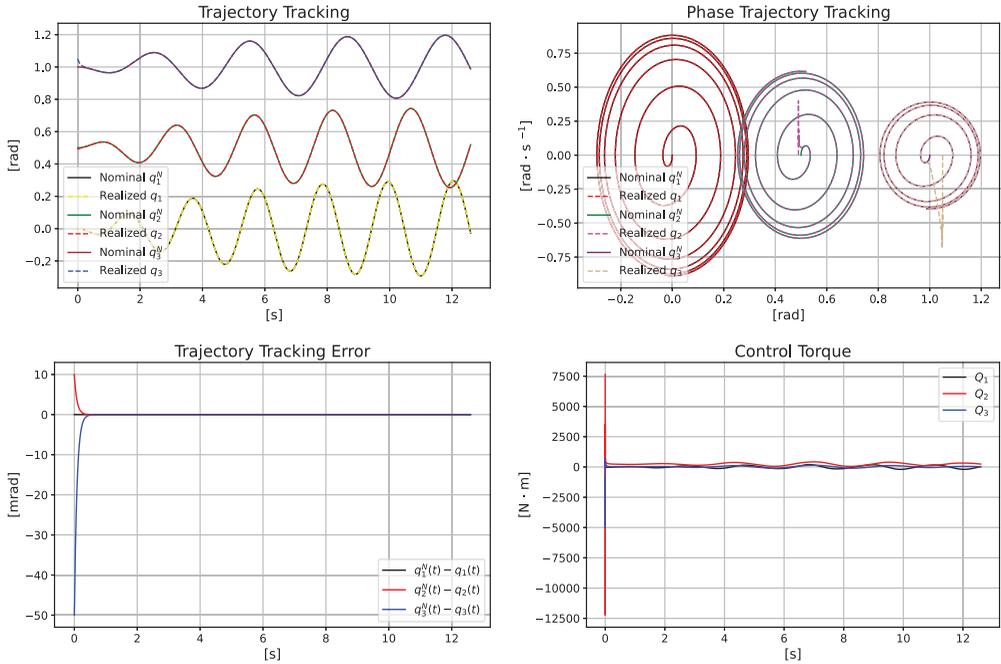


Figure 4. Simulation results for fixed point iteration-based adaptive control with weighted sum feedback ($H = 200$, $p = 2$, $\Lambda = 12 \text{ s}^{-1}$, $\lambda_a = 0.06$).

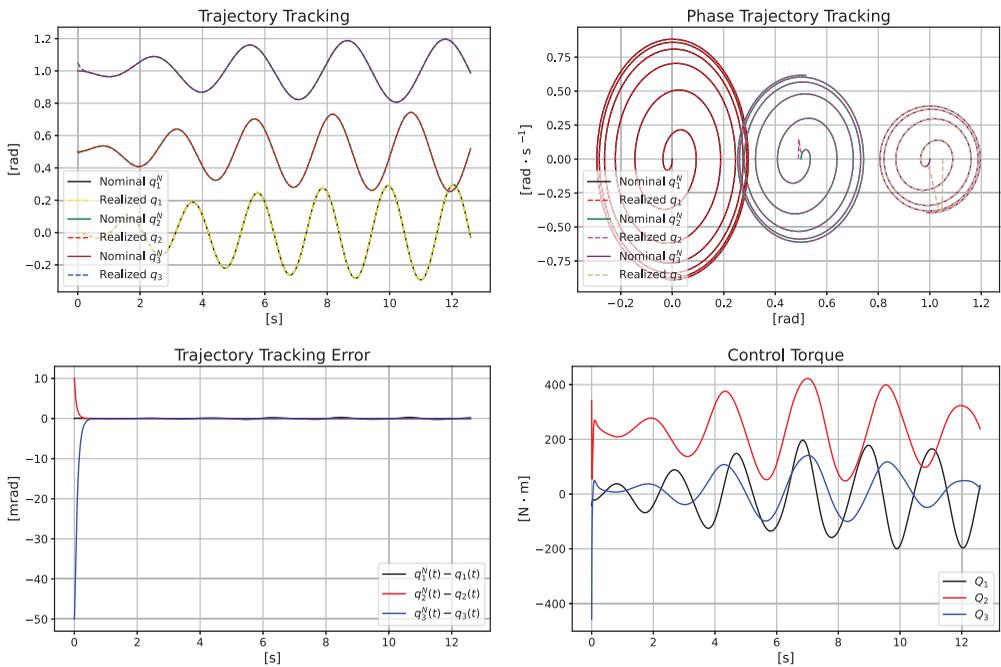


Figure 5. Simulation results for fixed point iteration-based adaptive control with the proposed weighted sum PD-type feedback ($H = 200$, $p = 2$, $\Lambda = 12 \text{ s}^{-1}$, $\lambda = 25 \text{ s}^{-1}$, $\lambda_a = 0.06$).

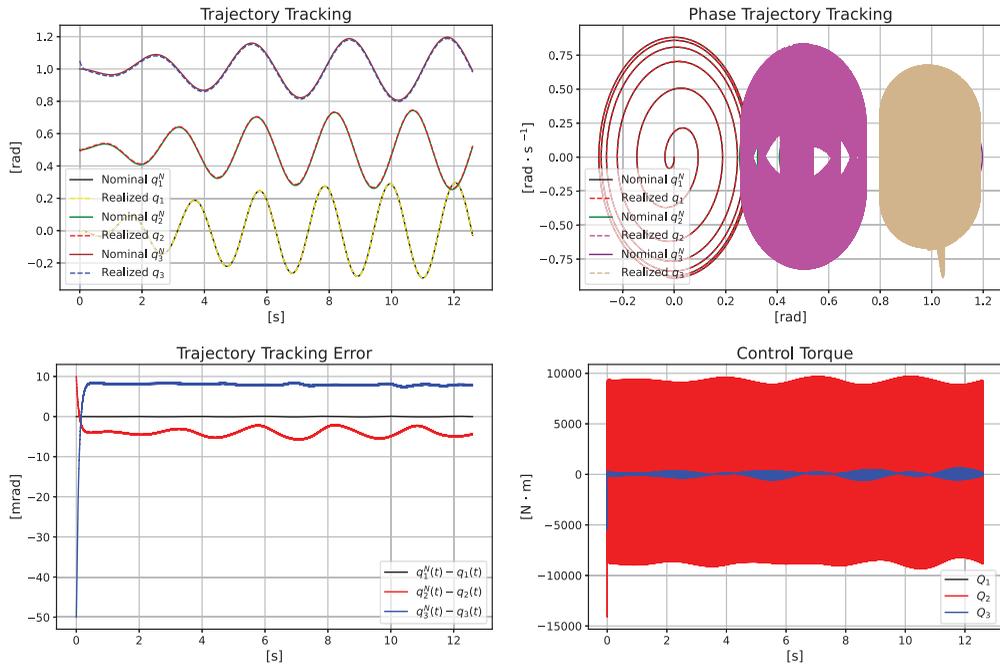


Figure 6. Simulation results for weighted sum feedback without adaptive deformation ($H = 200$, $p = 2$, $\Lambda = 12 \text{ s}^{-1}$, $\lambda_a = 0.06$).

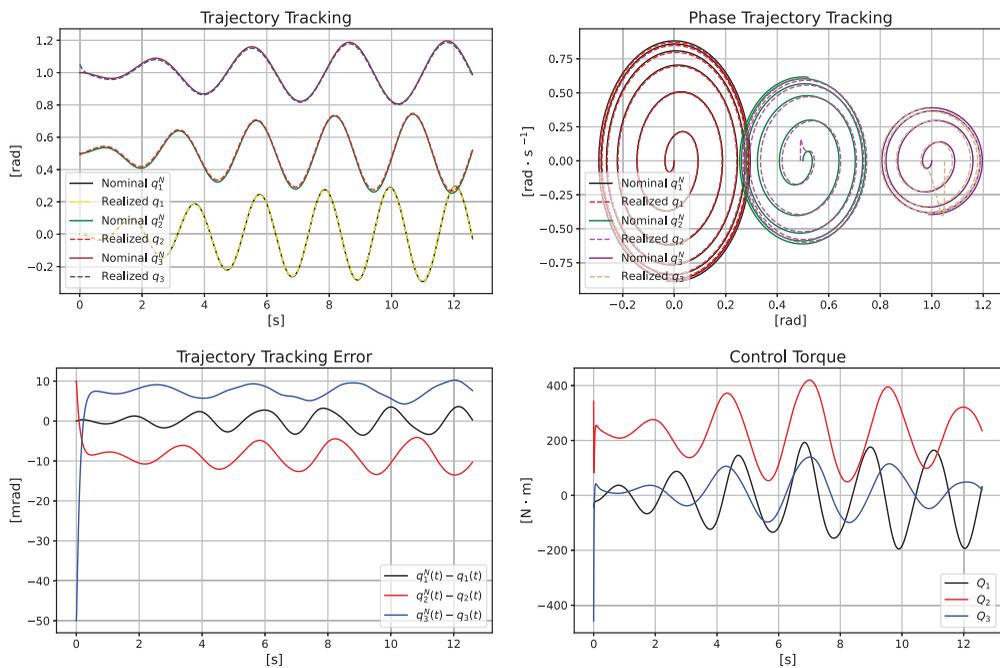


Figure 7. Simulation results for the proposed weighted sum PD-type feedback with no adaptive deformation ($H = 200$, $p = 2$, $\Lambda = 12 \text{ s}^{-1}$, $\lambda = 25 \text{ s}^{-1}$, $\lambda_a = 0.06$).

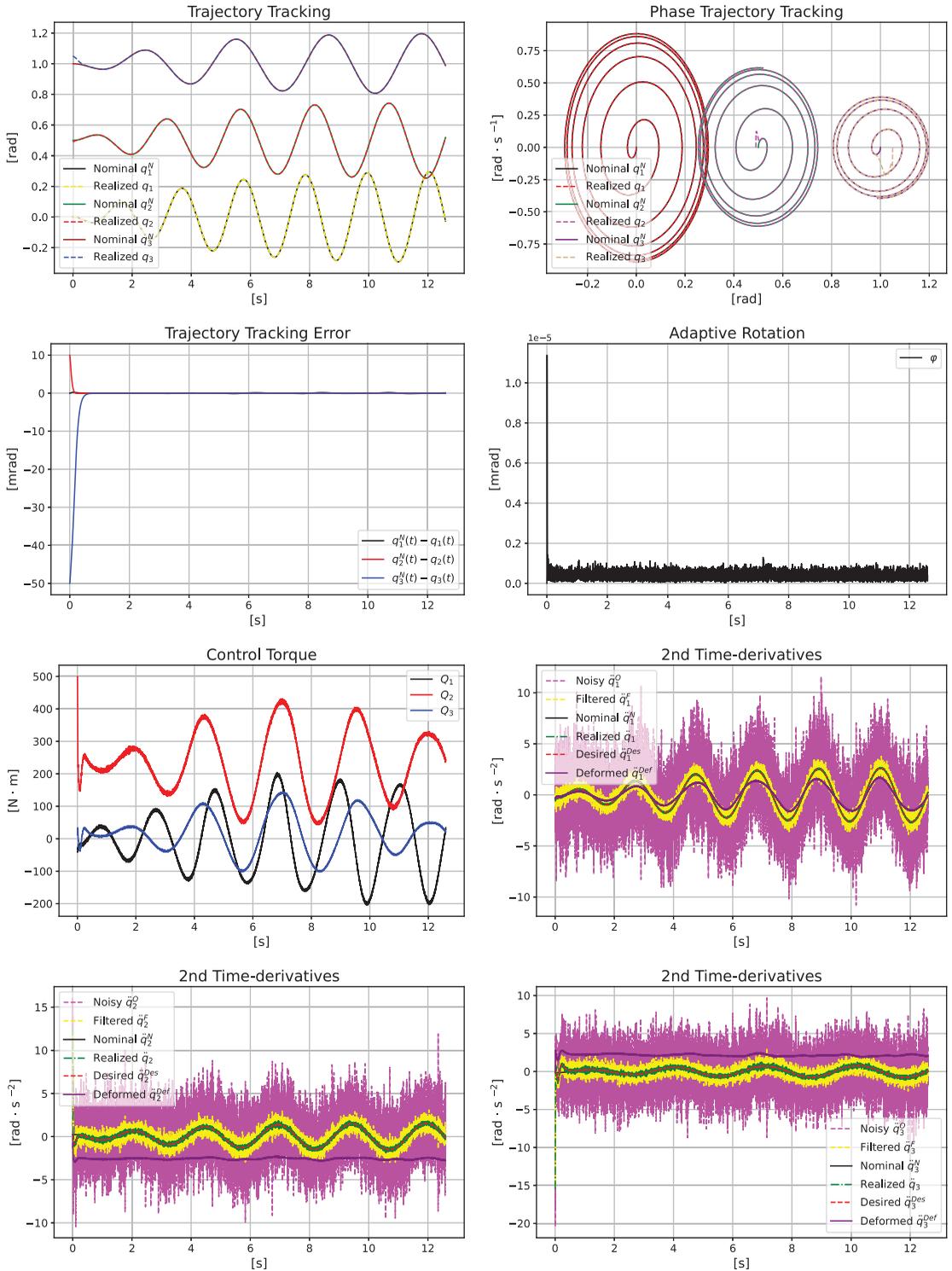


Figure 8. Simulation results for 3-DoF PUMA-type robot arm using fixed point iteration-based adaptive control with the proposed weighted sum PD-type feedback with Gaussian noise ($\sigma = 10^{-6}$ rad) on the feedback ($H = 200$, $p = 2$, $\Lambda = 12 \text{ s}^{-1}$, $\lambda = 25 \text{ s}^{-1}$, $\lambda_a = 0.06$).

$$\left(\Lambda_f + \frac{d}{dt} \right)^3 q^s(t) = \Lambda_f^3 q^o(t), \tag{35a}$$

$$\ddot{q}^s(t) = \Lambda_f^3(q^o(t) - q^s(t)) - 3\Lambda_f^2\dot{q}^s(t) - 3\Lambda_f\ddot{q}^s(t), \tag{35b}$$

where $q^s(t)$ is the filtered signal and $q^o(t)$ is the observed noisy signal, from $\ddot{q}^s(t)$ the $\ddot{q}^s(t)$, $\dot{q}^s(t)$, $q^s(t)$ values were calculated using Euler integration. In Fig. 8, the effect of the applied noise filtering technique can be well seen as there is a significant difference between observed $\ddot{q}^o(t) \approx (q^o(t_{i+1}) - 2q^o(t_i) + q^o(t_{i-1}))/\delta t^2$ and filtered $\ddot{q}^s(t)$ values. In the last simulation to reduce the initial force applied by the controller, the feedback gains in the kinematic block are gradually incremented by function,

$$g = g^N \tanh \frac{S}{S^{\max}}, \tag{36}$$

where g is a particular control gain (Λ or λ), g^N is the nominal value of the control gain, S is the current control step and S^{\max} is the number of steps to reach the nominal gain (simulation setting $S^{\max} = 120$). This technique could reduce the joint torques in the beginning of the control (most significant reduction was achieved in case of joint 2), the effect is also visible on the \ddot{q} values. Figure 8 shows that with noisy feedback signal still high precision trajectory tracking could be achieved. The results also nicely exhibits properties of an FPI-based adaptive control solution, as in the figures the deformed values of the second derivatives of the joint coordinates (\ddot{q}^{Def}) are significantly different from the desired values (\ddot{q}^{Des}) which are calculated purely on kinematic basis. This is also indicated by the angle of abstract rotation which is a good measure of the adaptive deformation.

The proposed solution may be applicable in various practical fields, especially in life sciences [71, 72], fluid tank control [73] or in robotics where control robustness and fast asymptotic error convergence is essential.

6. Experimental analysis

To further validate the effectiveness of the proposed control method, a simple experimental setup is presented in this section. The control method was evaluated through a DC motor control application with time varying load.

The experimental setup consisted of a 12 V DC motor (type FIT0185 [74]) with an inbuilt encoder and planetary gearbox with 131:1 reduction ratio. The electric motor was driven with a BTS7960-M [75] dual half bridge motor drive.

The output shaft was connected to a spring through a special coupling which provided a time varying load in our system. The loading torque (Q_L) applied by the spring on the output shaft can be calculated using Lagrangian-equation and Euler-Lagrange Equation of motion as

$$Q_L = \theta_w \ddot{q} + D_s l_e l_t \sin q \left(1 - \frac{l_{r0}}{\sqrt{l_e^2 + l_t^2 - 2l_e l_t \cos q}} \right), \tag{37}$$

where θ_w is the inertia of the coupling, D_s is the spring constant, l_e denotes the lever on which the loading torque of the spring is applied, l_t is the distance between the fixed mounting point of the spring and the motor shaft and finally l_{r0} is the length of the spring in the initial position of the motor (shown in Fig. 9).

The position of the output shaft was measured through the inbuilt incremental encoder, which has a relatively low resolution (16 CPR – Counts per Rotations – for the motor shaft). However, the position of the output shaft could be precisely measured, as the encoder resolution for the output shaft is 2096 CPR, due to the high reduction ratio of the planetary gearbox.

In our experimental setup, the control algorithm was written in C++ and running on an Arduino DUE Board with Atmel SAM3X8E ARM Cortex-M3 32-bit processor. The adaptive deformation was applied for the PWM (Pulse Width Modulation) output of the controller (see Fig. 10). All measurement data were sent to a PC with a custom-made monitoring software, which allowed us to display multiple

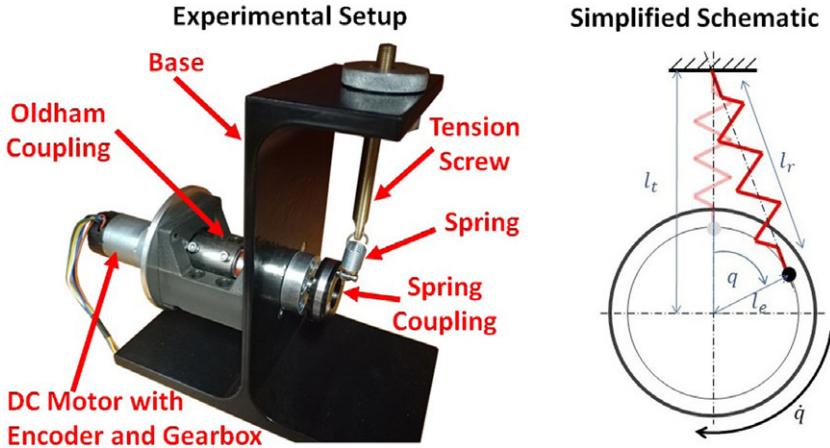


Figure 9. Mechanical design of the experimental setup (left) and simplified dynamic model (right).

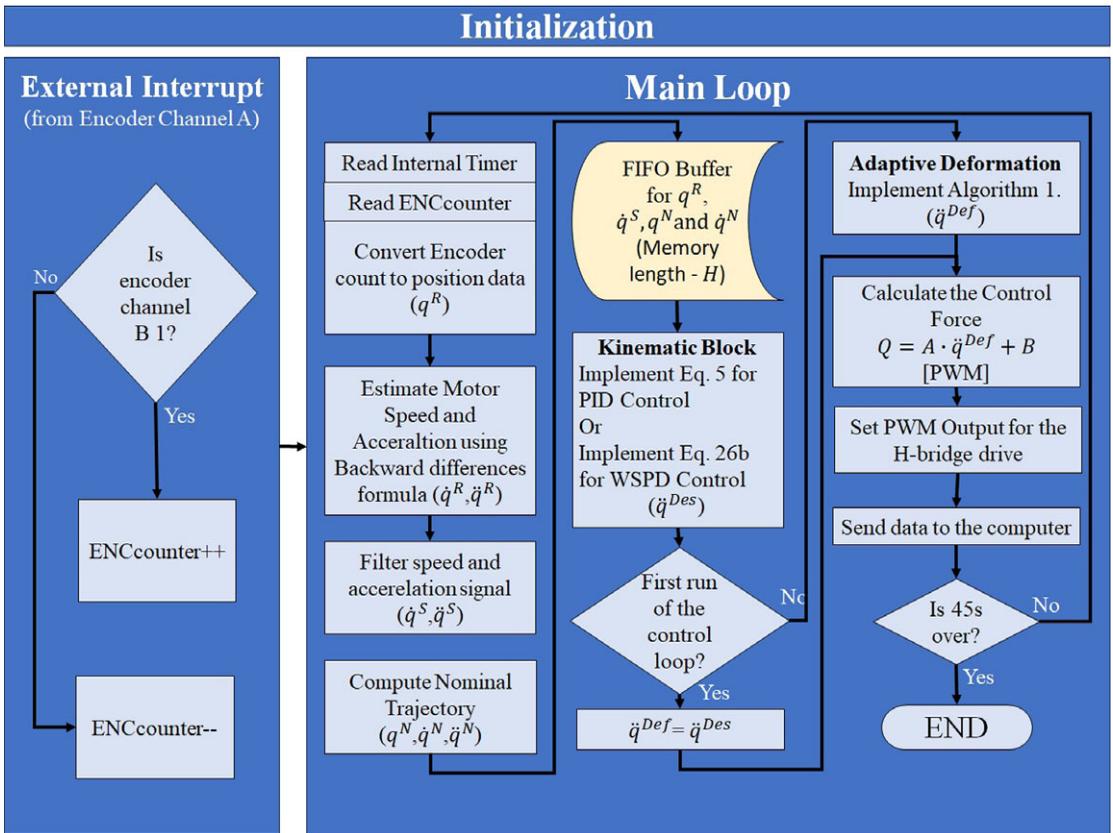


Figure 10. Program flow chart for the two different control solutions. In case of WSPD controller, the FIFO buffer with appropriate memory length (H) is used (short-term memory of the proposed control solution), for PID control $H = 2$, only the current and the data from the previous control cycle is used to do the backward difference estimation for the $\dot{q}^R(t)$ and $\ddot{q}^R(t)$ signals.

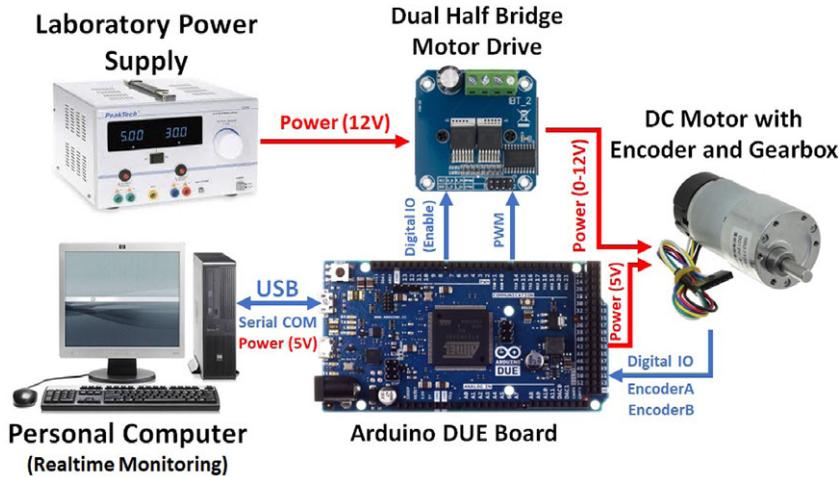


Figure 11. Schematic of the experimental setup.

data in real-time. The measurement results were saved in *.csv* (comma separated values) file and then processed in a separate script. The schematic of the full experimental setup is given in Fig. 11.

6.1. Control design for the experimental setup

The position feedback of the output shaft was obtained through the inbuilt magnetic quadrature encoder of the motor. The speed measurement (\dot{q}^R) was characterized by considerable noise component due to quantization error (which is typical for low-resolution encoders). In order to reduce effects of the measurement noise, a simple Infinite Impulse Response low pass filter was implemented in our control algorithm. The cutoff frequency of the low pass filter was selected as $f_c = 15\text{Hz}$, which resulted in good noise attenuation without significant delay. The transfer function of the digital low-pass filter was obtained through bilinear transformation [76].

The control design procedure was divided into three parts according to Fig. 1.

- **Mathematical Model:** In Fixed Point Iteration-based adaptive control methods, the necessary control forces (Q) are generated from the adaptively deformed \ddot{q}^{Def} value using an “available” dynamic model of the controlled system which is typically imprecise. However, in ref. [77], “quasi” model-free approach was introduced which utilized a very simple affine model as $Q = A\ddot{q}^{Def} + B$ to control an under-actuated system where not all state variables could be measured. The affine model parameters A, B could be arbitrarily set by the user and it was shown through simulations that the parameters could be set in wide range without significantly compromising the control performance. To further simplify the control approach and reduce the amount of “tunable” parameters, in our experiments, the adaptive deformation was directly applied on the PWM output (Q_{PWM}) of the controller, essentially $\ddot{q}^{Def} \equiv Q_{PWM}$.
- **Adaptive Deformation:** Throughout the trajectory tracking applications, in our experiments the same adaptive deformation block was applied, which was implemented using abstract rotations [66]. The essence of this solution is given in Section 4. The adaptive parameter was $\lambda_a = 1$ and the initial condition was $\ddot{q}^{Des}(1) = \ddot{q}^{Def}(1)$.
- **Kinematic Prescription:** The kinematic prescription can be formulated in various manners. The most commonly used solution is a PID-type feedback, which will be our baseline for comparison. For the PID-type feedback, the desired value of the 2nd derivative of the angular position (\ddot{q}^{Des}) was calculated according Eq. (4). For the proposed WSPD feedback in Eq. (30), each weight

Table III. Transient response analysis (steady-state error (e_{ss} [rad]), percentage overshoot ($\sigma_{\%}$ [%]) and settling time (t_s [s])), kinematic block parameters (Λ [s^{-1}], λ [s^{-1}]).

	Set point 1 (45°)			Set point 2 (90°)			Set point 3 (-90°)		
	t_s	e_{ss}	σ	t_s	e_{ss}	σ	t_s	e_{ss}	σ
PID ($\Lambda = 20$)	0.58	0.009	66.8	0.58	-0.006	67.6	2.45	0.006	94.4
PID ($\Lambda = 15$)	0.78	-0.006	60.7	0.9	0.006	53.4	1.76	0.0	81.8
PID ($\Lambda = 12$)	3.02	0.018	51.5	2.02	0.0	56.1	2.26	-0.015	73.6
WSPD ($\Lambda = 25, \lambda = 95$)	0.21	-0.006	0.8	0.21	-0.003	0.8	0.46	0.003	0.12
WSPD ($\Lambda = 60, \lambda = 85$)	0.25	0.006	8.8	0.25	0.003	9.2	0.55	-0.006	4.8
WSPD ($\Lambda = 12, \lambda = 75$)	0.27	-0.006	1.1	0.27	-0.012	1.5	0.55	0.006	0.3

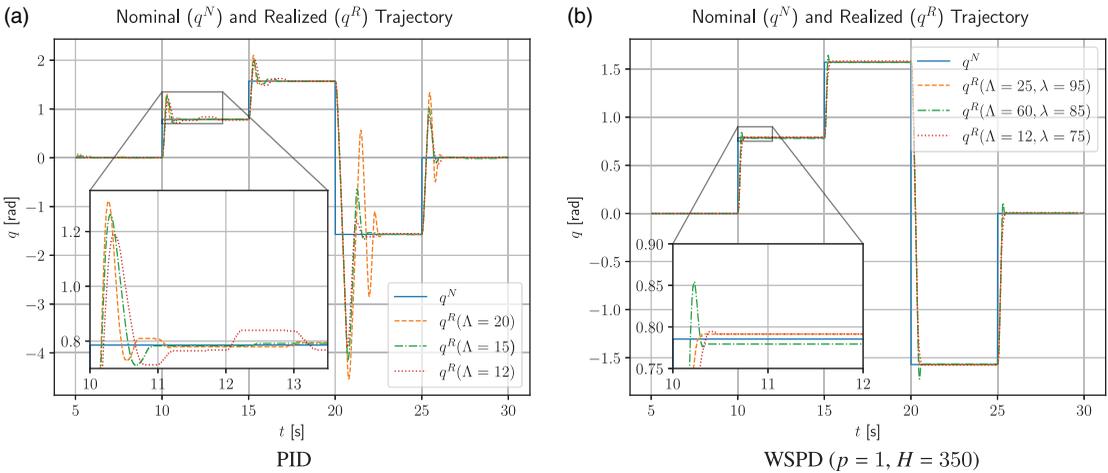


Figure 12. Measurement results for set-point tracking with multiple gain settings (Λ [s^{-1}], λ [s^{-1}]) with no adaptive deformation and no load on the motor.

coefficient was set as $C_t = ((\epsilon + \ell)\delta t)^{-p}$, where $\epsilon = 0.1$, δt is the cycle time of the controller (or sampling time) and p was appropriately set to achieve sufficient control performance. The memory length for all of our experiments is $H = 350$.

6.2. Set-point tracking

In the first set of experiments, the motor was driven through multiple set points with no adaptive deformation, which provides a good basis for comparison of the transient behavior of different control solutions. The quality of the control was characterized by the steady-state error (e_{ss} [rad]), the percentage overshoot ($\sigma_{\%}$ [%]) and the settling time (t_s [s]), shown in Table III. The settling time was measured for 0.05 rad error band. The measurement results are given in Fig. 12 for PID controller on the left and for the proposed WSPD on the right. The experiments were carried out for multiple gain settings for both control solutions.

The measurement result meets our preliminary expectations. In case of PID controller, higher Λ values yield lower steady-state error and faster control response, however, induce more oscillations and more significant overshoots, especially for higher jumps in the set point (e.g., set-point 3). Furthermore, more oscillations can compromise settling time as well. On the other hand, low Λ values produce slow control responses, which yield high settling times as well and also higher steady-state errors. In our

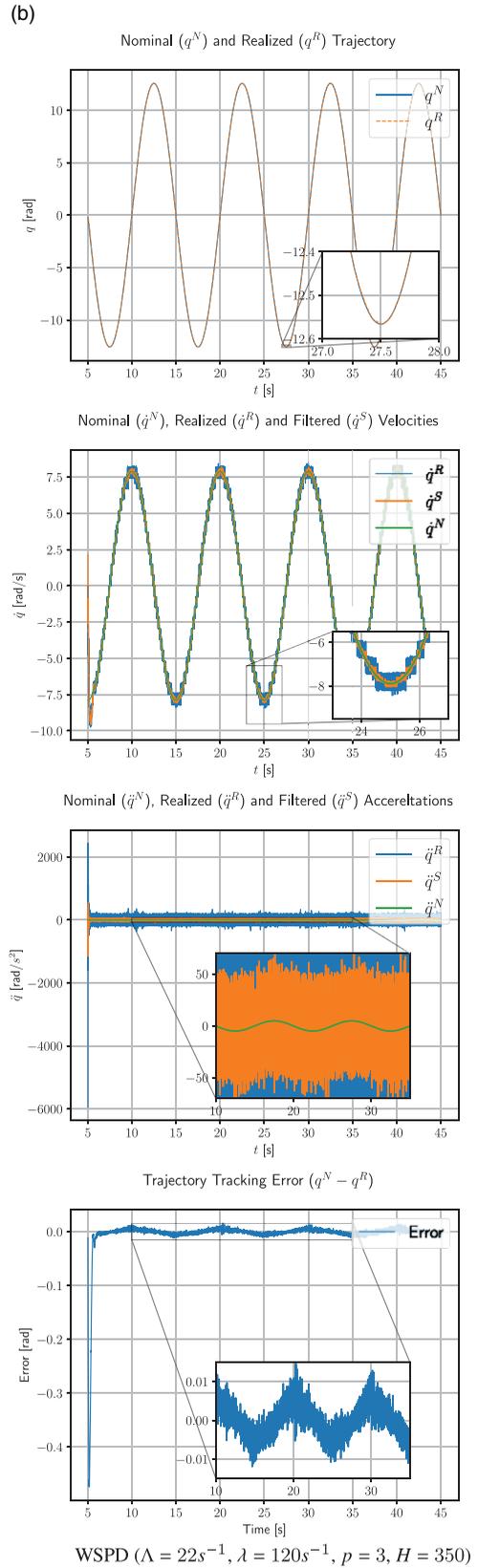
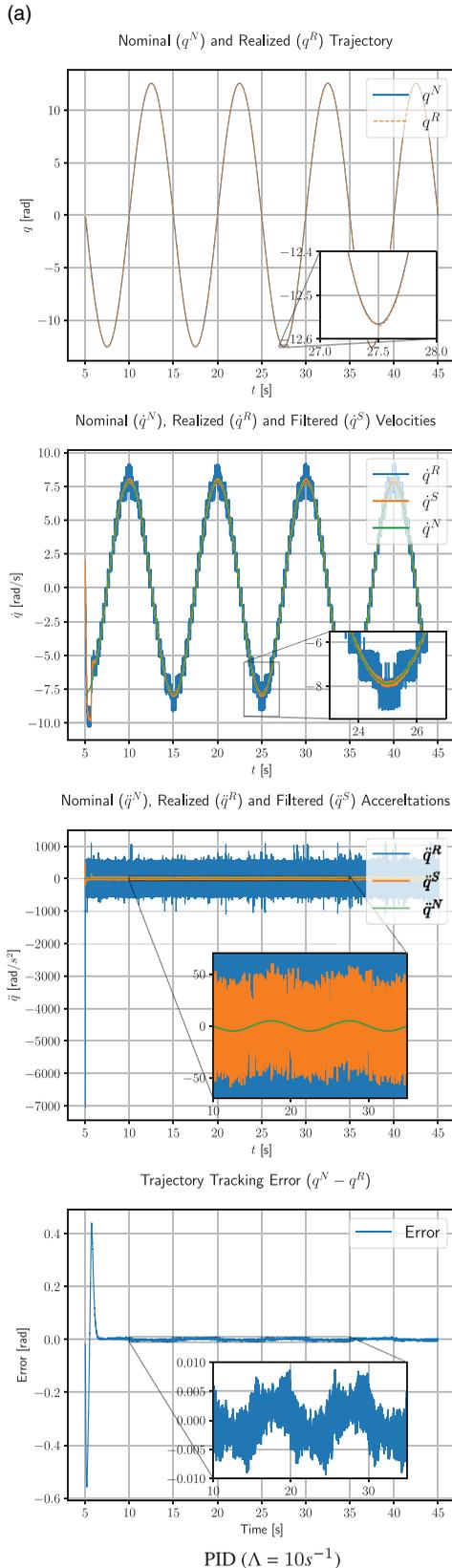


Figure 13. Measurement results with adaptive deformation and no load on the motor.

Table IV. Trajectory tracking comparison (maximum absolute error – e_{\max} [rad], average absolute tracking error – μ_e [rad], standard deviation of the trajectory tracking error – σ_e).

	With adaptive deformation			Without adaptive deformation		
	e_{\max}	μ_e	σ_e	e_{\max}	μ_e	σ_e
WSPD – No Load	0.0163	0.0034	0.0025	0.1734	0.0903	0.0448
WSPD – Load 1	0.0315	0.0048	0.0042	0.2801	0.1037	0.0846
WSPD – Load 2	0.0173	0.0041	0.0031	0.2516	0.0978	0.066
PID – No Load	0.0093	0.0025	0.0017	0.2598	0.0661	0.0535
PID – Load 1	0.0393	0.0091	0.0068	0.5309	0.2249	0.1495
PID – Load 2	0.0218	0.0051	0.0039	0.3604	0.1418	0.0925

experiments, we found $\Lambda = 15$ as the optimal gain value for set-point tracking as it provided acceptable oscillations and overshoot with a relatively low settling time.

In our experiments, we found that relatively low forgetting speed ($p = 1$) yielded good results for set-point tracking for the proposed WSPD. With appropriate Λ and λ settings, very nice asymptotic error convergence was achieved with no overshoot and without compromising the settling time (fast control response). However, high Λ gain combined with a relatively low λ value could still produce some oscillations with small overshoot (e.g., $\Lambda = 60$ with $\lambda = 85$) but still much less significant than that in case of PID feedback. We found that approximately 1 : 4, Λ to λ ratio (e.g., $\Lambda = 25$, $\lambda = 95$) yielded very nice control response and no oscillations for set-point tracking. All in all, the proposed weighted sum feedback outperformed the simple PID controller for set-point tracking, since it eliminated overshoots which yielded much lower settling times with similar steady-state error.

6.3. Trajectory tracking

In the second set of experiments, the proposed feedback term was tested in a trajectory tracking application, with $q^N = 4\pi \cos(0.2\pi t)$ reference trajectory. Our experiments included multiple scenarios for PID control and the proposed WSPD feedback as well, including experiments with- and without FPI-based adaptive deformation and different loading conditions to test the robustness of the controllers. The control performance indexes for these experiments were:

- Maximum absolute error: $e_{\max} = \max_{i=1,2,\dots,N} (|e(i)|)$;
- Average absolute tracking error: $\mu_e = \frac{1}{N} \sum_{i=1}^N |e(i)|$;
- Standard deviation of the trajectory tracking error: $\sigma_e = \sqrt{\frac{1}{N} \sum_{i=1}^N (|e(i)| - \mu_e)^2}$.

The performance indexes were calculated only for the 10 – 45 s interval, that way eliminating the effects of the transient phase, since it was already observed that the proposed control method can eliminate the oscillations and overshoots at the initial phase of the control.

In Fig. 13, some results are given for FPI-based adaptive controller with PID and WSPD kinematic block. In these experiments, the spring was not mounted on the DC motor, that way no external load was applied to the output shaft. These figures nicely exhibit the effect of the quantization error of the encoder signal, causing considerable noise in the first derivative (\dot{q}^R) and more so in the second derivative (\ddot{q}^R) of the state variable q . The noise was attenuated by the applied low pass filter ($f_c = 15$ Hz) and the filtered signals (\dot{q}^S , \ddot{q}^S) were used to calculate the control response. The control cycle time for the FPI-based adaptive PID controller was $\delta t_{PID} = 0.0015$ s and for the FPI-based adaptive WSPD controller, it was $\delta t_{WSPD} = 0.004$ s due to higher computational requirements. Both controllers yielded precise trajectory tracking, and the average and the standard deviation were near to the resolution of the encoder (2096

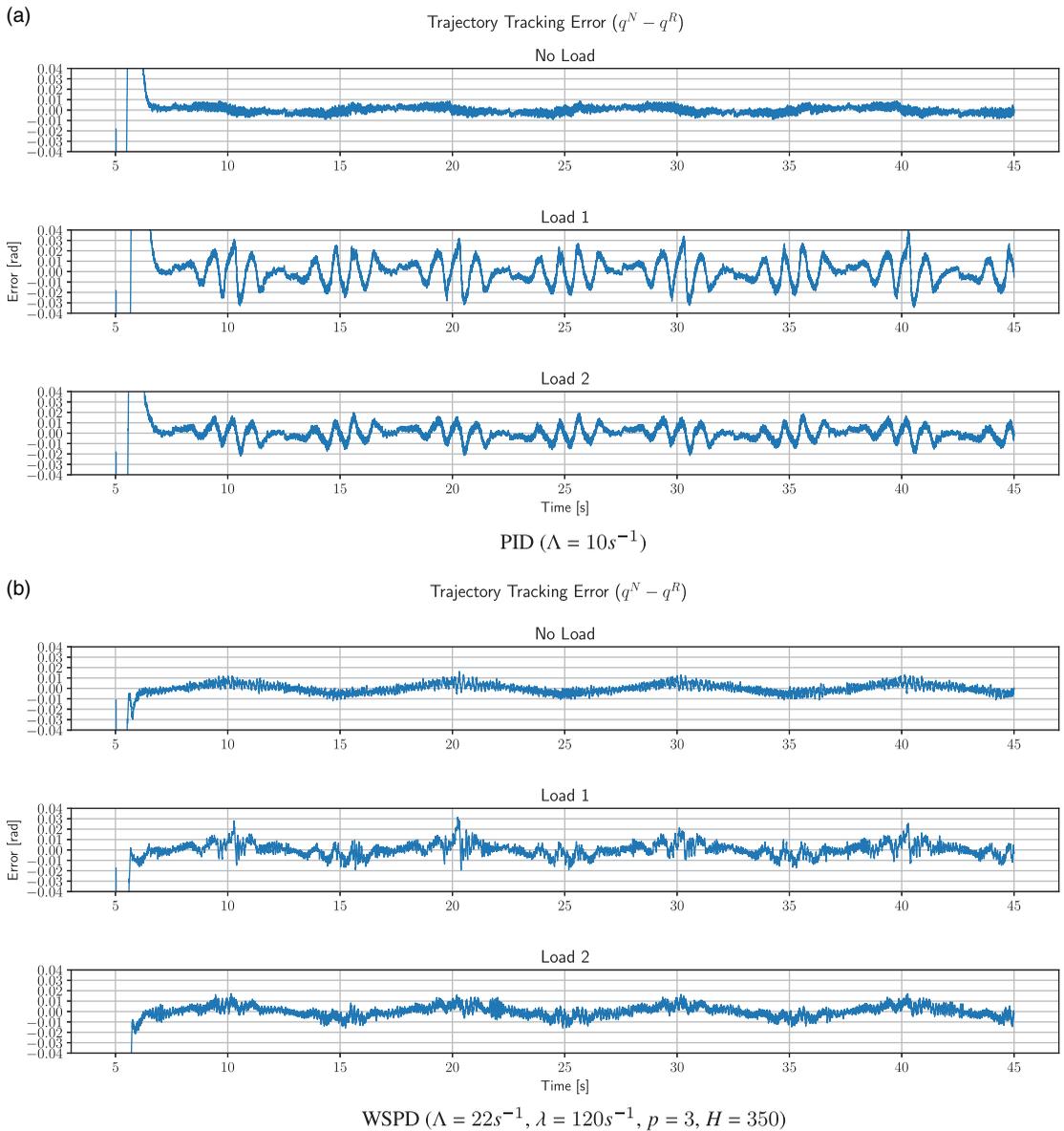


Figure 14. Tracking error for DC motor control with different loads (Load 1 – $D_s = 1.611 \frac{N}{mm}$ Load 2 – $D_s = 0.822 \frac{N}{mm}$) using FPI control, implemented with different kinematic blocks (PID and WSPD).

CPR yields approximately 0.003 rad resolution). The proposed WSPD method in combination with the adaptive deformation could completely eliminate the initial oscillations, however, it is more noise sensitive and produced slightly higher error. In Table IV, PID controller with FPI-based adaptive deformation slightly outperformed the adaptive WSPD controller almost in all aspects when no load was applied.

In the next step, we repeated the same experiments with the spring applied to the shaft. To test the controllers’ robustness, the measurements were made with two different springs (Fig. 14):

- Load 1 – $D_s = 1.611 \frac{N}{mm}$;
- Load 2 – $D_s = 0.822 \frac{N}{mm}$.

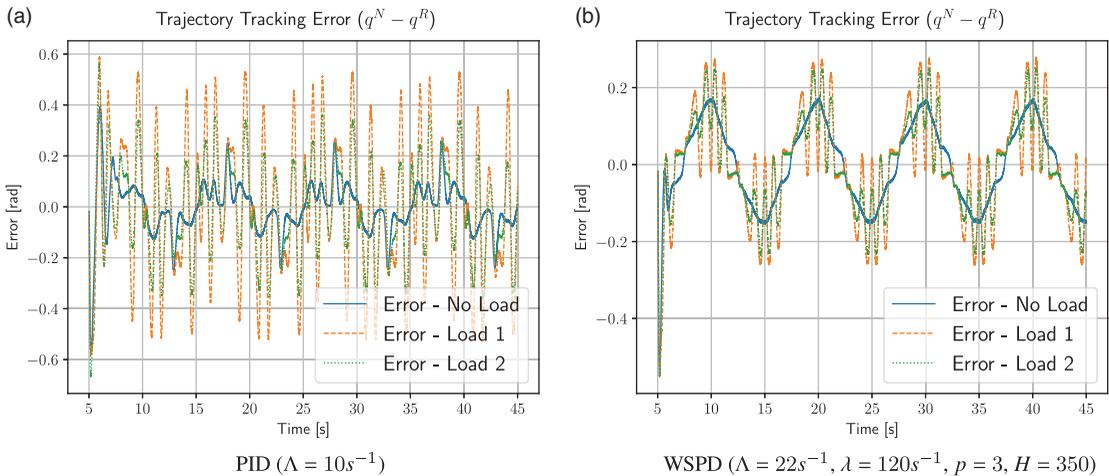


Figure 15. Measurement results for DC motor control without adaptive deformation ($\Lambda[\text{s}^{-1}]$, $\lambda[\text{s}^{-1}]$) under multiple loading conditions (Load 1 – $D_s = 1.611 \frac{\text{N}}{\text{mm}}$ Load 2 – $D_s = 0.822 \frac{\text{N}}{\text{mm}}$).

These results are given in Table IV as well, which revealed that, despite that the adaptive PID controller yielded better average tracking error and standard deviation when no load was applied, the proposed adaptive WSPD controller had better results for both performance indices when the load was applied. The difference in robustness is more significant when no adaptive deformation was applied (Fig. 15).

7. Conclusion

In this paper, a fractional order calculus-inspired weighted sum-type feedback was investigated, which can be used in a fixed point iteration-based control scenario for motion control applications. It is similar to FOC solutions since the proposed feedback term exhibits memory effect, which correlates to better transient response and robust behavior but is exempt from the complex formal restrictions of fractional calculus and has a very simple mathematical structure. This control method does not require the continuous tuning of control gain parameters compared to other adaptive FOC solutions [27], which can be computationally expensive. It also directly aims the decrease the error of certain state variables in an iterative manner, that way the unexpected behavior of the state variables can be avoided in the transient state. The proposed controller can work with a very simple affine mathematical model in a “quasi” model-free approach. The basic idea was introduced in ref. [29], with a simple weighted sum feedback. However, it was shown through simulations that original solution can generate very high control force which can lead to significant chattering in the control output. To avoid this effect the control method was further developed in this paper by incorporating the solution in a PD-type feedback. The proposed control structure is applicable for strongly non-linear systems, where traditional frequency domain-based (e.g., ref. [26]) controller tuning methods are hard to implement.

It was shown on simulation basis that the proposed Weighted Sum PD-type (WSPD) feedback yields very good transient response, similar to its predecessor without generating excessive control force and output chattering. However, it was also revealed that it requires relatively large number of terms (longer memory) and low control cycle time in the weighted sum design. However, it was also shown through the experiments that modern micro-controllers are more than capable to deal with the extra computational demand since relatively low control cycle time was achieved for the weighted sum design ($\delta t_{\text{WSPD}} = 0.004 \text{ s}$), although for the PID-type kinematic block, it was $\delta t_{\text{PID}} = 0.0015 \text{ s}$. The robustness of the WSPD method was also exhibited in this paper since load variations had less effect on the trajectory tracking

error, but it had significantly affected the PID controller for adaptive and non-adaptive cases as well. At the same time in certain control scenarios, the PID controller could provide better control performance (e.g., when no load was applied) than the WSPD method, which originates from two issues. (1) In WSPD method, much higher derivative gain is applied, so the solution is slightly more noise-sensitive. (2) With PID-based kinematic block, lower control cycle time was achieved and that way, more adaptive iteration could be made in the same interval compared to the WSPD method. It is also confirmed by the non-adaptive control scenarios presented in Fig. 15 where the WSPD feedback resulted in better control performance for the unloaded case as well.

The proposed experimental setup served as a “good proof of concept” which has revealed that a fixed point iteration-based control algorithm can be implemented even on relatively low-power equipment. The noise sensitivity of WSPD method can be investigated in the future by applying a more sophisticated filter design. Also, further investigation can be done about the optimal tuning of the proposed Weighted Sum PD-type feedback using some kind of genetic algorithm.

Author contributions. All authors contributed equally to this work.

Financial support. The publication of this paper was supported by the ÚNKP-22-3 New National Excellence Program of the Ministry for Culture and Innovation from the source of the National Research, Development and Innovation Fund. The research reported in this paper is the part of project no. 2019-1.1.1-PIACI-KFI-2019-00303 supported by the Hungarian State.

Competing interests. The authors declare no competing interests exist.

References

- [1] B. Ross, *A Brief History and Exposition of the Fundamental Theory of Fractional Calculus* (Springer Berlin Heidelberg, Berlin, Heidelberg, 1975) pp. 1–36.
- [2] B. Ross, “The development of fractional calculus 1695–1900,” *Hist. Math.* **4**(1), 75–89 (1977).
- [3] E. C. de Oliveira and J. A. T. Machado, “A review of definitions for fractional derivatives and integral,” *Math. Probl. Eng.* **2014**(Article ID 238459), 6 (2014).
- [4] J. T. Machado and V. Kiryakova, “The chronicles of fractional calculus,” *Fract. Calc. Appl. Anal.* **20**(2), 307–336 (2017).
- [5] P. Shah and S. Agashe, “Review of fractional pid controller,” *Mechatronics* **38**, 29–41 (2016).
- [6] D. Copot, R. D. Keyser, J. Juchem and C. Ionescu, “Fractional order impedance model to estimate glucose concentration: In vitro analysis,” *Acta Polytech. Hung.* **14**(1), 207–220 (2017).
- [7] E.-H. Dulf, D. Timis and C.-I. Muresan, “Robust fractional order controllers for distributed systems,” *Acta Polytech. Hung.* **14**(1), 163–176 (2017).
- [8] M. Sasso, G. Palmieri and D. Amodio, “Application of fractional derivative models in linear viscoelastic problems,” *Mech. Time-DEPEND Mater.* **15**(4), 367–387 (2011).
- [9] J. T. Machado and B. Moghaddam, “A robust algorithm for nonlinear variable-order fractional control systems with delay,” *Int. J. Nonlinear Sci. Numer. Simul.* **19**(3), 1–7 (2016).
- [10] C. Monje, B. Vinagre, V. Feliu and Y. Chen, “Tuning and auto-tuning of fractional order controllers for industry applications,” *Control Eng. Pract.* **16**(7), 798–812 (2008).
- [11] D. Baleanu, “Fractional variational principles in action,” *Phys. Scr.* **2009**(T136), 014006 (2009).
- [12] S. Folea, R. De Keyser, I. R. Birs, C. I. Muresan and C. Ionescu, “Discrete-time implementation and experimental validation of a fractional order PD controller for vibration suppression in airplane wings,” *Acta Polytech. Hung.* **14**(1), 191–206 (2017).
- [13] R. E. Gutierrez, J. M. Rosário and J. T. Machado, “Fractional order calculus: Basic concepts and engineering applications,” *Math. Probl. Eng.* **2010**, 1–19 (2010).
- [14] A. Tepljakov, B. B. Alagoz, C. Yeroglu, E. A. Gonzalez, S. H. Hosseinnia, E. Petlenkov, A. Ates and M. Cech, “Towards industrialization of fopid controllers: A survey on milestones of fractional-order control and pathways for future developments,” *IEEE Access* **9**, 21016–21042 (2021).
- [15] P. Lino, G. Maione, S. Stasi, F. Padula and A. Visioli, “Synthesis of fractional-order pi controllers and fractional-order filters for industrial electrical drives,” *IEEE/CAA J. Autom. Sin.* **4**(1), 58–69 (2017).
- [16] S. Kizir and A. Elşavi, “Position-based fractional-order impedance control of a 2 dof serial manipulator,” *Robotica* **39**(9), 1560–1574 (2021).
- [17] B. Deutschmann, C. Ott, C. Monje and C. Balaguer, “Robust Motion Control of a Soft Robotic System Using Fractional Order Control,” *Advances in Service and Industrial Robotics - Proceedings of the 26th International Conference on Robotics in Alpe-Adria-Danube Region (RAAD 2017), June 21-23 2017, Torino, Italy* (C.Ferraresi and G.Quaglia, eds.) (2017) pp. 142–149.

- [18] A. Dumlu and K. Erenturk, "Trajectory tracking control for a 3-DOF parallel manipulator using fractional-order $PI^{\lambda}D^{\mu}$ control," *IEEE Trans. Ind. Electron.* **61**(7), 3417–3426 (2014).
- [19] S. Khubalkar, A. Chopade, A. Junghare, M. Aware and S. Das, "Design and realization of stand-alone digital fractional order pid controller for buck converter fed dc motor," *Circ. Syst. Signal Process.* **35**(6), 2189–2211 (2016).
- [20] I. Podlubny, "Fractional-order systems and fractional-order controllers," *Inst. Exp. Phys. Slovak Acad. Sci. Kosice* **12**(3), 1–18 (1994).
- [21] I. Podlubny, *Fractional Differential Equations: An Introduction to Fractional Derivatives, Fractional Differential Equations, to Methods of Their Solution and Some of Their Applications* (Academic Press, San Diego, 1998).
- [22] A. Oustaloup, F. Levron, B. Mathieu and F. Nanot, "Frequency-band complex noninteger differentiator: Characterization and synthesis," *IEEE Trans. Circ. Syst. I Fund. Theory Appl.* **47**(1), 25–39 (2000).
- [23] D. Xue, C. Zhao and Y. Chen, "A Modified Approximation Method of Fractional Order System," *2006 International Conference on Mechatronics and Automation* (2006) pp. 1043–1048.
- [24] I. Petráš, "Fractional - order feedback control of a dc motor," *J. Electr. Eng.* **60**(3), 117–128 (2009).
- [25] D. Zhou, A. Al-Durra, I. Matraji, A. Ravey and F. Gao, "Online energy management strategy of fuel cell hybrid electric vehicles: A fractional-order extremum seeking method," *IEEE Trans. Ind. Electron.* **65**(8), 6787–6799 (2018).
- [26] E.-H. Dulf, "Simplified fractional order controller design algorithm," *Mathematics* **7**(12), 1166 (2019).
- [27] D. D. Timis, C. I. Muresan and E.-H. Dulf, "Design and experimental results of an adaptive fractional-order controller for a quadrotor," *Fractal Fract.* **6**(4), 204 (2022).
- [28] J. T. Machado, "Analysis and design of fractional-order digital control systems," *Syst. Anal. Model. Simul.* **27**(2-3), 107–122 (1997).
- [29] B. Varga, R. Horváth and J. K. Tar, "Fractional Order Calculus-Inspired Kinematic Design in Adaptive Control," **In: Advances in Service and Industrial Robotics** (A. Müller and M. Brandstötter, eds.) (Springer International Publishing, Cham, 2022) pp. 218–225.
- [30] J. Tar, I. Rudas, A. Dineva and A. Várkonyi-Kóczy, "Stabilization of a Modified Slotine-Li Adaptive Robot Controller by Robust Fixed Point Transformations," *Proceedings of Recent Advances in Intelligent Control, Modelling and Simulation, Cambridge, MA, USA* (2014) pp. 35–40.
- [31] A. Dineva, A. Várkonyi-Kóczy and J. Tar, "Combination of RFPT-based Adaptive Control and Classical Model Identification," *Proceedings of the IEEE 12th International Symposium on Applied Machine Intelligence and Informatics (SAMi 2014), Herl'any, Slovakia* (2014) pp. 35–40.
- [32] J. Tar, J. Bitó, L. Nádai and J. T. Machado, "Robust fixed point transformations in adaptive control using local basin of attraction," *Acta Polytech. Hung.* **6**(1), 21–37 (2009).
- [33] S. Bennett, "A brief history of automatic control," *IEEE Control Syst. Mag.* **16**(3), 17–25 (1996).
- [34] S. Bennett, "Nicholas Minorsky and the automatic steering of ships," *IEEE Control Syst. Mag.* **4**(4), 10–15 (1984).
- [35] B. Armstrong, O. Khatib and J. Burdick, "The Explicit Dynamic Model and Internal Parameters of the PUMA 560 arm," *Proceedings of IEEE Conference on Robotics and Automation 1986* (1986) pp. 510–518.
- [36] P. Corke and B. Armstrong-Helouvy, "A Search for Consensus Among Model Parameters Reported for the Puma 560 Robot," *Proceedings of the 1994 IEEE International Conference on Robotics and Automation*, vol. 2 (1994) pp. 1608–1613.
- [37] A. Lyapunov, "A General Task about the Stability of Motion. (in Russian) *Ph.D. Thesis* (University of Kazan, Tatarstan, Russia, 1892).
- [38] A. Lyapunov, *Stability of Motion* (Academic Press, New-York and London, 1966).
- [39] A. M. Díaz and A. Rodríguez-Angeles, "A pi2d Feedback Control Type for Second Order Systems," **In: Advances in PID Control** (V. D. Yurkevich, ed.) (IntechOpen, Rijeka, 2011), ch. 4
- [40] J. Munkhammar, *Riemann-Liouville Fractional Derivatives and the Taylor-Riemann Series* (Department of Mathematics, Uppsala University, Sweden, 2004).
- [41] H. Redjimi and J. K. Tar, "Extrapolated State Estimation in Fixed Point Transformation-based Adaptive Control Using Fractional Order Feedback," *2018 IEEE 16th World Symposium on Applied Machine Intelligence and Informatics (SAMi)* (2018) pp. 000027–000032.
- [42] J. Tar, J. Bitó, L. Kovács and T. Faitli, "Fractional Order PID-Type Feedback in Fixed Point Transformation-based Adaptive Control of the FitzHugh-Nagumo Neuron Model with Time-Delay," *3rd IFAC Conference on Advances in Proportional-Integral-Derivative Control, May 9-11, 2018, Ghent, Belgium* (IFAC, 2018) pp. 906–911.
- [43] I. Podlubny, *Fractional Differential Equations* (Academic Press, San Diego, 1999).
- [44] A. Attinga, J. F. Bitó and J. K. Tar, "On the Simulation of Lower Order Control Strategies for Higher Order Systems," *2022 IEEE 22nd International Symposium on Computational Intelligence and Informatics and 8th IEEE International Conference on Recent Achievements in Mechatronics, Automation, Computer Science and Robotics (CINTI-MACRo)* (2022) pp. 000119–000124.
- [45] Å. K. Johan and W. Björn, *Adaptive Control* (Addison Wesley, Boston, 1995).
- [46] Z. T. Dydek, A. M. Annaswamy and E. Lavretsky, "Adaptive control and the nasa x-15-3 flight revisited," *IEEE Control Syst. Mag.* **30**(3), 32–48 (2010).
- [47] Å. K. Johan, "Theory and applications of adaptive control—a survey," *Automatica* **19**(5), 471–486 (1983).
- [48] P. A. Ioannou and B. Fidan, *Adaptive Control Tutorial* (SIAM, Philadelphia, PA, 2006).
- [49] P. Parks, "Liapunov redesign of adaptive control systems," *IEEE Trans. Autom. Control* **11**(3), 362–367 (1966).
- [50] C. Nicol, C. Macnab and A. Ramirez-Serrano, "Robust adaptive control of a quadrotor helicopter," *Mechatronics* **21**(6), 927–938 (2011).

- [51] Y. Fang, B. Ma, P. Wang and X. Zhang, “A motion planning-based adaptive control method for an underactuated crane system,” *IEEE Trans. Control Syst. Technol.* **20**(1), 241–248 (2012).
- [52] F. Zouari, K. B. Saad and M. Benrejeb, “Adaptive Backstepping Control for a Single-Link Flexible Robot Manipulator Driven DC Motor,” *2013 International Conference on Control, Decision and Information Technologies (CoDIT)* (2013) pp. 864–871.
- [53] F. Zouari, K. B. Saad and M. Benrejeb, “Robust adaptive control for a class of nonlinear systems using the backstepping method,” *Int. J. Adv. Robot. Syst.* **10**(3), 166 (2013).
- [54] J. Zou and J. K. Schueller, “Adaptive backstepping control for parallel robot with uncertainties in dynamics and kinematics,” *Robotica* **32**(6), E1 (2014).
- [55] M. Mohammad and J. K. Tar, “Adaptive Backstepping Control Design for Nonlinear System,” *2023 IEEE 17th International Symposium on Applied Computational Intelligence and Informatics (SACI)* (2023) pp. 000147–000152.
- [56] J. Na, S. Wang, Y.-J. Liu, Y. Huang and X. Ren, “Finite-time convergence adaptive neural network control for nonlinear servo systems,” *IEEE Trans. Cybern.* **50**(6), 2568–2579 (2020).
- [57] J. Na, Y. Huang, X. Wu, S.-F. Su and G. Li, “Adaptive finite-time fuzzy control of nonlinear active suspension systems with input delay,” *IEEE Trans. Cybern.* **50**(6), 2639–2650 (2020).
- [58] S. Wang, J. Na and X. Ren, “Rise-based asymptotic prescribed performance tracking control of nonlinear servo mechanisms,” *IEEE Trans. Syst. Man Cybern. Syst.* **48**(12), 2359–2370 (2018).
- [59] J. Na, Q. Chen, X. Ren and Y. Guo, “Adaptive prescribed performance motion control of servo mechanisms with friction compensation,” *IEEE Trans. Ind. Electron.* **61**(1), 486–494 (2014).
- [60] S. Wang, X. Ren, J. Na and T. Zeng, “Extended-state-observer-based funnel control for nonlinear servomechanisms with prescribed tracking performance,” *IEEE Trans. Autom. Sci. Eng.* **14**(1), 98–108 (2017).
- [61] S. Wang, H. Yu, J. Yu, J. Na and X. Ren, “Neural-network-based adaptive funnel control for servo mechanisms with unknown dead-zone,” *IEEE Trans. Cybern.* **50**(4), 1383–1394 (2020).
- [62] A. Dineva, J. Tar and A. Várkonyi-Kóczy, “Novel Generation of Fixed Point Transformation for the Adaptive Control of a Nonlinear Neuron Model,” *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics, October 10-13, 2015, Hong Kong (SMC, 2015)* pp. 987–992.
- [63] A. Dineva, J. Tar, A. Várkonyi-Kóczy and V. Piuri, “Generalization of a Sigmoid Generated Fixed Point Transformation from SISO to MIMO Systems,” *Proceedings of the IEEE 19th International Conference on Intelligent Engineering Systems, September 3-5, 2015, Bratislava, Slovakia* (2015) pp. 135–140.
- [64] H. Issa and J. K. Tar, “Noise Sensitivity Reduction of the Fixed Point Iteration-based Adaptive Control,” *2021 IEEE 19th International Symposium on Intelligent Systems and Informatics (SISY)* (2021) pp. 171–176.
- [65] J. K. Tar, J. F. Bitó and I. J. Rudas, “Replacement of Lyapunov’s Direct Method in Model Reference Adaptive Control with Robust Fixed Point Transformations,” *2010 IEEE 14th International Conference on Intelligent Engineering Systems* (2010) pp. 231–235.
- [66] B. Csanádi, P. Galambos, J. Tar, G. Györök and A. Serester, “A Novel, Abstract Rotation-based Fixed Point Transformation in Adaptive Control,” *Proceedings of the 2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC2018), October 7-10, 2018, Miyazaki, Japan* (2018) pp. 2577–2582.
- [67] H. Issa and J. K. Tar, “Improvement of an adaptive robot control by particle swarm optimization-based model identification,” *Mathematics* **10**(19), 3609 (2022).
- [68] R.-C. Roman, R.-E. Precup, E. M. Petriu and F. Dragan, “Combination of data-driven active disturbance rejection and Takagi-Sugeno fuzzy control with experimental validation on tower crane systems,” *Energies* **12**(8), 1548 (2019).
- [69] D. Fényes, B. Németh and P. Gáspár, “Design of LPV control for autonomous vehicles using the contributions of big data analysis,” *Int. J. Control* **95**(7), 1–12 (2021).
- [70] F. Ding, X. Luo, Y. Cai and W. Chang, “Acceleration feedback control for enhancing dynamic stiffness of fast tool servo system considering the sensor imperfections,” *Mech. Syst. Signal Process.* **141**, 106429 (2020).
- [71] J. Tašić, M. Takács and L. Kovács, “Control engineering methods for blood glucose levels regulation,” *Acta Polytech. Hung.* **19**(7), 127–152 (2022).
- [72] A. Dineva, J. Tar, A. Várkonyi-Kóczy and V. Piuri, “Adaptive Controller Using Fixed Point Transformation for Regulating Propofol Administration Through Wavelet-based Anesthetic Value,” *Proceedings of the IEEE International Symposium on Medical Measurements and Applications (MeMeA 2016), 05.15-05.18, Benevento, Italy* (2016) pp. 650–655.
- [73] H. Khan, H. Issa and J. K. Tar, “Application of the robust fixed point iteration method in control of the level of twin tanks liquid,” *Computation* **8**(4), 96 (2020).
- [74] DFRobot, “Metal dc geared motor w/encoder - 12v 83rpm 45kg.cm (2022). Accessed: 2022-11-10. Available at: <https://www.dfrobot.com/product-633.html>.
- [75] HESore, “Bts7960-m - motordrive (2022). Accessed: 2022-11-10. Available at: https://www.hestore.hu/prod_10038105.html.
- [76] E. C. Ifeachor and B. W. Jervis, *Digital Signal Processing: A Practical Approach* (Pearson Education, Edinburgh, UK, 2002)
- [77] B. Varga, R. Horváth and J. K. Tar, “Adaptive Control of a Nonlinear System Avoiding State Estimation,” *2022 IEEE 20th Jubilee World Symposium on Applied Machine Intelligence and Informatics (SAMI)* (2022) pp. 000323–000328.