2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

Using the Chebyshev Basis for Energy Optimal Motion Profiles in Path-Constrained Applications

Nick Van Oosterwyck, Robbe De Laet, Lorenzo Scalera, Annie Cuyt, Alessandro Gasparetto, and Stijn Derammelaere, *Member, IEEE*

***Abstract*— Motion profile optimization is a powerful optimization technique that allows to reduce the energy consumption of robotic systems by changing the temporal profile of the joint position setpoints. However, despite the extensive exploration of these techniques for robotic systems following constrained paths, many existing methodologies rely on complex optimization processes or a larger number of design parameters. This paper introduces a novel approach that leverages the Chebyshev basis to optimize the motion along a fixed geometric path, thereby achieving a measured torque difference of -15% while requiring only a limited number of design parameters. By employing the Chebyshev basis, the formulation leads to a smooth objective function and enables the definition of linear inequality constraints that accurately enclose the feasible design space. This unique combination of features not only simplifies the optimization problem but also enhances the probability of locating the global optimum, particularly illustrated in the two-dimensional case. The methodology is established in a generic and model-independent manner, setting a promising direction for future research in motion profile optimization for constrained-path robotic systems.**

I. Introduction

As the world is facing significant energy challenges and clean energy sources remain scarce, every sector, including robotics, must contribute to enhancing the energy efficiency of its operations. This initiative not only fosters environmental sustainability but also reduces operational costs and is often necessary to comply with increasingly stringent legal requirements, as indicated in the recently updated ecodesign directive [1]. For instance, electric motors such as the ones deployed in robotic systems are estimated to comprise 50% of the total electricity consumption [1], underscoring the potential for major energy savings in this field.

One prevalent method for reducing the energy consumption of robotic systems is the utilization of motion profile optimization, which aims at finding an energy-optimal trajectory for moving a robot from a specified start to end position [2]. In the literature, robotic motion profile optimizations can be categorized into two types. On the one hand, *free-path* optimizations consider only the start and end pose, permitting the robot to move unconstrained between these positions, albeit with considerations for physical robot limits and potential collisions [3]. On the other hand, *fixed-path* optimizations are focused on enhancing the motion profile along a predetermined and constrained end-effector trajectory. This geometric path may originate from various requirements. For instance, a robot may need to adhere rigorously to a defined path to execute a path-dependent task or circumvent obstacles, as is often the case in manufacturing processes such welding or spray-painting [4]. Furthermore, a higher-level obstacle avoidance algorithm may have generated a series of collision-free paths that fulfil a task and now require evaluation [5].

In this paper, we specifically focus on the optimization of fixed-path or path-constrained motion profiles. The traditional approach to this type of problem involves introducing a state variable s, symbolizing the robot's distance along the designated geometric path, followed by the optimization of the temporal profile of this state variable $s(t)$ [6]. This process effectively simplifies the multi-axis robotic optimization into a one degree-of-freedom (DOF) motion profile optimization [7]. Within the academic literature, a variety of methods have been proposed for defining and optimizing the path variable s. These methodologies range from employing B-splines to achieve time-optimal execution, as introduced by [8], [9], to utilizing B-splines in converting the problem into a convex optimization challenge, as explored by [10]. Furthermore, the combination of cubic splines with genetic algorithms to minimize motion time is investigated in [11]. Other strategies encompass the usage of quintic polynomials for real-time optimization by [12], or the application of asymmetric s-curves to attain time-optimality, as evidenced in [13]. However, it's noteworthy that many of these strategies primarily concentrate on time optimality, which neglects the fact that in many practical applications, the motion time is often fixed by the process requirements.

Therefore, other approaches in the literature extend beyond time optimality and engage with other assessments, such as the trade-off between time and energy optimality. Specifically, Pontryagin's maximum principle has been employed to investigate this compromise in [14]. Along similar lines, [15] has employed dynamic programming to analyze this trade-off, while [16] has adapted the Pontryagin principle for the time optimality of redundant robots. However, it must be noted that these algorithms present certain challenges, as highlighted by [6], [17]; they can be complex to implement,

N. Van Oosterwyck, R De Laet and S. Derammelaere are with the Department of Electromechanics, Cosys-Lab, University of Antwerp, 2020 Antwerp, Belgium, and also with AnSyMo/CoSys, Flanders Make, the strategic research center for the manufacturing industry (e-mail: nick.vanoosterwyck@uantwerpen.be; robbe.delaet@uantwerpen.be; stijn.derammelaere@uantwerpen.be)

A. Cuyt is with the Department of Computer Science, Computational Mathematics, University of Antwerp, CMI, 2020 Antwerp, Belgium, and also with the Division of Computing Science and Mathematics, University of Stirling, FK9 4LA Stirling, Scotland (e-mail: annie.cuyt@uantwerpen.be)

L. Scalera and A. Gasparetto are with the Polytechnic Department of Engineering and Architecture (DPIA), University of Udine, 33100 Udine, Italy (e-mail: lorenzo.scalera@uniud.it; alessandro.gasparetto@uniud.it)

may encounter issues relating to numerical stability, and might lead to suboptimal solutions. Furthermore, the absence of guarantees for global optimality, as pointed out in [18], further complicates the applicability and robustness of these methodologies.

Another prevalent research direction involves discretizing the control state s into a series of piecewise constant parts. For instance, [18] employed a piecewise constant acceleration motion profile, enabling conversion into a convex optimization problem with linear constraints. This approach was further generalized to second-order systems by [5], who also proposed an algorithm to reduce the computation time. However, the convexity requirement restricts the applicability to a limited set of objectives. Other researchers, such as [19], have also employed discretization, specifically to find time-optimal values based on torque limits. Further variations include the use of piecewise rational functions by [7], and the employment of reinforcement learning by [20] to minimize tracking errors. Finally, [21] focused on optimizing the discrete timings of a predefined end-effector path to reduce the total motion time and energy, but only considered a pseudo-power metric.

In general, prevailing methods tend to overlook energy consumption, employ complex algorithms, or risk entrapment in sub-optimal solutions. Moreover, they often resort to discretized motion profiles that require a large amount of design parameters. However, [22] delineated a novel motion profile approach employing the Chebyshev basis, enabling the straightforward optimization of these motion profiles with a minimal set of design parameters, albeit only for single-axis systems. This methodology was extended to multi-axis robotic systems in [3], yet it did not address path-constrained scenarios, which are particularly prevalent in industrial applications. By integrating the Chebyshev basis into path-constrained optimization of robotic systems, this paper seeks to contribute the following critical advancements:

- Although considerable research has been devoted to time-optimality, the specific focus on energy minimization for tasks with a predetermined motion time is frequently neglected. Therefore, this paper introduces the use of the Chebyshev basis for the energy optimal fixed-path motion profile optimization.
- While numerous existing methods typically rely on discretized states, demanding a substantial number of design parameters, the employment of the Chebyshev basis enables energy savings with a limited number of design parameters.
- Using the Chebyshev basis in combination with a linearized constraint formulation reduces the risk of convergence to local minima compared to a classical polynomial basis. Moreover, for certain tasks, this approach can even reveal the global optimum, as evidenced in the optimization involving two design parameters.
- To thoroughly evaluate the method's flexibility, stability, and feasibility, a series of optimizations and experiments were conducted on a 7-DOF serial robot using four distinct end-effector paths. These measurements were designed to represent a variety of real-world scenarios, thereby demonstrating the method's performance under diverse operating conditions.

The remainder of the article is structured as follows: Section II defines the optimization problem and introduces the rescaled path profile $s(t)$. In Section III, the kinematic and dynamic models of the robot are outlined. Finally, Section IV validates the feasibility of the method by presenting the optimization results and measurements, followed by a conclusion in Section V that summarizes the work.

II. Optimization

A. Fixed-Path Definition

The initial step in a fixed-path optimization involves outlining the necessary end-effector path. In this paper, an approach similar to [21] is followed, where the end-effector path is captured or defined as a set of N_{path} discrete joint position vectors $\boldsymbol{\theta}_{\text{path}} = \{\boldsymbol{\theta}_{\text{path},1}, \ldots, \boldsymbol{\theta}_{\text{path},N_{path}}\}$. Using the forward kinematics and Denavit-Hartenberg (DH)-matrices, the N_{path} corresponding end-effector positions $\mathbf{p}_{\text{path}} = \{\mathbf{p}_{\text{path},1}, \ldots, \mathbf{p}_{\text{path},N_{path}}\}$ are then determined. Conversely, if the end-effector positions $\mathbf{p}_{\text{path}}$ and rotations are specified, the corresponding position vectors $\boldsymbol{\theta}_{\text{path}}$ of a 6-DOF robot can be identified using the inverse kinematics, as outlined in Sec. III. This approach eliminates the need for an analytic description of the path, allowing data collection from the real robot executing the prescribed path, thus allowing to improve the energy efficiency of existing systems.

Next, the path length L can be calculated by summing the Euclidean distance between each pair of consecutive points:

$$L = \sum_{i=1}^{N_{path}-1} \|\mathbf{p}_{\text{path},i+1} - \mathbf{p}_{\text{path},i}\| \tag{1}$$

For the fixed-path optimization, a path variable $s \in [0, 1]$ is introduced, signifying the relative position of the robot along the path (Fig. 1). By utilizing the path length L from (1), the path data points $\mathbf{p}_{\text{path}}$ can be mapped to the corresponding path distance $\mathbf{s}_{\text{path}} = [\mathbf{s}_{\text{path},1}, \ldots, \mathbf{s}_{\text{path},N_{path}}]$, where each l-th element is defined by its relative distance along the path:

$$\mathbf{s}_{\text{path},l} = \frac{\sum_{i=1}^{l-1} \|\mathbf{p}_{\text{path},i+1} - \mathbf{p}_{\text{path},i}\|}{L}, \quad l \in 1, 2, \ldots, N_{path}. \tag{2}$$

Note that since each point in $\mathbf{p}_{\text{path}}$ corresponds to a point in $\boldsymbol{\theta}_{\text{path}}$. This mapping creates a linkage between the path state $\mathbf{s}_{\text{path}}$ and the path joint vectors $\boldsymbol{\theta}_{\text{path}}$, a connection that is subsequently utilized in (3) for interpolation.

To optimize how the robot moves along the path, a time-dependent function for the path variable $s(t), t \in [t_A, t_B]$ can be defined, with t_A and t_B representing the start and end times of the motion, respectively. By sampling the function $s(t)$ with a specific sample time t_s, the discrete s-values can then be used to linearly interpolate path data $\{\mathbf{s}_{\text{path}}, \boldsymbol{\theta}_{\text{path}}\}$ and deduce the corresponding discrete joint angles $\boldsymbol{\theta}$. In summary, for the l-th time sample t_l, the path distance $s_l = s(l \cdot t_s + t_A)$ is determined, which is then used in the

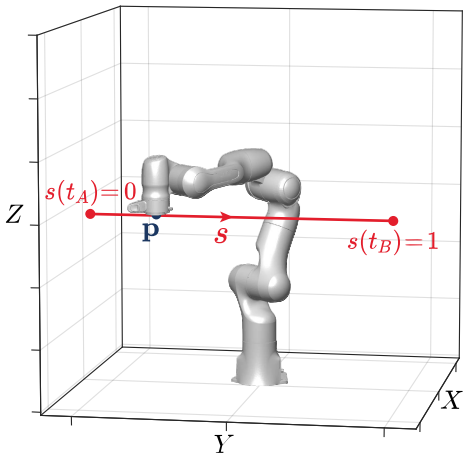

Fig. 1. The end-effector path variable $s(t), t \in [t_A, t_B]$ for a given end-effector path.

subsequent linear interpolation to determine the new joint position θ_i:

$$\theta_i = \theta_{\text{path},k} + \frac{\theta_{\text{path},k+1} - \theta_{\text{path},k}}{s_{\text{path},k+1} - s_{\text{path},k}} \cdot (s_l - s_{\text{path},k}), \tag{3}$$

where k is the smallest index such that $s_{\text{path},k} \leq s_l \leq s_{\text{path},k+1}$. By repeating this procedure for each time sample t_l, the new joint profiles $\theta(t)$ are obtained, which can be utilized in the dynamic model (12) or sent to the robot to assess its performance. It should be noted that reducing the sample time t_s leads to a finer resolution of the joint profiles, but also necessitates more frequent evaluation of the dynamic equation. Consequently, we proceed with a sample time of $t_s = 0.01\,\text{s}$, which emerges as the optimal balance between accuracy and computational effort [3].

Furthermore, although interpolating the path in Cartesian space would be more accurate, it would also necessitate repeated inverse kinematics computations during the optimization, substantially increasing the computational load. For this reason, the end-effector path is interpolated in joint space. Since the path is represented using a sufficiently dense discretization, in this case $N_{\text{path}} = 401$ points, the resulting approximation error is negligible.

B. Motion Profile

In the preceding section, a method was detailed for converting a specific path profile $s(t)$ into a set of joint angles θ, ensuring the robot's end-effector stayed on the predefined path. The remaining task is to determine the optimal temporal profile for the function $s(t)$. To achieve this, the motion profile is expressed as a linear combination of Chebyshev polynomials of degree d, denoted by $\phi(x)$:

$$\phi(x) = \sum_{k=0}^{d} p_k T_k(x), \quad x \in [-1, 1], \tag{4}$$

where each Chebyshev polynomial $T_k(x)$ is obtained using the recurrence relation:

$$\begin{aligned} T_0(x) &= 1, \quad T_1(x) = x, \\ T_{k+1}(x) &= 2x\,T_k(x) - T_{k-1}(x). \end{aligned} \tag{5}$$

However, since $\phi(x)$ should be defined on the interval $x \in [-1, 1] \to [-1, 1]$, it cannot be directly applied to the path function $s(t)$. Following the approach in [3], we therefore introduce a rescaled path profile:

$$s(t) = \frac{1}{2}\phi\left(\frac{2\,(t - t_A)}{t_B - t_A} - 1\right) + \frac{1}{2}, \tag{6}$$

which sets up the mapping $\phi(x) : [-1, 1] \to [-1, 1]$ to the rescaled motion profile $s(t) : [t_A, t_B] \to [0, 1]$.

C. Constraints

The next phase in defining the optimization problem involves defining the constraints. As the scenario under consideration is a rest-to-rest motion, the subsequent position θ, speed $\dot{\theta}$, and acceleration $\ddot{\theta}$ requirements must be met [23]:

$$\begin{aligned} \theta(t_A) = \theta_A \quad , \quad \dot{\theta}(t_A) = 0 \quad , \quad \ddot{\theta}(t_A) = 0, \\ \theta(t_B) = \theta_B \quad , \quad \dot{\theta}(t_B) = 0 \quad , \quad \ddot{\theta}(t_B) = 0, \end{aligned} \tag{7}$$

where $\theta(t_A) = \theta_A, \theta(t_B) = \theta_B$ represent the joint positions of the robot's start and end locations on the path. In the optimization, however, it is more efficient to have the constraints defined directly to this path profile $s(t)$ to avoid any added inaccuracy by interpolating according to (3). The position requirements in (7) are automatically met by letting s vary between $[0, 1]$. However, for the time derivatives in (7), they are initially reformulated using the chain rule, as described in [18]:

$$\dot{\theta} = \frac{\partial \theta(s)}{\partial s}\dot{s}, \quad \ddot{\theta} = \frac{\partial \theta(s)}{\partial s}\ddot{s} + \frac{\partial^2 \theta(s)}{\partial s^2}\dot{s}^2, \tag{8}$$

from which it is evident that, to satisfy (7), the time derivatives of $s(t)$ must also be zero. This results in the updated path motion requirements:

$$\begin{aligned} s(t_A) = 0 \quad , \quad \dot{s}(t_A) = 0 \quad , \quad \ddot{s}(t_A) = 0, \\ s(t_B) = 1 \quad , \quad \dot{s}(t_B) = 0 \quad , \quad \ddot{s}(t_B) = 0. \end{aligned} \tag{9}$$

As pointed out in [24], these six requirements can be incorporated into the motion profile definition (6) by symbolically writing the six lower degree coefficients $[p_0, \ldots, p_5]^T$ as functions of the remaining higher degree coefficients $[p_6, \ldots, p_d]^T$. The latter forms the design parameter vector $\mathbf{o}$, which consists of $n_{\text{dp}} = d - 5$ design parameters.

The second type of constraint relates to the monotonicity of $s(t)$. To prevent back-and-forth movement along the path, the constraint $\dot{s} \geq 0$ is imposed. However, as noted in [3], by symbolically constructing (6) and discretizing it for the predefined set of time instances t_l, this single nonlinear constraint can be transformed into a series of linear inequality constraints $\dot{s}|_{t=t_l} \geq 0, t_l \in [t_A, \ldots, t_B]$, where $\dot{s}$ can be written as a linear combination of the design parameters in $\mathbf{o}$ [3]. This conversion accelerates convergence, as the resulting

linear constraints are easier for the optimization algorithm to handle than a single nonlinear inequality constraint [25].

In addition, the manufacturer-specified limits on joint velocity $\dot{\theta}_{\max}$, acceleration $\ddot{\theta}_{\max}$, jerk $\dddot{\theta}_{\max}$, and torque $\tau_{\max}$ are imposed as constraints.

D. Objective value

This paper aims to minimize the energy demand while keeping the total motion time Δt fixed. Consequently, an accurate quantification of the total energy consumption E_{tot} requires an appropriate energy model. However, precise predictions require knowledge of the joint motor parameters, such as armature resistance R_j and torque constant $k_{t,j}$, which are often unknown. Nevertheless, for high dynamical systems where friction is negligible, such as robotic manipulators, the joint j root-mean-square (rms) torque $\tau_{rms,j}$ can still provide an accurate estimate of the joint energy E_j [26]. Therefore, consistent with [27], the total rms torque $\tau_{rms,tot}$ is employed as the optimization metric:

$$\tau_{\text{rms,tot}} = \sum_{j=1}^{N} \tau_{\text{rms},j} = \sum_{j=1}^{N} \sqrt{\frac{1}{m}\left({\tau_{1,j}}^2 + \cdots + {\tau_{m,j}}^2\right)}, \quad (10)$$

with m the number of torque samples. This leads to the formulation of the final optimization problem:

$$\begin{aligned} \min_{\mathbf{o} \in [\frac{-4}{\pi}, \frac{4}{\pi}]^{n_{\text{dp}}}} \quad & \tau_{\text{rms,tot}} = \sum_{j=1}^{N} \tau_{\text{rms},j}(s(\mathbf{o})) \\ \text{s.t.} \quad & \dot{s}|_{t=t_l} \geq 0, \\ & -\dot{\theta}_{j,\max} < \dot{\theta}_j(t) < \dot{\theta}_{j,\max}, \\ & -\ddot{\theta}_{j,\max} < \ddot{\theta}_j(t) < \ddot{\theta}_{j,\max}, \\ & -\dddot{\theta}_{j,\max} < \dddot{\theta}_j(t) < \dddot{\theta}_{j,\max}, \\ & -\tau_{j,\max} < \tau_j(t) < \tau_{j,\max}, \\ & -\dot{\tau}_{j,\max} < \dot{\tau}_j(t) < \dot{\tau}_{j,\max}, \\ \text{with} \quad & j \in [1, \ldots, N], \\ & t_l \in [t_A, \ldots, t_B]. \end{aligned} \quad (11)$$

Similar to [3], a zero initial design parameter vector $\mathbf{o} = \mathbf{0}$ is selected for the optimization, causing the path function $s(t)$ to resolve to a classic 5th-degree polynomial, which is also used as the reference point.

It should be emphasized that although the minimization of the total rms torque $\tau_{\text{rms,tot}}$ is presented here, the optimization of another metric could easily be integrated by merely altering the objective function without necessitating any changes to the previously described method. Additionally, this approach does not place any requirements on the dynamic model itself, allowing for easy substitution in the future with a more sophisticated energy model or simulation.

III. System Modeling

Although the methodology put forth is applicable across a broad spectrum of robotic arms, its validation is performed with the 7-DOF Panda robot from Franka Emika GmbH (Fig. 1.) This serves to highlight both the universality of the approach and its concrete applicability in a specific real-world context. Here, we briefly recall the kinematic and dynamic models of the robotic system necessary to compute all components of (11).

The first step in setting up the kinematic model consists of defining the Denavit-Hartenberg (DH) reference frames and parameters a_j, α_j and d_j for each joint j. Here, we follow the classic (distal) convention [28] as reported in [3] and make sure the $N = 7$ joint angles $\boldsymbol{\theta} = [\theta_1 \ldots \theta_N]^T$ are consistent with the Franka Control Interface (FCI).

However, considering the Panda robot's redundancy with its 7 DOF, the inverse kinematics that determines the joint angles $\boldsymbol{\theta}$ from the end-effector position vector $\mathbf{p} = [p_x, p_y, p_z]^T$ lacks unique definition. Nonetheless, to maintain the paper's focus, the redundancy exploitation will not be our prime concern. Therefore, similar to [20] we assume that this redundancy is handled by a higher level planning algorithm and we will proceed under the assumption that for every imposed end-effector path, the joint angle $\theta_3 = 0$. This allows the problem to be treated as a conventional 6-DOF robot and enables a straightforward computation of the inverse kinematics.

To compute the total rms torque $\tau_{\text{rms,tot}}$, a dynamic model is required that can calculate the joint torque $\boldsymbol{\tau} = [\tau_1 \ldots \tau_N]^T$ as a function of joint positions $\boldsymbol{\theta}$, velocities $\dot{\boldsymbol{\theta}}$, and accelerations $\ddot{\boldsymbol{\theta}}$. Here, the Lagrangian formalism, is utilized:

$$\boldsymbol{\tau}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}, \ddot{\boldsymbol{\theta}}) = \mathbf{M}(\boldsymbol{\theta})\ddot{\boldsymbol{\theta}} + \mathbf{c}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}})\dot{\boldsymbol{\theta}} + \mathbf{g}(\boldsymbol{\theta}) + \boldsymbol{\tau}_f(\dot{\boldsymbol{\theta}}); \quad (12)$$

with the robot's mass matrix $\mathbf{M}(\boldsymbol{\theta}) \in \mathbb{R}^{N \times N}$, the Coriolis and centrifugal terms $\mathbf{c}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) \in \mathbb{R}^{N \times 1}$, the gravity vector $\mathbf{g}(\boldsymbol{\theta}) \in \mathbb{R}^{N \times 1}$, and the joint friction vector $\boldsymbol{\tau}_f(\dot{\boldsymbol{\theta}}) \in \mathbb{R}^{N \times 1}$.

The inertial parameters for these matrices are obtained from [29], along with the friction model for $\boldsymbol{\tau}_f$, for which a sigmoidal function was identified.

IV. Results

This section details the experimental validation of the fixed-path motion profile optimization on the Franka robot. Following the approach outlined in [12], Task 1 is defined as a straight-line horizontal point-to-point motion in Cartesian space. To assess the algorithm's adaptability across varying motion types, Task 2 introduces a 1-meter vertical linear motion, with the same start and end points taken from [3]. Furthermore, to account for curved end-effector trajectories, a circular path with a radius of 0.5 meters is defined in Task 3 between the same start and end points as in Task 2. Task 4 is adopted from [30]. The corresponding initial and final joint angles $\boldsymbol{\theta}_A$, $\boldsymbol{\theta}_B$, and the motion duration $\Delta t = t_B - t_A$ are provided in Tab. I. Fig. 2 depicts the robot configurations along these paths, while Fig. 3 illustrates the resulting three end-effector trajectories.

In Tab. II, both the virtual and measured optimization results are displayed. The simulation data is generated using Matlab R2024a software and the gradient-based SQP optimization algorithm from the `fmincon` function with

TABLE I

START AND END JOINT POSITIONS θ, [rad], AND THE MOTION TIME DURATION Δt, [s] OF THE CONSIDERED TASKS

	joint j							
	1	2	3	4	5	6	7	Δt
[rad]								[s]
Task 1								
$\theta_{A,j}$	-1.01	0.10	0	-1.51	0	1.610	-0.23	4
$\theta_{B,j}$	1.01	0.10	0	-1.51	0	1.610	1.80	
Task 2,3								
$\theta_{A,j}$	0	0.14	0	-1.11	0	1.26	0	4
$\theta_{B,j}$	0	0.34	0	-2.21	0	2.55	0	
Task 4								
$\theta_{A,j}$	-0.79	0.66	0	-2.10	0	2.76	0	5
$\theta_{B,j}$	0.46	0.48	0	-2.51	0	2.99	1.25	

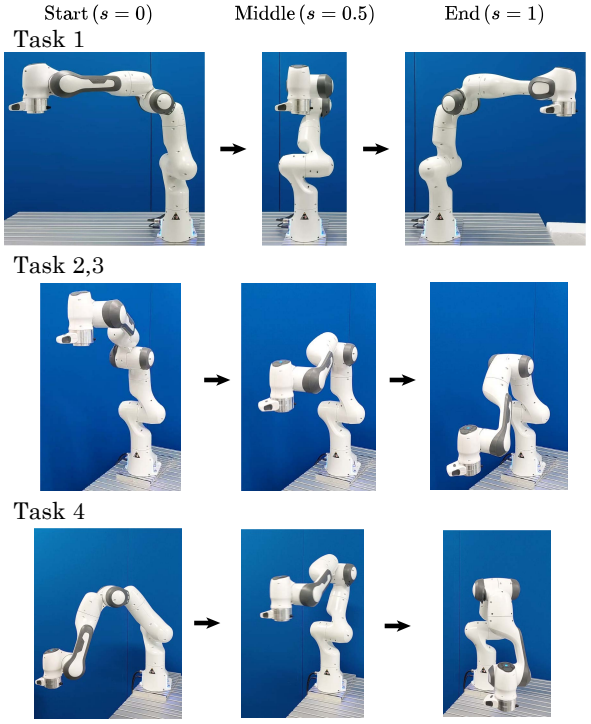

Fig. 2. The start and end configurations of the robot, along with its position halfway along the path.

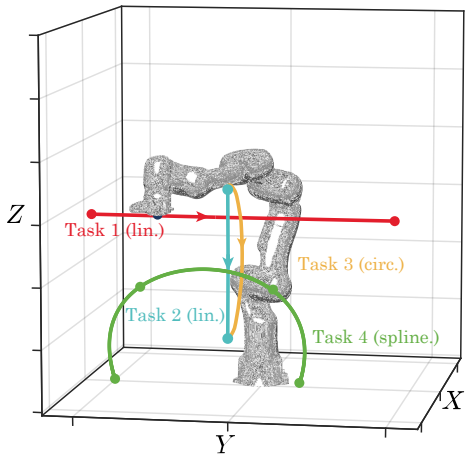

Fig. 3. The various end-effector trajectories examined in this study.

default convergence criteria. The measurements are obtained by transmitting the optimized joint setpoints $\theta(t)$ via Python and ROS Melodic to the Franka robot, which operates in position control mode and publishes data at a rate of 1 kHz.

The results presented in Tab. II validate the dynamic model by demonstrating a strong correlation between the predicted and measured root-mean-square (rms) torques $\tau_{\text{rms,tot}}$. For Task 1, Task 3 and Task 4, despite the constraint that the robot must follow the same path within the same time duration, the measured $\tau_{\text{rms,tot}}$ shows a notable difference of up to -15%. This clearly highlights the effectiveness and potential of the proposed optimization procedure. Furthermore, even with only two design parameters $n_{\text{dp}} = 2$, a substantial energy reduction is already achieved, requiring only a limited number of function evaluations and therefore enabling a very fast optimization process. These findings emphasize the benefit of using the Chebyshev basis, in contrast to classical polynomial bases such as those used in [24], where two design parameters did not lead to significant energy savings.

However, the results also indicate that the potential for energy savings is highly dependent on the imposed end-effector path. For instance, in the case of the linear path of Task 2, both the theoretical and measured torque savings are limited.

In Fig. 4, Fig. 5, Fig. 6, and Fig. 7; the measured profiles of position $\theta(t)$, velocity $\dot{\theta}(t)$, and torque $\tau(t)$ are presented for the three respective tasks, along with the corresponding path profiles $s(t)$ for both the reference and the best-obtained solution ($n_{\text{dp}} = 8$) as reported in Tab. II. For Task 1, Task 3 and Task 4, the comparison shows that to reduce the total torque $\tau_{\text{rms,tot}}$, the optimized profile accelerates more rapidly and then remains longer near the midpoint. This behavior is expected since a less extended configuration requires less torque to maintain position, as illustrated in Fig. 2, thereby demonstrating the effectiveness of the method.

In contrast, for Task 2, the torque savings were significantly lower. This is reflected here by the relatively stable torque components throughout the motion, indicating that no particular segment of the path exhibits a substantially reduced torque demand that could be exploited by the algorithm. This explains the limited potential for optimization in this case. Readers are strongly encouraged to view the supplementary research video, which includes recordings of the robot performing this movement.

In addition, to demonstrate the effectiveness of the proposed approach, the same optimizations were performed using a conventional polynomial method [2]. In this formulation, the motion profile, here represented by the path variable, is defined using classical polynomials, $s(t) = \sum_{k=0}^{d} p_k\, t^k$, as summarized in Table III. From these results, it becomes evident that the polynomial representation in the Chebyshev basis clearly outperforms the classical formulation. While

TABLE II

SIMULATED AND MEASURED RESULTS OF THE MOTION PROFILE OPTIMIZATION AS DEFINED IN (11) FOR ALL CONSIDERED TASKS AND DIFFERENT NUMBER OF DESIGN PARAMETERS n_{dp}, WHICH DEMONSTRATES A CONVERGING SAVINGS POTENTIAL AND STRONG CORRELATION BETWEEN THE SIMULATED AND MEASURED TORQUES $\tau_{\mathrm{rms,tot}}$.

	simulation			measurement	
	$\tau_{\mathrm{rms,tot}}$ [Nm]	change [%]	N_f []	$\tau_{\mathrm{rms,tot}}$ [Nm]	change [%]
Task 1					
$n_{\mathrm{dp}}=0$	42.74	ref.	-	43.06	ref.
$n_{\mathrm{dp}}=2$	39.20	-8.3	9	39.20	-9.0
$n_{\mathrm{dp}}=4$	37.88	-11.4	163	37.85	-12.1
$n_{\mathrm{dp}}=6$	37.35	-12.6	432	37.31	-13.4
$n_{\mathrm{dp}}=8$	37.17	-13.0	298	37.07	-13.9
Task 2					
$n_{\mathrm{dp}}=0$	47.37	ref.	-	48.35	ref.
$n_{\mathrm{dp}}=2$	46.75	-1.3	19	48.09	-0.5
$n_{\mathrm{dp}}=4$	46.50	-1.8	21	47.00	-0.7
$n_{\mathrm{dp}}=6$	46.44	-2.0	89	47.93	-0.9
$n_{\mathrm{dp}}=8$	46.42	-2.0	207	47.97	-0.8
Task 3					
$n_{\mathrm{dp}}=0$	41.11	ref.	-	41.99	ref.
$n_{\mathrm{dp}}=2$	37.70	-8.3	26	38.52	-8.3
$n_{\mathrm{dp}}=4$	35.90	-12.7	70	36.27	-13.6
$n_{\mathrm{dp}}=6$	35.23	-14.3	277	35.71	-15.0
$n_{\mathrm{dp}}=8$	35.08	-14.7	595	35.68	-15.0
Task 4					
$n_{\mathrm{dp}}=0$	45.49	ref.	-	45.89	ref.
$n_{\mathrm{dp}}=2$	43.00	-5.5	127	43.08	-6.1
$n_{\mathrm{dp}}=4$	41.94	-7.8	280	42.02	-8.4
$n_{\mathrm{dp}}=6$	41.75	-8.2	565	41.80	-8.9
$n_{\mathrm{dp}}=8$	41.62	-8.5	375	41.64	-9.3

for $n_{\mathrm{dp}} = 2$ the classical polynomials still manage to reach comparable solutions, the optimization problem is significantly more ill-conditioned and far more susceptible to convergence to local minima, which is reflected in the inconsistent and noticeably worse optimization outcomes for higher number of design parameters n_{dp}.

Finally, to demonstrate the influence of using the Chebyshev basis on the objective function $\tau_{\mathrm{rms,tot}}$, we consider the case where $n_{\mathrm{dp}} = 2$ of Task 1, leading to the design parameter vector $\mathbf{o} = [p_6, p_7]^T \in \left[\frac{-4}{\pi}, \frac{4}{\pi}\right]$. However, while this interval is guaranteed to include the global minimum [25], by considering the linear inequality constraints $\dot{s}|_{t=t_l} \geq 0, t_l \in [t_A, \ldots, t_B]$, the design space can be substantially reduced, as depicted in Fig. 8, where the boundaries of these linear equalities are plotted for 20 t_l steps.

Furthermore, in Fig. 8, the objective function $\tau_{\mathrm{rms,tot}}$ is depicted within this feasible domain, underscoring the absence of local minima in the 2D case and affirming that the

TABLE III

COMPARISON OF CLASSICAL POLYNOMIAL OPTIMIZATION [2] AND THE PROPOSED CHEBYSHEV-BASIS METHOD, SHOWING SUPERIOR OPTIMIZATION PERFORMANCE OF THE LATTER.

	Chebyshev basis		classic polynomials [2]	
	$\tau_{rms,tot}$ [Nm]	change [%]	$\tau_{rms,tot}$ [Nm]	change [%]
Task 1				
$n_{\mathrm{dp}}=0$	42.74	ref.	42.74	ref.
$n_{\mathrm{dp}}=2$	39.20	-8.3	39.20	-8.3
$n_{\mathrm{dp}}=4$	37.88	-11.4	39.19	-8.3
$n_{\mathrm{dp}}=6$	37.35	-12.6	40.58	-5.05
$n_{\mathrm{dp}}=8$	37.17	-13.0	41.86	-4.04
Task 2				
$n_{\mathrm{dp}}=0$	47.37	ref.	47.37	ref.
$n_{\mathrm{dp}}=2$	46.75	-1.3	46.75	-1.3
$n_{\mathrm{dp}}=4$	46.50	-1.8	46.60	-1.6
$n_{\mathrm{dp}}=6$	46.44	-2.0	46.53	-1.8
$n_{\mathrm{dp}}=8$	46.42	-2.0	46.82	-1.2
Task 3				
$n_{\mathrm{dp}}=0$	41.11	ref.	41.11	ref.
$n_{\mathrm{dp}}=2$	37.70	-8.3	37.70	-8.3
$n_{\mathrm{dp}}=4$	35.90	-12.7	37.67	-8.4
$n_{\mathrm{dp}}=6$	35.23	-14.3	37.82	-8.0
$n_{\mathrm{dp}}=8$	35.08	-14.7	36.05	-12.3
Task 4				
$n_{\mathrm{dp}}=0$	45.49	ref.	45.49	ref.
$n_{\mathrm{dp}}=2$	43.00	-5.5	43.00	-5.5
$n_{\mathrm{dp}}=4$	41.94	-7.8	43.06	-5.3
$n_{\mathrm{dp}}=6$	41.75	-8.2	45.29	-0.4
$n_{\mathrm{dp}}=8$	41.62	-8.5	45.49	-0.0

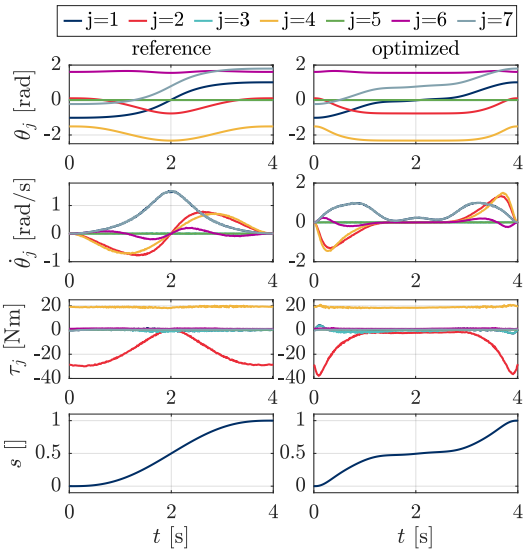

Fig. 4. The measured motion profiles $\theta_j(t)$, joint torques $\tau_j(t)$, and path profile $s(t)$ for both the reference poly5 (on the left) and the most optimal profile with $n_{\mathrm{dp}} = 8$ (on the right) of Task 1.

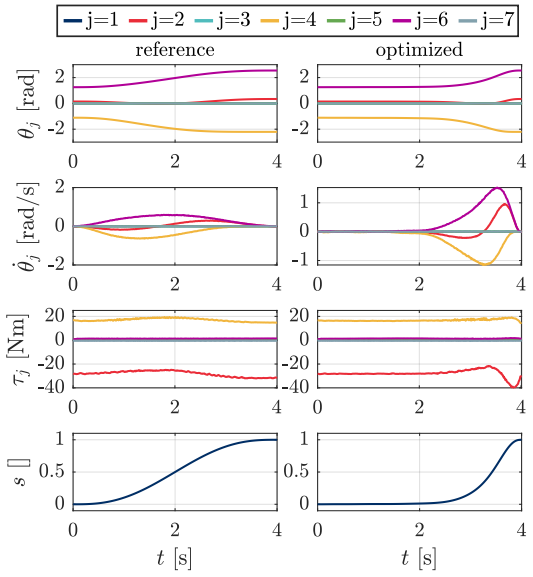

Fig. 5. The measured motion profiles $\theta_j(t)$, joint torques $\tau_j(t)$, and path profile $s(t)$ for both the reference poly5 (on the left) and the most optimal profile with $n_{\text{dp}} = 8$ (on the right) of Task 2.

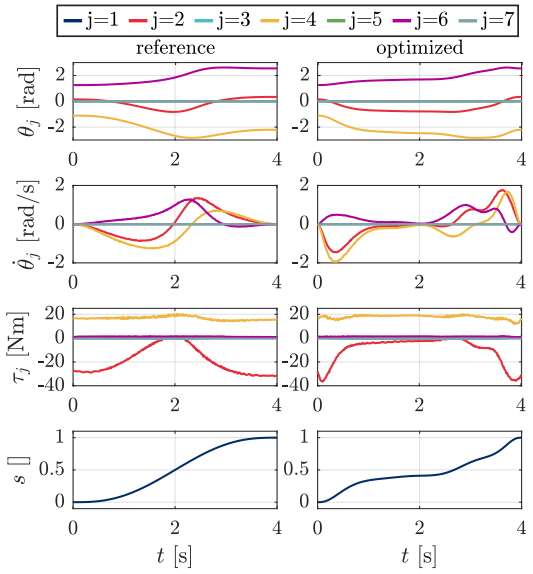

Fig. 6. The measured motion profiles $\theta_j(t)$, joint torques $\tau_j(t)$, and path profile $s(t)$ for both the reference poly5 (on the left) and the most optimal profile with $n_{\text{dp}} = 8$ (on the right) of Task 3.

achieved minimum (green dot) is the global minimum. While visual detection is no longer feasible, a similar pattern is anticipated for higher-dimensional problems, a belief reinforced by the fact that in Tab. II, the objective value converges to a fixed value.

V. Conclusions

In this paper, the Chebyshev basis was employed to optimize the motion profile of a path-constrained motion task. The kinematic and dynamic models were first detailed, followed by introducing the fixed-path and rescaled Chebyshev basis. Subsequently, constraints were incorporated, culminating in the final optimization problem formulation. The utilization of the Chebyshev basis and linearized constraints resulted in a compact feasible design space and smooth objective function, which increases the likelihood of revealing the global optimum, even with fast gradient-based algorithms. The feasibility of the method was validated through measurements on the actual setup for distinct motion tasks, yielding RMS torque difference of up to -15%. Moreover, it was shown that a substantial portion of the savings potential could be attained with as few as two design parameters. By opting for an approach based on discrete end-effector path data, the method can be readily applied to existing machines lacking a mathematical description of the end-effector trajectory, thus demonstrating the method's broad applicability.

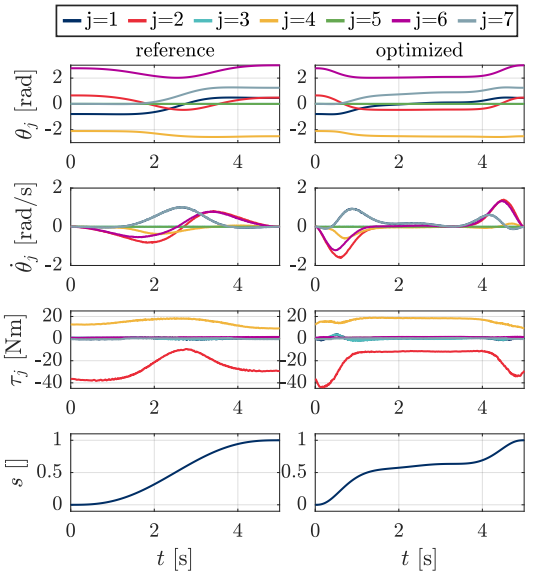

Fig. 7. The measured motion profiles $\theta_j(t)$, joint torques $\tau_j(t)$, and path profile $s(t)$ for both the reference poly5 (on the left) and the most optimal profile with $n_{\text{dp}} = 8$ (on the right) of Task 4.

Acknowledgment

This research was funded by the Research Foundation Flanders (FWO) via three PhD grants [1S88120N, 1SC2525N, G0B1123N] and a concurrent travel grant [V402023N], and was partially developed within the Laboratory for Big Data, IoT, Cyber Security (LABIC) funded by Friuli Venezia Giulia region, Italy, and the Laboratory for Artificial Intelligence for Human-Robot Collaboration (AI4HRC) funded by Fondazione Friuli, Italy

References

[1] EuropeanCommission, "Commisson Regulation (EU) 2019/1781," *Official Journal of the European Union*, 2019.

[2] G. Carabin, E. Wehrle, and R. Vidoni, "A Review on Energy-Saving Optimization Methods for Robotic and Automatic Systems," *Robotics*, vol. 6, no. 4, p. 39, Dec. 2017.

[3] R. De Laet, N. Van Oosterwyck, L. Scalera, A. Cuyt, A. Gasparetto, and S. Derammelaere, "Energy-efficient motion planning for robotic systems using polynomials in the chebyshev basis," *Robotics and Autonomous Systems*, vol. 192, p. 105051, 2025.

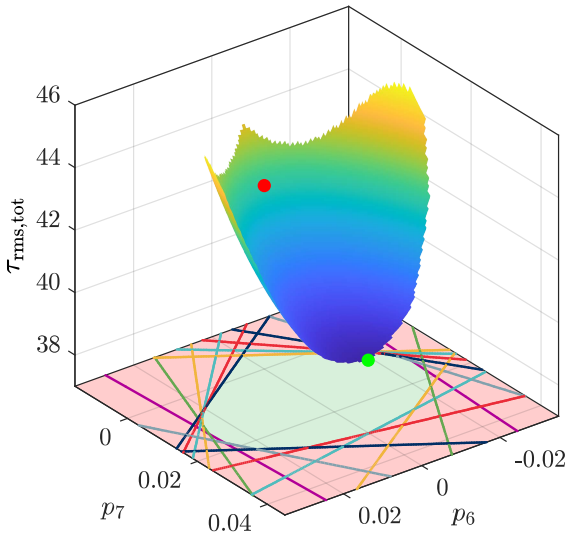

Fig. 8. The objective function $\tau_{\text{rms,tot}}$ of Task 1 for $n_{\text{dp}} = 2$, in conjunction with the 20 linear inequality constraints $\dot{s}|_{t=t_l} \geq 0$, illustrates the bounded feasible design space (green shaded area). The starting (red) and ending (green) points of the optimization algorithm are added, which shows the detection of the global optimum.

[4] G. Trigatti, P. Boscariol, L. Scalera, D. Pillan, and A. Gasparetto, "A look-ahead trajectory planning algorithm for spray painting robots with non-spherical wrists," in *Mechanism Design for Robotics*, A. Gasparetto and M. Ceccarelli, Eds. Cham: Springer International Publishing, 2019, pp. 235–242.

[5] T. Lipp and S. Boyd, "Minimum-time speed optimisation over a fixed path," *International Journal of Control*, vol. 87, no. 6, pp. 1297–1311, Jun. 2014.

[6] R. Wang, Y. Xie, X. Chen, and Y. Li, "Path-Constrained Time-Optimal Motion Planning for Robot Manipulators With Third-Order Constraints," *IEEE/ASME Transactions on Mechatronics*, pp. 1–12, 2023.

[7] S. D. Timar, R. T. Farouki, T. S. Smith, and C. L. Boyadjieff, "Algorithms for time–optimal control of CNC machines along curved tool paths," *Robotics and Computer-Integrated Manufacturing*, vol. 21, no. 1, pp. 37–53, Feb. 2005.

[8] M. S. Paing, E. W. Nshama, and N. Uchiyama, "A Kinematically Constrained Reparameterization Approach to Optimal Time and Jerk Motion of Industrial Machines," *IEEE Access*, vol. 9, pp. 97 843–97 854, 2021.

[9] M. S. Paing and N. Uchiyama, "A spline-based approach to smooth and time-optimal trajectory generation for CNC machines with guaranteed kinematic constraints," *The International Journal of Advanced Manufacturing Technology*, vol. 121, no. 5, pp. 3385–3398, Jul. 2022.

[10] Q. Zhang, S.-R. Li, and X.-S. Gao, "Practical smooth minimum time trajectory planning for path following robotic manipulators," in *2013 American Control Conference*, Jun. 2013, pp. 2778–2783, iSSN: 2378-5861.

[11] N. J. M. van Dijk, N. van de Wouw, H. Nijmeijer, and W. C. M. Pancras, "Path-Constrained Motion Planning for Robotics Based on Kinematic Constraints," in *Proceedings of the ASME 2007 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, vol. 8. American Society of Mechanical Engineers Digital Collection, Sep. 2009, pp. 1071–1080.

[12] S. Macfarlane and E. Croft, "Jerk-bounded manipulator trajectory planning: design for real-time applications," *IEEE Transactions on Robotics and Automation*, vol. 19, no. 1, pp. 42–52, Feb. 2003.

[13] T. Liu, J. Cui, Y. Li, S. Gao, M. Zhu, and L. Chen, "Time-Optimal Asymmetric S-Curve Trajectory Planning of Redundant Manipulators under Kinematic Constraints," *Sensors*, vol. 23, no. 6, p. 3074, Jan. 2023.

[14] Z. Shiller, "Time-Energy Optimal Control of Articulated Systems With Geometric Path Constraints," *Journal of Dynamic Systems, Measurement, and Control*, vol. 118, no. 1, pp. 139–143, Mar. 1996.

[15] O. Wigstrom, B. Lennartson, A. Vergnano, and C. Breitholtz, "High-Level Scheduling of Energy Optimal Trajectories," *IEEE Transactions on Automation Science and Engineering*, vol. 10, no. 1, pp. 57–64, Jan. 2013.

[16] Q.-C. Pham and O. Stasse, "Time-Optimal Path Parameterization for Redundantly Actuated Robots: A Numerical Integration Approach," *IEEE/ASME Transactions on Mechatronics*, vol. 20, no. 6, pp. 3257–3263, Dec. 2015.

[17] H. Pham and Q.-C. Pham, "A New Approach to Time-Optimal Path Parameterization Based on Reachability Analysis," *IEEE Transactions on Robotics*, vol. 34, no. 3, pp. 645–659, Jun. 2018.

[18] D. Verscheure, B. Demeulenaere, J. Swevers, J. De Schutter, and M. Diehl, "Time-Optimal Path Tracking for Robots: A Convex Optimization Approach," *IEEE Transactions on Automatic Control*, vol. 54, no. 10, pp. 2318–2327, Oct. 2009.

[19] T. Zhang, M. Zhang, and Y. Zou, "Time-optimal and Smooth Trajectory Planning for Robot Manipulators," *International Journal of Control, Automation and Systems*, vol. 19, no. 1, pp. 521–531, Jan. 2021.

[20] Y. Wen, H. He, A. Julius, and J. T. Wen, "Motion Profile Optimization in Industrial Robots using Reinforcement Learning," in *2023 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*, Jun. 2023, pp. 1309–1316, iSSN: 2159-6255.

[21] S. Riazi, K. Bengtsson, R. Bischoff, A. Aurnhammer, O. Wigström, and B. Lennartson, "Energy and peak-power optimization of existing time-optimal robot trajectories," in *2016 IEEE International Conference on Automation Science and Engineering (CASE)*, Aug. 2016, pp. 321–327, iSSN: 2161-8089.

[22] N. Van Oosterwyck, A. Ben Yahya, A. Cuyt, and S. Derammelaere, "CAD Based Trajectory optimization of PTP Motions using Chebyshev Polynomials," in *2020 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*, Jul. 2020, pp. 403–408, iSSN: 2159-6255.

[23] N. Van Oosterwyck, F. Vanbecelaere, M. Haemers, D. Ceulemans, K. Stockman, and S. Derammelaere, "CAD Enabled Trajectory optimization and Accurate Motion Control for Repetitive Tasks," in *2019 IEEE 15th International Conference on Control and Automation (ICCA)*, Jul. 2019, pp. 387–392, iSSN: 1948-3457.

[24] Y.-L. Hsu, M.-S. Huang, and R.-F. Fung, "Energy-saving trajectory planning for a toggle mechanism driven by a PMSM," *Mechatronics*, vol. 24, no. 1, pp. 23–31, Feb. 2014.

[25] N. Van Oosterwyck, F. Vanbecelaere, F. Knaepkens, M. Monte, K. Stockman, A. Cuyt, and S. Derammelaere, "Energy optimal point-to-point motion profile optimization," *Mechanics Based Design of Structures and Machines*, pp. 1–18, Aug. 2022.

[26] G. Berselli, F. Balugani, M. Pellicciari, and M. Gadaleta, "Energy-optimal motions for Servo-Systems: A comparison of spline interpolants and performance indexes using a CAD-based approach," *Robotics and Computer-Integrated Manufacturing*, vol. 40, pp. 55–65, Aug. 2016.

[27] K. K. Ayten, M. N. Sahinkaya, and A. Dumlu, "Optimum Trajectory Generation for Redundant/Hyper-Redundant Manipulators," *IFAC-PapersOnLine*, vol. 49, no. 21, pp. 493–500, Jan. 2016.

[28] J. Denavit and R. S. Hartenberg, "A Kinematic Notation for Lower-Pair Mechanisms Based on Matrices," *Journal of Applied Mechanics*, vol. 22, no. 2, pp. 215–221, Jun. 1955.

[29] C. Gaz, M. Cognetti, A. Oliva, P. Robuffo Giordano, and A. De Luca, "Dynamic Identification of the Franka Emika Panda Robot With Retrieval of Feasible Parameters Using Penalty-Based Optimization," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 4147–4154, Oct. 2019.

[30] F. Lozer, L. Scalera, P. Boscariol, and A. Gasparetto, "Planning optimal minimum-jerk trajectories for redundant robots," *Robotics and Autonomous Systems*, vol. 192, p. 105049, 2025.