CAD Based Trajectory Optimization of PTP Motions using Chebyshev Polynomials

Nick Van Oosterwyck[1,3], Abdelmajid Ben yahya[1], Annie Cuyt[2], Stijn Derammelaere[1]

Abstract— **Trajectory optimization of position-controlled servo systems driving repetitive tasks is very appealing in research as it allows for more performant and efficient machines without any additional investment in hardware. In particular, monoactuator systems with position-dependent system properties have received much attention in the literature as they represent a large part of mechatronic systems and can easily be modeled with existing CAD packages. Although techniques for optimization of the position profile are well documented, most literature is based on the heuristic and iterative optimization of polynomial and piecewise position curves. This results in locally optimal solutions and long calculation times. In this context, the purpose of the present paper is twofold. On the one hand, this paper outlines the theoretical framework for a fast trajectory optimization approach of single degree of freedom (1-DOF) systems with Chebyshev polynomials. On the other hand, a comparison of these solutions with results obtained using state-of-the-art techniques is provided. To do so, system property data is extracted with CAD motion simulations and converted into closed mathematical descriptions using polynomial interpolation. By combining these polynomial system models with a reconditioning and scaling of the trajectory definition, a novel problem formulation is obtained, which allows for fast gradient-based optimization techniques while limiting the risk of getting stuck in a local optimum. In this study, it was found that although substantial savings are achieved with state-of-the-art optimization methods, our proposed method with Chebyshev polynomials results in an extra saving potential of up to 6.7%. The findings also reveal a significant reduction in computational complexity. The latter not only supports the applicability of the proposed technique but also has implications for new system designs where real-time trajectory optimization is of interest.**

I. INTRODUCTION

The energy efficiency of industrial machinery is becoming a topic of primary importance due to stricter government regulations and economic considerations. This increasing interest has heightened the need for new techniques that have lower energy demands without compromising the system performance. Statistics reveal that electrical motors are responsible for about 40% of overall power consumption, which indicates that there are major savings to be made in this field [1], [2]. For instance, previous research has demonstrated that acquiring new machinery with existing well-established energy-conserving technologies results in savings of approximately 11–18% [1].

[1]Department of Electromechanics, CoSys-Lab, University of Antwerp, Groenenborgerlaan 171, 2020 Antwerpen, Belgium

[2]Department of Mathematics and Computer Science, University of Antwerp, Middelheimlaan 1, 2020 Antwerpen, Belgium

[3]Corresponding author: nick.vanoosterwyck@uantwerpen.be

Research funded by a PhD grant of the Research Foundation Flanders (FWO) [1S88120N].

Nevertheless, the adoption of new equipment entails certain costs, which hampers the wide spread of these innovations. Trajectory optimization on the other hand, is an example of a cost-effective alternative [3] with a wide application area, as it requires no additional investment in hardware. Moreover, the fact that machine users very often only define the start point, endpoint and duration of a motion task implies that the position function $\theta(t)$ between those two fixed points can effectively be optimized.

Many recent studies regarding trajectory optimization have paid attention to the energy dissipation in the electric motor in order to reduce the consumed electrical energy and thus enhance the system sustainability. However, apart from these electrical savings, reducing losses also has the beneficial effect that it may prevent the motor from overheating, which is one of the main reasons for motor failures (25%) [4]. Therefore, energy-optimal motion planning can provide a solution in manufacturing lines where productivity and reliability are of utmost importance.

Given the tendency to evolve from a monoactuator driving all machine components towards dedicated positioners for each machine movement [5], it is evident that one machine can contain numerous trajectory optimization opportunities. Moreover, by controlling and optimizing each movement separately, a flexible and reconfigurable machine is obtained that is able to cope with the ever-growing production needs [3].

A. Related Work

Within this scenario, several techniques for trajectory optimization have been proposed in multiple disciplines. An overview of this is covered in survey [6] and books [7], [8]. In general, all these optimization methods can be divided into two main categories: (1) *indirect* and (2) *direct* optimization techniques.

On the one hand, an indirect method uses calculus of variations and is based on *Pontryagin's Maximum Principle* [8]. Although there have been some promising results [9], [10], it tends to be abandoned recently due to the small convergence area and difficulties incorporating constraints [11]. Thus, it delivers impractical solutions.

On the other hand, direct approaches transform the optimization problem into a nonlinear programming problem (NLP), which can be solved with well known numerical techniques [8]. These methods are fundamentally different from indirect approaches as the state and/or control are discretized and parameterized. As indicated in the following paragraphs, they have proven successful for several complex applications.

For 3D robotic manipulators, *interval analysis* was employed in [12] and [13] to determine global optimal trajectories in terms of respectively time and jerk. In [14], interval procedures were applied in order to provide an estimate of the global optimum while incorporating torque constraints. Nevertheless, they do not consider the energy efficiency of the system.

In [15], [16] and [11], a multiobjective optimization was established using *Sequential Quadratic Programming* (SQP), in which time and root mean square (RMS) jerk/torque were considered simultaneously. In addition, [17] employs the same optimization technique but considers the input electrical energy.

However, the objective functions in these works ensuing from piecewise trajectories are characterized by many local minima, causing the risk of getting stuck in a suboptimal solution. For instance, [12] mentions an extra savings potential of 18% for the global solution compared to local solutions when piecewise cubic splines are employed. Moreover, as indicated by [18], the results of an SQP method are greatly influenced by the selected starting points, which are to be chosen arbitrarily.

Further, (meta-)heuristic methods relying on randomized optimization iterations, such as *genetic algorithms* (GA) [18], [19], [20], [21] and *generalized pattern search* (GPS) [22] are popular for trajectory optimization given their flexibility to solve many kinds of problems. Yet, the stochastic property implies there is a risk of getting stuck in a local optimum, even after a prolonged optimization time. In addition, [19] and [20] show that heuristic methods require long calculation time as ad hoc solve times of up to one hour were reported. Further, these techniques require tuning of the algorithm settings, which hinder an efficient solution.

As an alternative, [23] describes a convex path tracking approach for robots, which allows optimization based on *interior point* methods. However, an analytic description of the system dynamics is required, which is not easily obtained for many industrial 1-DOF systems. To avoid this obstacle, [3] and [9] focus on systems with constant inertia. Nonetheless, to cover the majority of machine applications, it is essential to consider varying properties such as position-dependent inertias $J(\theta)$. Therefore, [21] and [22] use the often already available CAD models to extract data samples of critical position-dependent system properties.

B. Method

In light of the considerations mentioned above, a CAD-based method for computing energy-optimal PTP (Point-to-Point) trajectories of single DOF mechanisms is presented in [21] and [22]. However, due to the classic trajectory descriptions and stochastic property of the employed algorithm, the obtained results can be improved. Therefore, building upon the aforementioned results, this paper is characterized by the following features:

- Similarly to [21], [22], CAD motion simulations are employed to extract discrete samples of system properties such as inertia J and load torque T_l. However, differently from the aforementioned literature, the property data is translated into closed mathematical descriptions using polynomial curve fitting in order to obtain an analytic model of the system, allowing a fast evaluation of the objective function.
- An orthogonal Chebyshev polynomial is selected as a position profile, instead of classic polynomials [21] or cubic splines [22], in an attempt to increase the tool robustness to get stuck in local minima. As these Chebyshev polynomials are defined on the interval [-1,1], the corresponding system equation should be altered accordingly.
- For the objective function, a torque-based approach is employed where the RMS torque is minimized in order to reduce the system's energy demand. Furthermore, our method is illustrated by a practical example of an industrial pick-and-place unit, which allows comparing the results with the energy savings achieved in [21].

In general, it will be shown how Chebyshev polynomials combined with analytic property descriptions outperform classic heuristic methods in terms of both achieved savings and solve time.

II. System Modeling

The dynamics of a 1-DOF mechanism can be described by means of the torque equation in (1). With reference to Fig.1, let us define $\theta = \theta(t)$ as the Lagrangian coordinate which describes the angular position of the main driving axis as a function of time t, $T_m(\theta)$ as the driving torque applied to this main axis, $T_l(\theta)$ as the load torque, J_m as the motor inertia and $J_l(\theta)$, $J(\theta)$ as the reduced moment of inertia of respectively load and complete system:

$$T_m(t) = T_l(\theta) + \frac{1}{2}\frac{\mathrm{d}J(\theta)}{\mathrm{d}\theta}(\dot{\theta})^2 + J(\theta)\ddot{\theta}. \qquad (1)$$

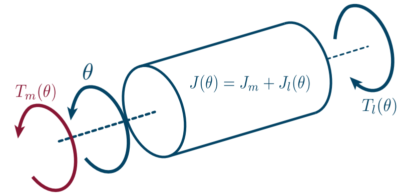

Fig. 1. Simplified model of a single axis driven mechanism with driving torque $T_m(\theta)$, inertia $J(\theta)$ and load torque $T_l(\theta)$.

In this paper the optimization procedure is illustrated by applying it to an industrial pick- and place unit (Fig. 2) that is expected to perform repetitive movements between start point A with related angular position θ_A at time instant t_A and endpoint B with angular position θ_B at time instant t_B (Fig. 5). Thus the system has a motion time $\Delta t = t_B - t_A$.

For an efficient optimization, it is essential to obtain an analytical description of the positional system properties

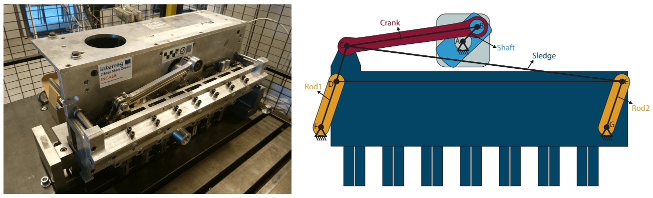

Fig. 2. Experimental set-up (left) and schematic overview (right) of the pick- and place unit.

$J(\theta)$ and $T_l(\theta)$. Therefore, the following passages provide a method to obtain a mathematical function of these critical properties.

A. Extracting System Properties from CAD Models

Due to all the position varying parameters in the torque equation, identification of the highly nonlinear differential equation (1) is not straightforward. Fortunately, machine builders design their machines in 3D CAD multibody software. One could try to determine the driving torque $T_m(t)$ for a certain position profile $\theta(t)$ with a single CAD simulation. However, employing them as such in an iterative optimization routine, intolerably increases the computational burden.

Therefore, it is convenient to use the available CAD model more thoughtfully. For this reason, [21] and [22] describe a technique to derive the position dependency of critical parameters inertia J and load torque T_l, based on three CAD motion simulations.

The motion simulations deliver n_s samples of inertia $\mathbf{J} = [J_1, \ldots, J_{n_s}]^T$, load torque $\mathbf{T}_l = [T_{l,1}, \ldots, T_{l,n_s}]^T$ and corresponding angle query points $\boldsymbol{\theta} = [\theta_1, \ldots, \theta_{n_s}]^T$, where $\theta_1 = \theta_A$ and $\theta_{n_s} = \theta_B$. In what follows, the property vectors $\mathbf{J}$ and $\mathbf{T}_l$ are referred to by the generic property vector $\mathbf{Y} = [Y_1, \ldots, Y_{n_s}]^T$, allowing the method to be extended with extra properties. For more details on the procedure, the reader is referred to [21].

B. Polynomial Fitting of the Mechanical Properties

The property description obtained by the simulations described above will be a collection of separate samples. When envisaging an efficient optimization procedure, it is essential to obtain an explicit mathematical description of each parameter in (1). For this purpose, polynomial interpolation is applied to the discrete data samples.

However, when applying polynomial curve fitting to determine the polynomial property description $Y(\theta)$, a linear system with Vandermonde matrix needs to be solved, which can cause numeric problems for high-order polynomials. In order to improve the numerical properties of the method and to produce a more reliable fit, the angle vector $\boldsymbol{\theta}$ is rescaled to the vector $\boldsymbol{\phi}$ in the interval $[-1, 1]$:

$$\phi_i = \frac{2(\theta_i - \theta_1)}{\theta_{n_s} - \theta_1} - 1, \quad i = 1, .., n_s. \tag{2}$$

Subsequently, polynomial interpolation is applied to the vectors $\mathbf{Y}$ with query points $\boldsymbol{\phi}$, where a polynomial $Y(\phi)$ (3) of degree d with coefficients $\mathbf{a} = [a_0, a_1, \ldots, a_d]^T$ is determined in a least-squares sense:

$$Y(\phi) \approx \sum_{i=0}^{d} a_i \phi^i, \quad \phi \in [-1, 1]. \tag{3}$$

The problem is practically solved by applying the MATLAB function *polyfit*. The trajectory is defined by $\phi = \phi(t)$ with $t \in [t_A, t_B]$, $\phi(t_A) = -1$ and $\phi(t_B) = 1$ (Fig. 5), which holds the following relation with the original trajectory description $\theta(t)$:

$$\begin{aligned} \phi &= \frac{2}{(\theta_B - \theta_A)}\,\theta - \frac{(\theta_B + \theta_A)}{(\theta_B - \theta_A)} \\ &= C_1\,\theta + C_2. \end{aligned} \tag{4}$$

Moreover, as the property description $Y(\phi)$ is now defined on the rescaled interval $[-1, 1]$, the following relationship holds with regard to the derivative properties:

$$\frac{\mathrm{d}Y(\phi)}{\mathrm{d}\phi} = \frac{1}{2}(\theta_B - \theta_A)\frac{\mathrm{d}Y(\theta)}{\mathrm{d}\theta} = C_3\,\frac{\mathrm{d}Y(\theta)}{\mathrm{d}\theta}. \tag{5}$$

Scale factors C_1, C_2 and C_3 have been defined for the purpose of the following sections.

C. Convergence Analysis

It should be noted that higher degree polynomials can be oscillatory between the data points and can lead to an over-fitted model. In addition, a higher degree makes the calculations more complex as the functions consist of more terms and thus, requires more symbolic operations. Therefore, the degree d must be kept limited in order to prevent this phenomenon. Nevertheless, selecting the degree d too low can lead to a poor fit of the data.

It becomes apparent that an optimal degree d_c imposes itself where the data must be approximated accurately while keeping the polynomial degree as low as possible. This optimal degree d_c is obtained with the aid of a convergence analysis. The quality of the fit is determined by means of the Euclidean or L_2-norm (square root of the sum of squares) of the residuals $\mathbf{r}_Y = \mathbf{Y} - Y(\boldsymbol{\phi})$,which is an efficient and accurate technique for these kinds of fitting problems:

$$||\mathbf{r}_Y||_2 = \sqrt{\sum_{i=1}^{n_s} r_i^2} = \sqrt{\sum_{i=1}^{n_s} (Y_i - Y(\phi_i))^2}. \tag{6}$$

The degree d is systematically increased in order to obtain a better representation, while the L_2-norm is evaluated. As indicated in Fig. 3, from a certain convergence point d_c onwards, increasing the degree d will not add any significant improvements.

Fig. 4 shows the values of inertia $\mathbf{J}$ and load torque $\mathbf{T}_l$ data vectors as a function of the rescaled crank position ϕ for the considered mechanism, highlighting the high variability of both parameters across the range of motion.

Additionally, the corresponding fitted polynomials $J(\phi)$ and $T_l(\phi)$ are presented, which demonstrates a very strong similarity between the obtained CAD data $\mathbf{Y}$ and analytical

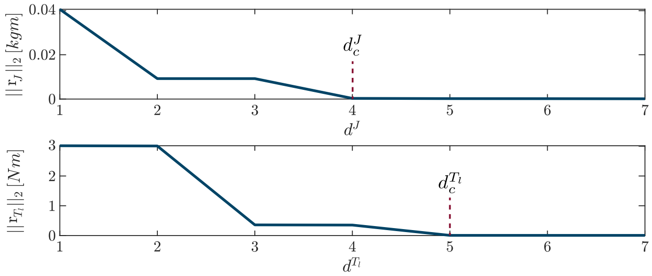

Fig. 3. Convergence analysis of system properties $J(\phi)$ (top) and $T_l(\phi)$ (bottom) with their corresponding convergence points d_c^J and $d_c^{T_l}$

descriptions $Y(\phi)$. Due to the continuity of the obtained property descriptions $Y(\phi)$, the property variations $\frac{\mathrm{d}Y(\phi)}{\mathrm{d}\phi}$ can be easily derived as well.

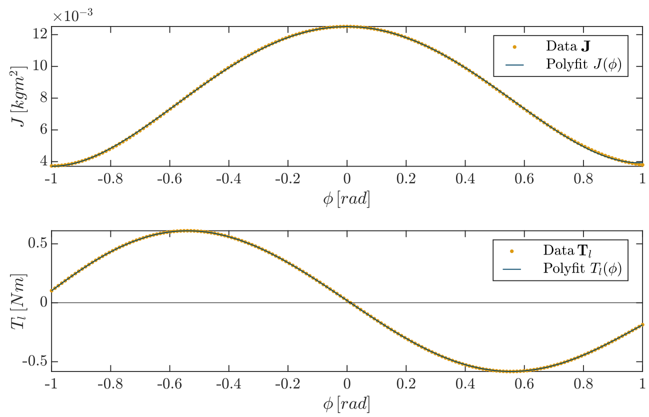

Fig. 4. Values of data vectors $\mathbf{J}$ and $\mathbf{T}_l$ and corresponding polynomials $J(\phi)$ and $T_l(\phi)$.

III. Optimization Approach

Every optimization routine uses certain design parameters which are adjusted by the algorithm in order to converge towards an optimal solution. For trajectory optimization, these design variables define the shape of the position profile between start point A and endpoint B (Fig.5).

A. Trajectory Definition

This paper studies the use of Chebyshev polynomials for the position profile, which consists of a sequence of orthogonal Chebyshev polynomials defined on the interval $[-1, 1]$. One usually distinguishes between Chebyshev polynomials of the first, second, third and fourth kind, which are named after the Russian mathematician Pafnuty Chebyshev. Here, will be focused on the most commonly used Chebyshev polynomials of the first kind T_n, defined by the recurrence relation:

$$\begin{aligned} T_0(x) &= 1, \\ T_1(x) &= x, \\ T_{n+1}(x) &= 2xT_n(x) - T_{n-1}(x). \end{aligned} \tag{7}$$

However, the position profile $\phi(t)$ with time variable t is defined in the finite range $[t_A, t_B]$ (Fig. 5). To use $T_n(x)$ as a representation of the position profile, a linear transformation into the range $[-1, 1]$ of x is required [24]:

$$\begin{aligned} t &= \frac{1}{2}(t_B - t_A)x + \frac{1}{2}(t_B + t_A) \\ &= C_4 x + C_5. \end{aligned} \tag{8}$$

Thus, the final trajectory description $\phi(x)$ of degree n with optimizable coefficients $\mathrm{p} = [p_0, p_1, \ldots, p_n]^T$ is obtained (Fig. 5), hereafter referred to as *cheb"n"*:

$$\phi(x) = \sum_{i=0}^{n} p_i T_i(x), \quad x \in [-1, 1]. \tag{9}$$

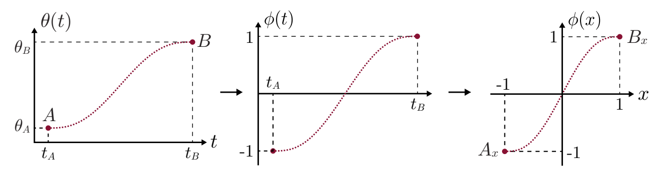

Fig. 5. Original (left), position rescaled (middle) and time rescaled (right) profile of the trajectory between start point A and endpoint B

B. Rescaling Torque Equation

Due to the rescalings (4), (5) and (8), employing the typical torque equation (1) as such, leads to improper results. Therefore, this equation is extended (10) with the scaling factors C_1, C_3 and C_4 defined above, where $\phi = \phi(x)$ and $\dot{\phi}$, $\ddot{\phi}$ are the first and second derivatives with respect to x:

$$T_m(x) = T_l(\phi) + \frac{1}{2}\frac{\mathrm{d}J(\phi)}{\mathrm{d}\phi}\frac{1}{C_3}\left(\frac{\dot{\phi}}{C_4.C_1}\right)^2 + J(\phi)\frac{\ddot{\phi}}{C_4^2.C_1}. \tag{10}$$

This new system equation ensures the system dynamics are equally scaled and the minima are not altered.

C. Constraints

Concerning the constraints, the mechanism's start point A and endpoint B are assumed to be defined by the machine builder and process-related tasks. Moreover, as PTP movements are covered, the velocity $\dot{\phi}$ and acceleration $\ddot{\phi}$ are assumed to be zero in the start and endpoint. Thus, with respect to Fig. 5, the following six constraints need to be incorporated in the trajectory formulation:

$$\begin{aligned} \phi(-1) = -1 \quad &, \quad \dot{\phi}(-1) = 0 \quad , \quad \ddot{\phi}(-1) = 0, \\ \phi(1) = 1 \quad &, \quad \dot{\phi}(1) = 0 \quad , \quad \ddot{\phi}(1) = 0. \end{aligned} \tag{11}$$

Referring to (9), and by incorporating the trajectory constraints, the lower degree coefficients $[p_0, ..., p_5]^T$ can be written as a function of the remaining coefficients $[p_6, ..., p_n]^T$, such that $n-5$ degrees of freedom (DOF) are kept available for the optimization algorithm [20]. Thus, the final position profile is solely determined by the optimization parameter vector $\mathbf{o} = [p_6, ..., p_n]^T$.

D. Objective

Once the rescaled position profile (9) and system dynamics (10) are defined, a suitable objective function can be introduced. In this paper, a torque based approach is selected in which the root mean square (RMS) torque T_{rms} (12) is minimized in order to reduce joule losses and increase energy efficiency:

$$T_{rms} = \sqrt{\frac{1}{T}\int_{t_A}^{t_B} T_m(t)^2 \, \mathrm{d}t} = \sqrt{\frac{1}{T}\int_{-1}^{1} T_m(x)^2 \, \mathrm{d}x}. \quad (12)$$

Note that determining the objective (12) on the original (1) and rescaled (10) torque equation yields the same results, due to the scaling factors used.

As the model is solely based on the mechanism dynamics, this torque based method has the advantage that it does not require any knowledge about the motor dynamics and related model constants [22], which are often not accurately known. Yet, they can be effectively employed for the envisaged high-dynamical systems where inertial loads are predominant [22].

Since a closed mathematical description of the torque equation (10) is obtained, the integral (12) can be calculated through an analytic integration. This approach yields much better results in terms of execution time, compared to a numerical integration procedure [15]. Moreover, the complete objective function is composed within the symbolic toolbox of the MATLAB environment allowing a very fast evaluation of the objective, which benefits the optimization time even more.

E. Optimization Algorithm

As an analytic description of the objective function is constructed, it is possible to select an optimization algorithm which exploits gradient information and is computationally superior to heuristic approaches [6]. As illustrated in Fig. 6, due to the use of orthogonal Chebyshev polynomials for the position profile $\phi(x)$, the trajectory optimization problem for this application is conditioned in a way that the probability of converging to a local minimum is diminished.

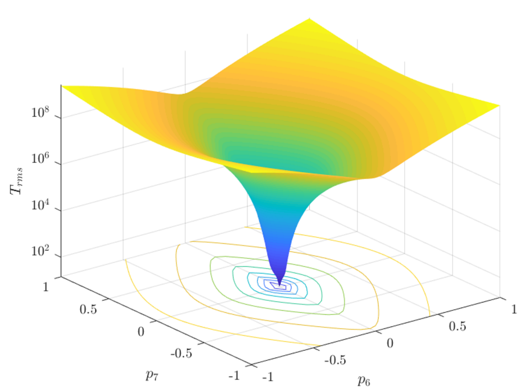

Fig. 6. Objective function of a cheb7 position profile with 2 DOF (p_6 and p_7).

Therefore, the resulting constrained nonlinear optimization problem is practically solved offline using the *fminunc* function within MATLAB. In this case, fminunc uses a BFGS (Broyden–Fletcher–Goldfarb–Shanno) quasi-Newton method, which represents the state-of-the-art in nonlinear programming and guarantees superlinear convergence [11].

The selection of the starting point for this iterative method has a major impact on the optimization result, especially when it is chosen arbitrarily [18]. The use of the Chebyshev basis $T_i(x)$ in representation (9) allows to initialize the optimization parameter vector at zero for the following reason. Under very mild assumptions on the trajectory function $\phi(x)$, one knows that the coefficients in its Chebyshev series expansion converge to zero [25]. We can safely assume some similar behavior for the coefficients p_i in (9).

At last, the optimized trajectory $\phi^*(x)$ and torque profile $T_m^*(x)$ are rescaled to the original task constraints, i.e., $\theta^*(t)$ and $T_m^*(t)$, so that they can be implemented in the motor drives and compared with other approaches which do not use any rescaling.

IV. Results

In order to assess the performance of the proposed optimization approach, a set of optimizations has been performed on the industrial pick-and-place unit depicted in Fig. 2. The mechanism is required to move between its start position θ_A of $0°$ and end position θ_B of $173.6°$ and has a motion time Δt of $73.5ms$. A fifth-degree polynomial is taken as the reference motion path for comparison purposes, hereafter referred to as *poly5*, which is the uniquely defined RMS jerk-optimal trajectory [22].

In Table I, the RMS torque T_{rms}, optimization solve times t_{sol}, and energy savings are presented. The results are compared with the optimal trajectory achieved in [21], which is a standard 17th-degree polynomial optimized with GA, referred to as *poly17*.

TABLE I
Saving potential achieved with Chebyshev polynomials.

Trajectory	T_{rms} **[Nm]**	**Savings [%]**	t_{sol} **[s]**
poly5	22.64	-	-
poly17 [21]	13.94	-38.4	500
cheb7	13.94	-38.6	0.07
cheb9	12.61	-44.3	25.9
cheb13	12.44	-45.1	6838

For example, when a Chebyshev polynomial with degree $n = 7$ is selected (cheb7), the proposed method already outperforms the results from [21] (poly17) by achieving similar energy savings, yet in a fraction of the computing time. The best results are obtained with a 13th degree Chebyshev polynomial (cheb13) where energy saving of -45% are reported. Thus, having an extra savings potential of 6.6%. However, due to the higher degree n, construction of the objective function is more time-consuming, which could hamper the applicability.

Therefore, also the results of the cheb9 trajectory are depicted, which turns out to be the convergence point and the optimal trade-off between energy savings and calculation time. Nevertheless, an additional energy reduction of 5.9% is found in a minimal optimization time.

At last, Fig. 7, shows the optimized position $\theta^*(t)$ and torque $T_m^*(t)$ profiles obtained when minimizing the RMS torque T_{rms}. It should be noticed that small variations in the trajectory can lead to very diverse torque profiles with very different energy demands, illustrating the importance of trajectory optimization even more. What can be clearly seen in this figure is that while the optimization is focused on minimizing the RMS torque T_{rms} and energy losses, the maximum torque T_{max} decreases as well, improving the life span of the machine.

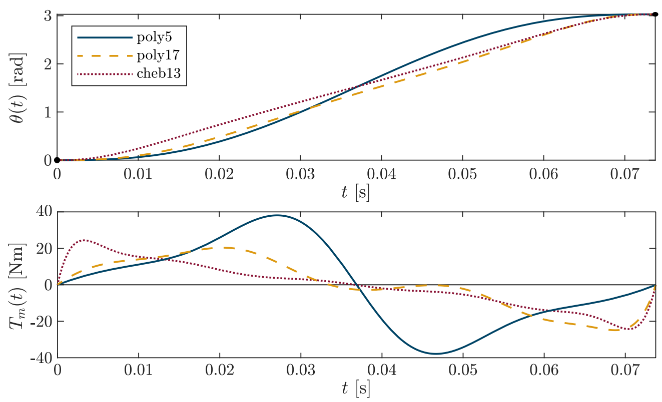

Fig. 7. Optimised position $\theta^*(t)$ and torque $T_m^*(t)$ profiles.

V. Conclusions

This study proposes a novel approach for trajectory optimization of PTP motions with Chebyshev polynomials. At first, system properties have been extracted with CAD motion simulations and converted into closed mathematical descriptions using polynomial interpolation. Subsequently, the trajectory is defined using Chebyshev polynomials and the system equation is rescaled accordingly. The latter enables the application of a fast gradient-based quasi-newton optimization algorithm. Finally, the numerical optimization results have been considered and compared with recent trajectory optimization results from [21].

The results clearly show that the proposed method outperforms previous trajectory optimization approaches and reveals an extra energy-saving potential of up to 6.7%. Moreover, due to the analytic construction of the objective and gradient-based optimization, the optimization time is significantly reduced with 94.8%, which benefits the applicability of the method and could allow real-time optimization of the trajectory.

References

[1] H. Bo, "Electric Motors – Alignment of Standards and Best Practice Programmes with APEC, Final Report," 2008.

[2] A. T. de Almeida, P. Fonseca, and P. Bertoldi, "Energy-efficient motor systems in the industrial and in the services sectors in the European Union: Characterisation, potentials, barriers and policies," *Energy*, vol. 28, no. 7, pp. 673–690, 2003.

[3] D. Richiedei and A. Trevisani, "Analytical computation of the energy-efficient optimal planning in rest-to-rest motion of constant inertia systems," *MECHATRONICS*, vol. 39, pp. 147–159, 11 2016.

[4] R. Saidur, "A review on electrical motors energy use and energy savings," *Renewable and Sustainable Energy Reviews*, vol. 14, no. 3, pp. 877–898, 2010.

[5] J. van Gerwen, "Electronic camming and gearing," *Assembly Automation*, vol. 19, no. 1, pp. 35–38, 1 1999. [Online]. Available: https://doi.org/10.1108/01445159910254136

[6] J. T. Betts, "A Survey of Numerical Methods for Trajectory Optimization," *Journal of Guidance, Control, and Dynamics*, vol. 21, no. 2, pp. 193–207, 1998.

[7] L. Biagiotti and C. Melchiorri, *Trajectory planning for automatic machines and robots*. Springer Science & Business Media, 2008.

[8] A. V. Rao, "Trajectory optimization: a survey," in *Optimization and optimal control in automotive systems*. Springer, 2014, pp. 3–21.

[9] J. S. Park, "Motion profile planning of repetitive point-to-point control for maximum energy conversion efficiency under acceleration conditions," *Mechatronics*, vol. 6, no. 6, pp. 649–663, 9 1996.

[10] Z. Shiller, "Time-energy optimal control of articulated systems with geometric path constraints," *Journal of Dynamic Systems Measurement and Control*, no. 118, pp. 139–143, 1996.

[11] T. Chettibi, H. Lehtihet, M. Hadda, and S. Hanchi, "Minimum cost trajectory planning for industrial robots," *European Journal of Mechanics - A/Solids*, vol. 23, no. 4, pp. 703–715, 2004.

[12] A. Piazzi and A. Visioli, "Global minimum-time trajectory planning of mechanical manipulators using interval analysis," *International Journal of Control*, vol. 71, no. 4, pp. 631–652, 1998.

[13] ——, "Global minimum-jerk trajectory planning of robot manipulators," *IEEE Transactions on Industrial Electronics*, vol. 47, no. 1, pp. 140–149, 2000.

[14] C. G. L. Bianco and A. Piazzi, "A Semi-Infinte Optimization Approach to Optimal Spline Trajectory Planning of Mechanical Manipulators," in *Semi-Infinite Programming*. Kluwer Academic Publisher, 2001, ch. 13, pp. 271–297.

[15] A. Gasparetto and V. Zanotto, "A new method for smooth trajectory planning of robot manipulators," *Mechanism and Machine Theory*, vol. 42, no. 4, pp. 455–471, 2007.

[16] ——, "Optimal trajectory planning for industrial robots," *Advances in Engineering Software*, vol. 41, no. 4, pp. 548–556, 2010. [Online]. Available: http://dx.doi.org/10.1016/j.advengsoft.2009.11.001

[17] M. Pellicciari, G. Berselli, and F. Balugani, "On Designing Optimal Trajectories for Servo-Actuated Mechanisms: Detailed Virtual Prototyping and Experimental Evaluation," *IEEE/ASME Transactions on Mechatronics*, vol. 20, no. 5, pp. 2039–2052, 2015.

[18] J. Huang, P. Hu, K. Wu, and M. Zeng, "Optimal time-jerk trajectory planning for industrial robots," *Mechanism and Machine Theory*, vol. 121, pp. 530–544, 2017. [Online]. Available: https://doi.org/10.1016/j.mechmachtheory.2017.11.006

[19] M. S. Huang, Y. L. Hsu, and R. F. Fung, "Minimum-energy point-to-point trajectory planning for a motor-toggle servomechanism," *IEEE/ASME Transactions on Mechatronics*, vol. 17, no. 2, pp. 337–344, 2012.

[20] 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, 2014.

[21] 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 ." *15th International Conference on Control and Automation (ICCA)*, p. 6, 2019.

[22] 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, 8 2016.

[23] 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, 2009.

[24] W. J. Thompson, "Chebyshev Polynomials: After the Spelling the Rest is Easy," *Computers in Physics*, vol. 8, no. 2, p. 161, 2013.

[25] H. Majidian, "On the decay rate of Chebyshev coefficients," *Applied Numerical Mathematics*, vol. 113, pp. 44–53, 2017. [Online]. Available: http://dx.doi.org/10.1016/j.apnum.2016.11.004