CAD Enabled Trajectory Optimization and Accurate Motion Control for Repetitive Tasks.

Nick Van Oosterwyck[1,4], Foeke Vanbecelaere[2,3], Michiel Haemers[1,2,3],
David Ceulemans[1], Kurt Stockman[2,3], Stijn Derammelaere[1,2,3]

***Abstract*— As machine users generally only define the start and end point of the movement, a large trajectory optimization potential rises for single axis mechanisms performing repetitive tasks. However, a descriptive mathematical model of the mechanism needs to be defined in order to apply existing optimization techniques. This is usually done with complex methods like virtual work or Lagrange equations. In this paper, a generic technique is presented to optimize the design of point-to-point trajectories by extracting position dependent properties with CAD motion simulations. The optimization problem is solved by a genetic algorithm. Nevertheless, the potential savings will only be achieved if the machine is capable of accurately following the optimized trajectory. Therefore, a feedforward motion controller is derived from the generic model allowing to use the controller for various settings and position profiles. Moreover, the theoretical savings are compared with experimental data from a physical set-up. The results quantitatively show that the savings potential is effectively achieved thanks to advanced torque feedforward with a reduction of the maximum torque by 12.6% compared with a standard 1/3-profile.**

I. Introduction

As the global energy demand will continue to rise and man's negative impact on global warming is known to be a fact, there is a strong desire to minimize the energy usage of industrial machinery [1]. While there is much research on energy efficiency improvements of machine actuators, a significant opportunity lies in optimization which does not require any adaptations or investments in the installed hardware. Trajectory optimization is an example of such a low-cost technique [2] that is very often applicable due to the fact that the exact position as a function of the time, or position function $\theta(t)$ (Fig. 1), in between these two points is very often not an issue for machine users.

To date, there is a lot of research on the 3D trajectory planning and optimization for multiple axis robot systems [3]. However, lots of repetitive point to point actions are driven by a single axis mechanism. Especially given the tendency, as indicated in [4], to evolve from a single actuator driving all machine components via mechanical transmission systems towards dedicated actuators for each machine movement. In many cases, these single axis driven mechanisms perform repetitive tasks and it is therefore certainly worthwhile to consider trajectory optimization since the effect will be perceptible every machine cycle.

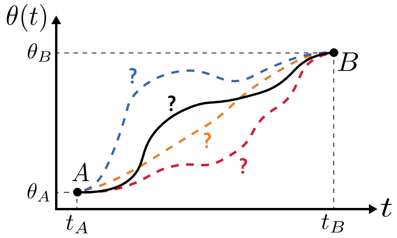

Fig. 1. Position function for the trajectory between starting point A and endpoint B

In order to optimize the position function $\theta(t)$, it is essential to describe its impact on the torque T_m necessary to drive the system [4]. Thus, it is essential to express the position dependency of the critical machine parameters. On the contrary, [5] and even more recent literature [2] on trajectory optimization solely consider simple mechanical systems with constant parameters inertia J, friction μ_k and load torque T_l. Nevertheless, as indicated in literature [4], [6], [7], it is essential to consider varying loads to cover the majority of machine applications, such as, but not limited to, position-dependent inertia's $J(\theta)$.

However, in those cases obtaining a description of the impact of the position function $\theta(t)$ on the torque T_m necessary to drive the system is challenging for machine builders and users [4]. For instance, [2], [5] assumes mechanical parameter values J, μ_k and T_l are given and known. In practice this will often not be the case, therefore [6] describes the use of a genetic algorithm to identify the model parameters such as inertia J and damping b, yet only for constant parameters. In [8], an attempt was made to determine the system properties based on Hamiltons principle and Lagrange multiplier. However, this approach is quite complex to be applicable by machine builders and very machine specific. Additionally, the majority of the machine builders design their machines in multibody 3D CAD packages. If the users of these software tools add the correct properties such as material density, friction parameters μ_k and flexibility b, CAD motion simulations can be used to determine the necessary driving torque T_m and position dependent system properties $J(\theta)$ and $T_l(\theta)$. In this paper, the procedure described in [4] will be used to enable machine builders to extract position dependent mechanical properties conveniently from CAD-motion simulations. However, in [4] no validation of the CAD model and achieved system properties on a real set-up were performed and therefore cannot reveal the potential

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

[2] Department of Electrical Energy, Metals, Mechanical Constructions and Systems, Ghent University campus Kortrijk, Kortrijk, Belgium

[3] Member of Flanders Make

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

savings of the optimization.

For trajectory optimization, various optimization algorithms exist, each with their strengths and weaknesses. Genetic algorithms (GA's) have proven their usefulness for trajectory optimization and their efficient and effective techniques ensure that these are widely used techniques in scientific and engineering areas. [9]. In contrast to gradient-based algorithms that do not search the entire design space, derivative-free algorithms like GA [10] must often sample a wide part of the design space in order to be successful. On the other hand, neuro-evolution techniques are known for their good training times and lower susceptibility to getting stuck in local maxima, especially in problems with high dimensionality [11]. In previous literature, genetic algorithms have been used for obstacle-avoiding robot trajectory planning [12] and trajectory planning in a 3D space [13], [14]. In [6], [15], [16], genetic algorithms were employed to determine the optimal coefficients of the trajectory polynomial of a single axis driven mechanism. However, only constant mechanical parameters were considered and therefore do not fully exploit the potential of the optimization. Furthermore, in [5], [12], the achieved savings of the optimization are based on theoretical assumptions and are not compared and validated on an experimental set-up.

Nevertheless, the energy savings as a result from performing trajectory optimization will only be achieved if the controller is capable of following the desired and optimized trajectory. An ineffective controller can lead to possible complete eradication of the achieved savings. In [17], [18], iterative learning control (ILC) was applied to reduce the tracking error. However, ILC is heuristic in nature and does not take the knowledge of the system into account.

On the other hand, a positive effect of the identification of the system parameters is the fact that the necessary driving torque can be calculated directly during the optimization. This information is very beneficial to program a feedforward controller which, bypasses low controller dynamics and ensures the machine accurately follows the desired trajectory. In literature [16], [15], [7] optimal trajectories were characterized and validated on an experimental set-up. Nonetheless, the control of the mechanism was allocated to standard PI(D)-controllers which require time-consuming controller tuning and may not be able to follow the desired trajectory. In this paper, the torque equation identified using CAD-motion simulations according to [4] will be used to program an accurate feedforward, enabling a reduction of the tracking error compared to standard robust PI-cascaded control by 93%.

The aim of the presented paper is to describe a generic procedure to optimize the trajectory for single axis driven mechanisms with position dependent properties and use this knowledge to design an accurate motion controller.

In Section II, the model is defined and a technique is elaborated to determine the position dependent system properties. In Section III a genetic algorithm is employed to optimize the point-to-point trajectory considering imposed constraints and different objectives. In Section IV, standard motion controllers are extended with the system information in order to create a more dynamic controller while reducing the tracking error. Finally, in Section V, the complete procedure is applied to the experimental set-up and the achieved savings are compared with a standard position profile. Furthermore, the motion controllers are implemented and evaluated on the set-up. In Section VI, the results are compared and discussed.

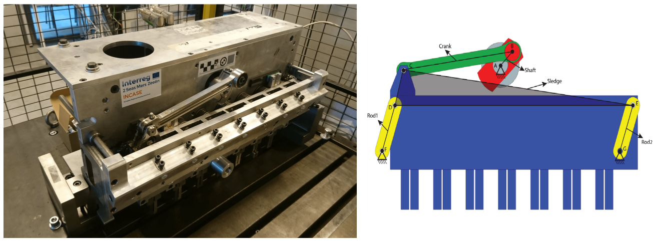

Fig. 2. Experimental set-up (left) and virtual CAD model (right) of the pick- and place unit used for validation.

II. Modeling and Extraction of System Properties

An industrial pick and place system used for validation of the optimization process is presented in Fig. 2. For optimization and control of the system, it is essential to have an adequate mathematical description of the machine properties. Therefore, the relationship between the shaft position θ and the necessary driving torque T_m needs to be known. This relation is represented by the torque equation, as explained in the following section.

A. Torque Equation

In the simplified model as shown in Fig. 3, all loads and inertias of the mechanism are related to the main driving axis resorting to the concept of reduced moment of inertia. On top of that comes the inertia of the motor shaft itself J_m which combines to the total inertia of the complete system. The load torque T_l contains both gravitational forces as well as external process powers that act on the mechanism. This approach permits to fit every possible mechanism with a known geometry to this model and allows to use a generic optimization approach.

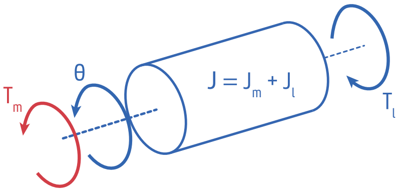

Fig. 3. Simplified model of a single axis driven mechanism with driving torque T_m, inertia J and load torque T_l.

Based on the concept of energy conservation, the driving torque can thus be written as in (1) with load torque T_l, inertia J, speed $\dot{\theta}$ and acceleration $\ddot{\theta}$ [4], [7], [19]. Note that $T_l(\theta)$ and $J(\theta)$ are based on the crank position θ. This equation holds for systems with negligible friction.

Furthermore, as indicated in [4], in such case torque-based approaches can be effectively employed.

$$T_m = T_l + \frac{1}{2}\frac{dJ}{d\theta}(\dot{\theta})^2 + J\ddot{\theta} \tag{1}$$

B. Extraction System Properties based on CAD Motion Simulations

The analytic determination of the system's dynamics for the identification of the driving torque T_m can be quite cumbersome resorting to methods of virtual work and Lagrange equations [20]. With CAD motion simulations, it is possible to determine the driving torque for a certain position profile. However, as these simulations require a lot of computational time, using them as such in an iterative optimization routine would intolerably increase the optimization time. Therefore, [4] describes a technique to derive the position dependency of critical parameters, such as inertia $J(\theta)$, based on a limited number of CAD motion simulations. These numerical results can subsequently be used by the optimization algorithm in order to define the impact of the position profile $\theta(t)$ on the necessary driving torque T_m.

The procedure (Fig. 4) requires three different motion simulations, each under different circumstances and settings, in order to determine the position dependency of inertia $J(\theta)$ and load torque $T_l(\theta)$. The position range is defined between the start (θ_A) and endpoint (θ_B). The simulation time t_{sim} results from this range and speed ω at which the simulation has been executed. The more advanced torque equation also takes friction and damping into account, although the described approach is limited to extracting inertia $J(\theta)$ and load torque $T_l(\theta)$. This means that before performing the three motion simulations, all friction and damping that was included in the motion environment must be turned off. The simulations are conducted as follows:

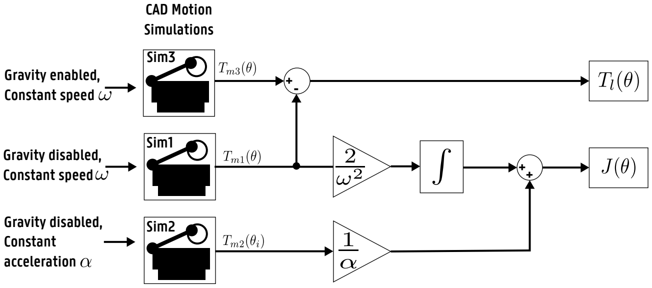

Fig. 4. Graphical overview of the procedure for extracting position dependent properties $J(\theta)$ and T_l based on three different CAD motion simulations with negligible friction.

- *Simulation 1*: While gravitational and external forces are disabled for the motion simulation, a constant speed ω is imposed upon the driving axis. Since there is no change in speed or acceleration, the resulting driving torque according to Equation (1) is:

$$T_{m1}(\theta) = \frac{1}{2}\frac{dJ(\theta)}{d\theta}(\omega)^2 \tag{2}$$

As a result, the inertia can be calculated as follows:

$$J(\theta) = \int_{\theta_A}^{\theta} \frac{2.T_{m1}(\theta)}{\omega^2} d\theta + C \qquad \theta \in [\theta_A, \theta_B] \tag{3}$$

Note that there still is an unknown constant of integration C.

- *Simulation 2*: Gravity is disabled and a constant acceleration α is selected. For this simulation, only the initial torque is essential since this will allow determining the initial inertia J_{init}. As the initial speed at the beginning of the simulation is zero, the torque equation (1) can be simplified as follows:

$$T_{m2}(\theta_A) = J_{init}\alpha \tag{4}$$

Equation (4) allows to calculate the constant of integration C (3):

$$J(\theta_A) = C = J_{init} = \frac{T_{m2}(\theta_A)}{\alpha} \tag{5}$$

- *Simulation 3*: The same constant speed ω as in simulation 1 is imposed and external and gravitational forces are enabled. The resulting driving torque according to (1) is:

$$T_{m3}(\theta) = T_l(\theta) + \frac{1}{2}\frac{dJ(\theta)}{d\theta}(\omega)^2 \tag{6}$$

Since the second term of this equation is known from simulation 1, the load torque T_l can be identified by:

$$T_l(\theta) = T_{m3}(\theta) - T_{m1}(\theta) \tag{7}$$

The knowledge of $J(\theta)$ and $T_l(\theta)$ allow describing the effect of the position θ on the driving torque T_m according to Equation (1).

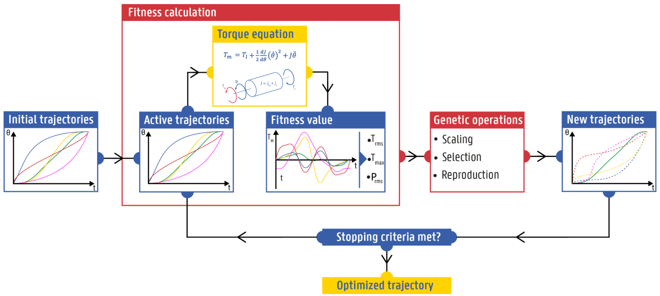

Fig. 5. Schematic overview of the working principles of applying genetic algorithms for trajectory optimization.

III. Trajectory Optimization with Genetic Algorithms

In this paper, a genetic algorithm (GA) is employed in order to optimize the design of the position trajectory between the defined and fixed position endpoints. A GA is a stochastic optimization searching algorithm first developed by Holland [21] that is based on evolution mechanisms such as mutation, crossover and selection in order to minimize a certain objective function. Starting from a randomly generated population of possible candidates, an iterative process is employed to evolve towards improved solutions [22], [23].

Applied to trajectory optimization and as depicted in Fig. 5, the routine will randomly generate a population of initial trajectories. The fitness value of these possible candidates is then determined and for this purpose, a description of the system established by the torque equation (1) is essential.

A. Position Profile

Every optimization routine uses certain design parameters which can be adjusted by the algorithm in order to converge towards an optimal solution. In this paper, these decision variables define the shape of the trajectory between start point θ_A at instant t_A and endpoint θ_B at instant t_B (Fig.1). As indicated in [2], optimality is often obtained through polynomial motion profile. Therefore, the trajectory or position profile itself is presented as a polynomial of order n with unknown coefficients $[p_0, p_1, \ldots, p_n]$ and $n+1$ DOF (8).

$$\theta(t) = p_0 + p_1 t^1 + p_2 t^2 + \ldots + p_n t^n \tag{8}$$

The lower order coefficients of this polynomial result from constraints of position θ, velocity $\dot{\theta}$ and acceleration $\ddot{\theta}$ in the start (θ_A) and endpoint (θ_B) (9). On the assumption that these constraints are imposed by the machine user, these six extra constraints result in $n-5$ degrees of freedom for the optimization algorithm. The higher order coefficients are free design parameters for the optimization algorithm [6]. A common trajectory profile in industry that can serve as a reference is a 1/3 motion profile which accelerates during 1/3 of the time, moves at a constant speed during 1/3 and decelerates at the last 1/3 of the motion time [2].

$$\begin{array}{lllll} \theta(t_A) = \theta_A & ; & \dot{\theta}(t_A) = 0 & ; & \ddot{\theta}(t_A) = 0 \\ \theta(t_B) = \theta_B & ; & \dot{\theta}(t_B) = 0 & ; & \ddot{\theta}(t_B) = 0 \end{array} \tag{9}$$

B. Objective Function and Constraints

Depending on the chosen optimization criterion, it is possible to have different objectives, declared in fitness functions. In this paper, three different objectives are compared and validated, being root-mean-square torque T_{rms}, maximum torque T_{max} and root-mean-square power P_{rms}. These objectives are theoretically defined as follows:

1) Root-Mean-Square Torque T_{rms}

$$T_{rms} = \sqrt{\frac{1}{T}\int_0^T {T_m}^2 \, \mathrm{d}t} \tag{10}$$

2) Maximum Torque T_{max}

$$T_{max} = max(abs(T_m)) \tag{11}$$

3) Root-Mean-Square Power P_{rms}

$$P_{rms} = \sqrt{\frac{1}{T}\int_0^T P^2 \, \mathrm{d}t} = \sqrt{\frac{1}{T}\int_0^T (T_m\dot{\theta})^2 \, \mathrm{d}t} \tag{12}$$

Next to the constraints in the start and end points of the trajectory, several other machine specific constraints remain which the GA can take into account. There can be limits on speed $\dot{\theta}$, acceleration $\ddot{\theta}$ and jerk $\dddot{\theta}$ depending on the type of application. Moreover, the maximum torque can be limited based on the motor selection and the rate of change in torque $max(\frac{\mathrm{d}T_m}{\mathrm{d}t})$ can be narrowed when the electrical dynamics are considered. Furthermore, it is possible to have other application dependent process constraints which will limit the possible trajectories. All these constraints are controlled by direct constraint handling that leaves the constraints as they are and adapts the GA to enforce them, in contradiction to constraints θ, $\dot{\theta}$, $\ddot{\theta}$, $\dddot{\theta}$ in the start and endpoint which are incorporated directly in the polynomial trajectory [24].

IV. Motion Controller Design

The savings potential of the trajectories achievable with trajectory optimization is only fulfilled when the motor is able to follow the optimized trajectory setpoint. Therefore, a good motion controller needs to be designed in order to keep the tracking error as low as possible. The system has a torque T^* or current input i^* which is determined by cascade PI speed and P position control loops. Both the measured position and speed determine the error between the desired and actual position/speed setpoint, which in turn serves as an input for the controllers (Fig. 6).

The basic controllers only react to occurring errors, meaning tracking errors are quintessential to this approach. Feedforward control, also depicted in Fig. 6, is a control loop expansion that anticipates load disturbances and drives the actuators before tracking errors occur. For speed feedforward, this means that the position setpoint gets derived into a speed set-point and injected into the speed controller input, bypassing the position controller. This enables a more robust controller since it relieves the feedback position (P) and speed (PI) controllers from their task to react dynamically and aggressively [25].

For applications with constant inertia, the speed setpoint can also be derived into the desired acceleration. By multiplying this acceleration with the inertia of the system, the required acceleration torque $J\ddot{\theta}$ can be determined. This results in an acceleration or torque feedforward (Fig. 6).

However, for systems with variable inertia, the desired driving torque is not limited to $J\ddot{\theta}$. Therefore, equation (1) should be considered. The implementation of (1) can be achieved in multiple ways, depending on what is achievable with the selected drive/software. Using CAD motion simulations, the required torque can be determined for a desired trajectory. These results can be exported as a lookup table and loaded into the drive. (Fig. 7 top). [26] Nonetheless, this approach is trajectory dependent and requires a new table for every new trajectory.

Another more advanced possibility is the implementation of the torque equation (1) in the controller scheme allowing to use the controller for different position profiles (Fig. 7). Upon this implementation, the aforementioned equation requires two different lookup tables; one that contains the inertia J with respect to the time and one that contains the variation of inertia $\frac{dJ}{d\theta}$. Instead, by using the chain rule (13) and combining it with equation (1), an alternate version of the torque equation (14) can be obtained. Since the time derivative of the inertia $\frac{dJ}{dt}$ is used in (14) instead of the position derivative in (1), this equation can be implemented in the controller with only one lookup table (Fig. 7 bottom).

$$\frac{1}{2}\frac{dJ}{d\theta}(\dot{\theta})^2 = \frac{1}{2}\frac{dJ}{dt}\frac{dt}{d\theta}(\dot{\theta})^2 = \frac{1}{2}\frac{dJ}{dt}\dot{\theta} \tag{13}$$

$$T_m = T_l + \frac{1}{2}\frac{dJ}{dt}\dot{\theta} + J\ddot{\theta} \tag{14}$$

For clarity of the scheme, the load torque T_l has been left out in Fig. 7, although in reality this torque is added to the output of the feedforward as a lookup table as well.

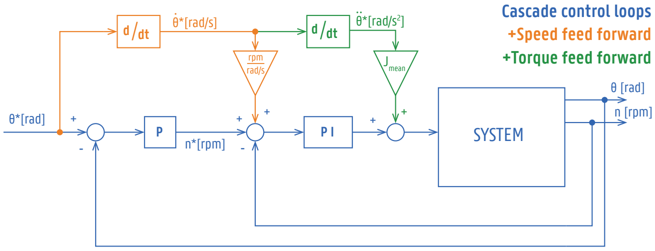

Fig. 6. Schematic overview of the feedforward motion control scheme with torque feedforward for systems with constant inertia.

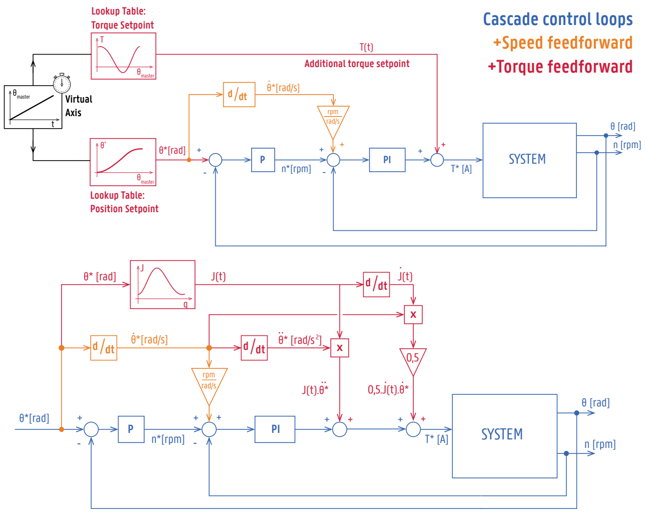

Fig. 7. Schematic overview of the motion controllers for systems with variable inertia. Depending on the driver capabilities, the torque feedforward can be implemented as lookup tables from CAD motion simulations (top) or as an implementation of the torque equation in the control scheme (bottom).

V. Results

In order to evaluate the effect of the aforementioned optimization procedure, the complete method is applied and validated on the experimental set-up (Fig. 2). The mechanism is required to cycle through a back-and-forth movement bounded by its start position θ_A of 0° and end position θ_B of 173.6° and has a cycle time of $0.3s$

Starting from the CAD model, the system dependent properties of the pick and place unit $J(\theta)$ and $T_l(\theta)$ are extracted. The results are passed on to the GA and the optimization routine is conducted for each of the three objectives (10)-(12). The trajectory is defined as a 17th order polynomial with 12 DOF (8). The results of the theoretical saving potential achieved with GA are presented in Table I.

TABLE I
Saving potential achieved with GA.

Trajectory	$T_{rms}[Nm]$	$T_{max}[Nm]$	$P_{rms}[W]$
1/3-profile (Reference)	15.09	63.62	919.2
Optimized to T_{rms}	12.06 *-20.1%*	37.26 *-41.4%*	619.0 *-32.7%*
Optimized to T_{max}	12.64 *-16.3%*	34.76 *-45.3%*	680.0 *-26.0%*
Optimized to P_{rms}	12.27 *-18.7%*	37.76 *-40.6%*	640.9 *-30.3%*

Theoretically, a maximum torque T_{max} reduction of 45.3% is possible and the root-mean-square torque T_{rms} can be diminished with 20.1%. It is interesting to note that the largest root-mean-square power P_{rms} savings (-32.7%) occur when the trajectory is optimized to the root-mean-square torque T_{rms}. This is a consequence of the functioning of GA's that can get stuck in local minima and cannot guarantee to find the global optimum. Nevertheless, a correlation exists between the root-mean-square torque T_{rms} and power P_{rms} as optimizing one will almost always also affect the other similarly.

In order to validate these savings on the experimental set-up, an adequate controller must be selected. Therefore, in Fig. 8, the previously described motion controllers are applied on the physical set-up and the performance is compared. The trajectory with maximum torque T_{max} as the optimization criterion is selected as set-point for the controllers as this limits the required torque and thus allows for a smaller motor.

The cascade position (P) and speed (PI) controller are configured as robust controllers in order to increase the stability of the system. However, as depicted in Fig. 8, solely employing these controllers leads to undesired behavior as the controller is not able to follow the desired trajectory and the constraints of the process are not satisfied. However, a speed feedforward can significantly increase the performance of the controller, with a maximum tracking error of 8.91°, Nevertheless, the best results are achieved when combining the speed and torque feedforward, especially on high dynamical applications, as illustrated in Fig. 7 with a maximum tracking error of 6.93°. When using feedforward, it is no longer necessary to rely solely on the feedback loops for high dynamics. Instead, the feedback controllers can be set more robust while maintaining an accurate execution of the trajectory.

Finally, the optimized trajectory is validated on the experimental set-up. Therefore, the trajectories are loaded into the motor drive and a motion controller with torque feedforward is implemented as in Fig. 7. Table II reports the theoretical and measured savings concerning a standard 1/3-profile and

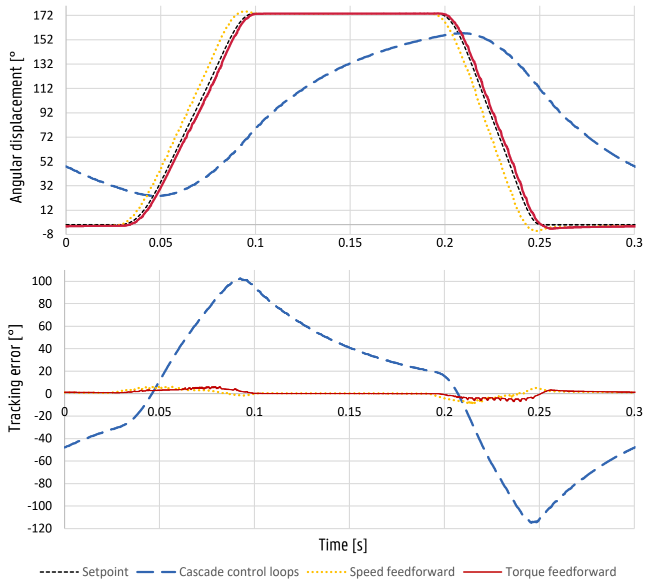

Fig. 8. Results of angular displacement and tracking error of the pick and place unit with different feedforward loops.

the trajectory optimized to maximum torque T_{max}. All measurements are averaged over 10 cycles. The theoretical values are adopted from Table I.

TABLE II

ACHIEVED SAVINGS ON THE EXPERIMENTAL SET-UP.

Trajectory		T_{rms} **[Nm]**	T_{max} **[Nm]**	P_{rms} **[W]**
1/3-profile	Theoretical	15.09	63.62	919.2
	Experimental	13.44	42.67	838.5
Optimized to T_{max}	Theoretical	12.64 -16.3%	34.76 -45.3%	680.0 -26.0%
	Experimental	11.47 -14.7%	37.3 -12.6%	646.5 -22.9%

VI. CONCLUSIONS

For what concerns the experimental results, large savings can be obtained when optimizing the design of the trajectory. It can be noted that the achieved savings are smaller than the theoretically obtained values, especially for what concerns the maximum torque T_{max} where a reduction of -12.6% is realized as opposed to the theoretical value of -45.3%. This is a consequence of the employed one-mass model which does not take the flexibility of the coupling into account. Nevertheless, large savings can be obtained with the described optimization procedure (T_{rms} -14.7%, T_{max} -12.6% and P_{rms} -22.9%) while reducing the required torque and therefore extending the life span of the machine.

REFERENCES

[1] D. A. Dornfeld, *Green manufacturing: fundamentals and applications*. Springer Science & Business Media, 2012.

[2] 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.

[3] M. Boryga and A. Graboś, "Planning of manipulator motion trajectory with higher-degree polynomials use," *Mechanism and Machine Theory*, vol. 44, no. 7, pp. 1400–1419, 2009.

[4] 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.

[5] 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.

[6] 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.

[7] 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.

[8] M. S. Huang, K. Y. Chen, and R. F. Fung, "Numerical and experimental identifications of a motor-toggle mechanism," *Applied Mathematical Modelling*, vol. 33, no. 5, pp. 2502–2517, 2009.

[9] K. Tang, K. Man, and S. Kwong, "Genetic algorithms and their applications," *IEEE SIGNAL PROCESSING MAGAZINE*, pp. 22–37, 1996.

[10] G. Wenzhong and S. K. Porandla, "Design optimization of a parallel hybrid electric powertrain," *2005 IEEE Vehicle Power and Propulsion Conference, VPPC*, vol. 2005, pp. 530–535, 2005.

[11] K. O. Stanley and R. Miikkulainen, "Evolving neural networks through augmenting topologies %J Evolutionary Computation," *Evolutionary Computation*, vol. 10, no. 2, pp. 99–127, 2002.

[12] L. Tian and C. Collins, "An effective robot trajectory planning method using a genetic algorithm," *Mechatronics*, vol. 14, no. 5, pp. 455–470, 2004.

[13] S. Števo, I. Sekaj, and M. Dekan, *Optimization of robotic arm trajectory using genetic algorithm*. IFAC, 2014, vol. 19, no. 3.

[14] V. Bende, P. M. Pathak, K. S. Dixit, and S. P. Harsha, "Energy optimal trajectory planning of an underwater robot using a genetic algorithm," *Proc. of the Institution of Mech. Engineers. Part I: Journal of Systems and Control Engineering*, vol. 226, no. 8, pp. 1077–1087, 2012.

[15] Y. L. Hsu, M. S. Huang, and R. F. Fung, "Convergent Analysis of an Energy-Saving Trajectory for a Motor-Toggle System," *Journal of Vibration Engineering & Technologies*, vol. 3, no. 1, pp. 95–112, 2015.

[16] 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.

[17] D. D. Roover and O. H. Bosgra, "Synthesis of robust multivariable iterative learning controllers with application to a wafer stage motion system," *International Journal of Control*, vol. 73, no. 10, pp. 968–979, 2000.

[18] S. Gunnarsson and M. Norrlof, "On the design of ILC algorithms using optimization," *Automatica*, vol. 37, no. 12, pp. 2011–2016, 12 2001.

[19] H. Dresig and F. Holzweißig, *Dynamics of Machinery: Theory and Applications*. Springer Berlin Heidelberg, 2010.

[20] K. Sollmann, M. Jouaneh, and D. Lavender, "Dynamic Modeling of a Two-Axis Parallel H-Frame-Type XY Positioning System," *IEEE/ASME Transactions on Mechatronics*, vol. 15, no. 2, pp. 280–290, 2010.

[21] J. H. Holland, *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence*. Cambridge, MA, USA: MIT Press, 1992.

[22] U. Bodenhofer, "Genetic Algorithms: Theory and Applications," Linz-Hagenberg, 2004.

[23] M. Mitchell, *An Introduction to Genetic Algorithms*. London: The MIT Press, 1998.

[24] B. Craenen, A. Eiben, and E. Marchiori, "How to Handle Constraints with Evolutionary Algorithms," *The Practical Handbook of Genetic Algorithms*, no. September 2012, 2000.

[25] C. G. Atkeson, J. D. Griffiths, and J. M. Hollerbach, "Experimental Evaluation of Feedforward and Computed Torque Control," *IEEE Transactions on Robotics and Automation*, vol. 5, no. 3, pp. 368–373, 1989.

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