Multibody System Dynamics
https://doi.org/10.1007/s11044-025-10127-4

RESEARCH

Model predictive pose optimization for energy efficient robotic machining

Philip Carstensen[1] · Robert Seifried[2] · Christian Möller[1] · Christian Böhlmann[1]

Received: 31 March 2025 / Accepted: 29 October 2025

Abstract
One focus of aviation research in recent years has been the reduction of greenhouse gases through weight savings in aircraft construction, e.g. by using lightweight materials such as CFRP components. Lighter aircraft lead to less fuel consumption, which impacts the global carbon footprint and is an important step towards CO_2-neutral flying. In order to achieve national climate goals, the environmental impact should already be assessed at the production stage. Most recently, hybrid kinematics consisting of serial kinematics (industrial robots on linear rails) and parallel kinematics (hexapod) mounted to the robot's end effector, have been studied at Fraunhofer IFAM for the purpose of machining distorted CFRP-components. This work analyzes to which extent the redundant degrees of freedom of such hybrid kinematics can be exploited to determine energetically advantageous poses during machining. Therefore, the inverse kinematics are solved within an Optimal Control Problem (OCP) including the forward kinematics as well as minimizing the system's energy consumption while following a reference trajectory.

Keywords Model predictive control · Pose optimization · Energy efficiency · Robotic machining

1 Introduction

Advanced lightweight materials such as Carbon-Fiber-Reinforced-Plastic (CFRP) components contribute significantly to a lower aircraft fuel consumption. This impacts the global carbon footprint and is an important step towards CO_2-neutral flying. With an increasing demand for sustainability and energy efficiency in order to achieve national climate goals and to tackle increased energy prices, the environmental impact should already be assessed at the production stage. Typically, large gantry machines are used for CNC machining in

✉ P. Carstensen
philip.carstensen@ifam.fraunhofer.de

R. Seifried
robert.seifried@tuhh.de

1 Institute for Manufacturing Technology and Advanced Materials, Fraunhofer Institute IFAM, Ottenbecker Damm 12, 21684 Stade, Germany

2 Institute for Mechanics and Ocean Engineering, Hamburg University of Technology, Eißendorfer Str. 42, 21073 Hamburg, Germany

Published online: 12 November 2025

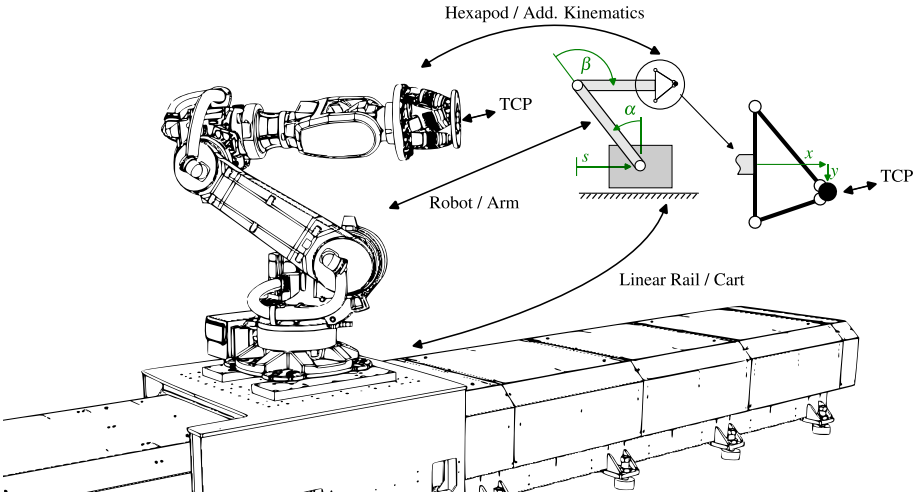

Fig. 1 Hybrid kinematics robot at Fraunhofer Institute IFAM of a linear rail (Güdel), serial kinematics industrial robot (Mabi Max 100) and parallel kinematics hexapod (PI Physics Instruments). The system is reduced to a simplified 2D multibody system

the aviation industry. These custom-built structures are highly expensive and can take on the dimensions of entire buildings in terms of weight and volume. Only recently, the use of industrial robots for CNC machining has been studied, see for example [1, 2] and [3]. Compared to conventional CNC machines, industrial robots offer a better ratio of size to working space and are significantly more cost effective. These advantages are contrasted by the lower precision and higher compliance of industrial robots.

To compensate for these disadvantages, Fraunhofer IFAM has been working on increasing the path accuracy of industrial robots through the extension of external sensors, new robot structure concepts designed for machining, and hybrid drive systems [4, 5]. Most recently, a hybrid kinematics concept consisting of serial kinematics (industrial robot on a linear rail) and parallel kinematics (hexapod) mounted at the robots end effector, as shown in Fig. 1, were studied. A combination of serial and parallel kinematics enables the fusion of the respective attributes of low stiffness with a large work space versus high stiffness as well as precision with a small work space of the respective systems. The additional kinematics (hexapod) are used to perform a sensor guided path adaption to track features along the machined component and thus react to and compensate for component deformation [6].

In terms of energy consumption, robotic systems with extended workspace as presented in Fig. 1, promise an advantage over conventional large scale CNC machines due to their size alone, as significantly less mass is moved. In this work however, a more detailed analysis is performed to determine the extent to which the redundant degrees of freedom of such hybrid kinematics can be exploited to determine energetically advantageous robot poses during machining.

The energy consumption of industrial robots is analyzed in [7] and multiple approaches exist to reduce the energy consumption of industrial robots. This includes mechanically retrofitting the robot with energy storing elements such as compliance mechanism or regenerative drives, in which energy induced by braking can be reused instead of being converted into heat, see [8] or [9].

In [10] the energy-efficient trajectory planning in operational space of an industrial *ABB* robot (*IRB1600*) is examined. Here, a pick and place task is considered without boundary conditions on the trajectory followed while moving from point A to point B. As one result utilizing the breaks while in standstill operation is emphasized. In machining however, the reference trajectory, i.e. the tool path, is given and path accuracy is crucial. Similarly, in [11] it is considered that in many robotic applications the task space is over determined, e.g., the orientation around the tool axis, which is also the case in the milling application. The additional degrees of freedom are used to derive a trajectory for the robot that is as energy-efficient as possible. A detailed overview on energy savings in robotics is found in [12]. Here, methods are separated in a hardware heavy approach as well as software like path planning and trajectory optimization.

While the former mentioned methods have an application in the field of pick and place, retrofitting or optimizations in the workspace in common, this work is focused on minimizing the energy consumption by finding energetically advantageous joint space poses during machining. Therefore, a hybrid kinematics model is derived and a standard algorithm for the calculation of the inverse kinematics is presented. The standard algorithm is then extended in order to utilize the system's redundancy for finding energetically advantageous poses. Afterwards, the problem is formulated as an optimal control problem (OCP) including the full system dynamics. The OCP is solved in a model predictive manner enabling the system to react to trajectory changes in advance.

The purpose of this work is to show the potential for energy savings based on simulations when the energy consumed by the system is already taken into account within the inverse kinematics calculation. In particular, the effect of adding additional kinematics is considered.

2 Dynamic robot model

In this section, the kinematic and dynamic model of the robot is derived, as both are needed in this work. A robot is represented as a kinematic chain of rigid bodies and joints. Each joint motion introduces a degree of freedom (dof) to the system and contributes to the overall movement. Thus, the kinematics of the robot can be described in generalized joint coordinates $\mathbf{q} \in \mathbb{R}^f$ with f degrees of freedom. By applying the Newton-Euler formalism for multibody systems, see e.g. [13], the robot's equation of motion (eqm)

$$\mathbf{M}(\mathbf{q})\,\ddot{\mathbf{q}} + \mathbf{k}(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{g}(\mathbf{q}) + \boldsymbol{\tau}, \tag{2.1}$$

is obtained, with inertia matrix $\mathbf{M}(\mathbf{q})$, the vector of generalized coriolis and centrifugal forces $\mathbf{k}(\mathbf{q}, \dot{\mathbf{q}})$, the generalized gravitational force $\mathbf{g}(\mathbf{q})$ and the drive torques $\boldsymbol{\tau}$. This can also be viewed in the form of a nonlinear control system

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}) \tag{2.2}$$

$$\mathbf{y} = \mathbf{h}(\mathbf{x}) \tag{2.3}$$

with system state $\mathbf{x} = \begin{bmatrix}\mathbf{q}, & \dot{\mathbf{q}}\end{bmatrix}^T$, the control input $\mathbf{u} \in \mathbb{R}^m$, e.g. the joint torques, the rearranged dynamics $\mathbf{f} : \mathbb{R}^{2f} \times \mathbb{R}^m \to \mathbb{R}^{2f}$ from equation (2.1) and $\mathbf{h} : \mathbb{R}^{2f} \to \mathbb{R}^n$ mapping the system state to the output $\mathbf{y} \in \mathbb{R}^n$.

In this work, a machining application is considered as a control task for the robot, i.e. trajectory tracking of the robot's tool center point (TCP). Thereby, equation (2.3) describes

precisely the forward kinematics of the robot and is only dependent of joint coordinates $\mathbf{q}$. Time differentiation yields a linear dependence

$$\dot{\mathbf{y}} = \mathbf{J}(\mathbf{q})\,\dot{\mathbf{q}} \tag{2.4}$$

between the system's output velocity $\dot{\mathbf{y}}$ and joint velocity $\dot{\mathbf{q}}$, with Jacobian matrix $\mathbf{J}(\mathbf{q}) \in \mathbb{R}^{n\times f}$ describing their kinematic relationship. The aim of this work is to find joint space poses $\mathbf{q}$ that are energetically beneficial while following a reference trajectory in task space $\mathbf{y}$. Since the Jacobian matrix $\mathbf{J}(\mathbf{q})$ is not square, it is not possible to solve equation (2.4) for corresponding joint velocities $\dot{\mathbf{q}}$ directly. In the case of $n < f$ a system is redundant, i.e. there exist more degrees of freedom than system outputs and internal joint motion is possible.

In order to analyze the extent to which the redundancy of a system can be exploited to obtain more energetically favorable poses when following a trajectory, the system shown in Fig. 1 is examined. For a simplified conceptual analysis the system is reduced to a 2D multibody system either with or without additional kinematics at the TCP. Redundancy holds for both, the system with $\mathbf{q} = \left[s, \alpha, \beta, x, y\right]^T$ and without additional kinematics $\mathbf{q} = \left[s, \alpha, \beta\right]^T$.

3 Trajectory design

For the purpose of this work, a trajectory is required that contains elements typical for machining. In order to ensure comparability, a test path is chosen as reference and depicted in Fig. 2. This path is proposed in the DIN ISO 9283 Norm [14] and commonly used to analyze the precision and path accuracy of industrial robots [4, 5, 15].

Here, it serves as a reference for machining applications, as it contains segments with straight lines, curves, sharp edges and corners. When traversing sharp turns, high accelerations or braking is necessary to follow the contour. This is particularly reflected in the drive torques required for such abrupt changes in direction.

In general, a distinction between path and trajectory is made. A path $\mathbf{p}(s) \in \mathrm{R}^p$ describes a sequence of points in space defined as a function of the variable s, e.g. the arc length, and it does not contain any temporal information. Whereas a trajectory describes the temporal sequence of points and is obtained by defining a timing law $s(t)$ for the path $\mathbf{p}(s(t))$ to follow.

In machining, the robot's end effector (TCP) position should follow the reference trajectory $\mathbf{p}(s(t))$, therefore

$$\mathbf{p}(s(t)) = \mathbf{h}(\mathbf{x}(t)) \tag{3.1}$$

must hold for the system output in equation (2.3) at all times t. Here, the dependency on $\mathbf{x} = \left[\mathbf{q}, \dot{\mathbf{q}}\right]^T$ serves the purpose of generalization from equation (2.3), as the end effector's position can be expressed solely in terms of the joint angles $\mathbf{q}$.

Velocity $\dot{\mathbf{p}}$ and acceleration $\ddot{\mathbf{p}}$ along the path depend as follows

$$\dot{\mathbf{p}}(t) = \frac{d\mathbf{p}(s)}{dt} = \frac{\partial \mathbf{p}(s)}{\partial s}\frac{ds}{dt}, \tag{3.2}$$

$$\ddot{\mathbf{p}}(t) = \frac{d\mathbf{p}_{\mathbf{v}}(s)}{dt} = \frac{\partial^2 \mathbf{p}(s)}{\partial s^2}\left(\frac{ds}{dt}\right)^2 + \frac{\partial \mathbf{p}(s)}{\partial s}\frac{d^2 s}{dt^2} \tag{3.3}$$

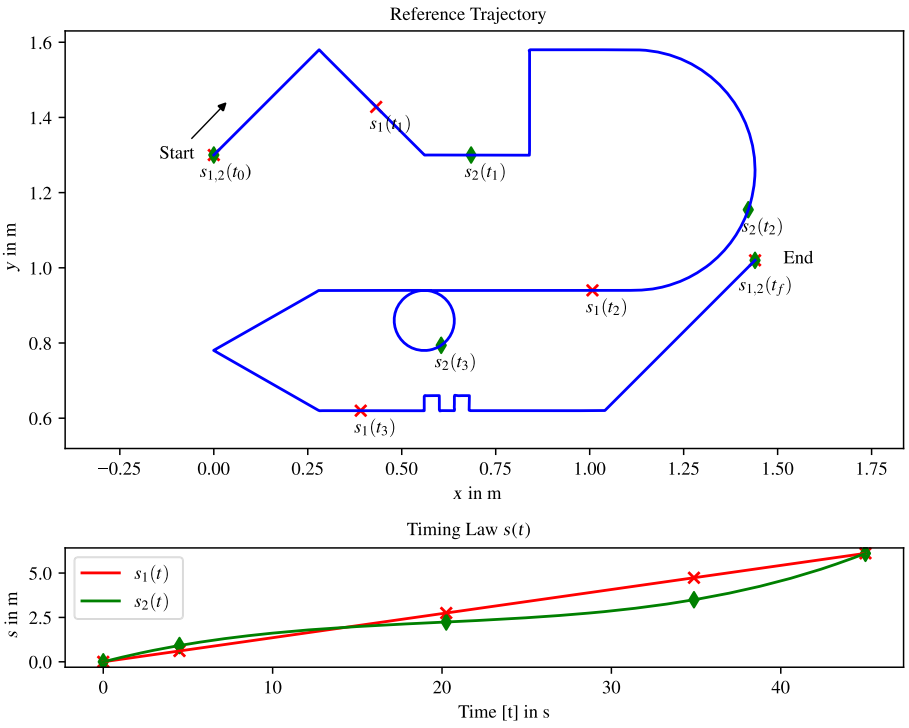

Fig. 2 Test path for a typical machining application. The influence by a different choice of timing law $s(t)$ is visualized by the different progress when following the reference path

on the choice of timing law $s(t)$. The timing law should therefore be selected as a continuous function, which is at least two times differentiable, in order to ensure a continuous velocity and acceleration along the path and thus feasible trajectories.

In order to minimize the energy required during machining, it is interesting to choose a suitable timing law, e.g. induce braking before sharp turns. For visualization, Fig. 2 also shows two different choices for the timing law $s(t)$, with $s_1(t)$ being linear in time and $s_2(t)$ being an arbitrary continuous function. To show the effect of choosing a different timing law $s(t)$, the traveled distance along the path is depicted for both cases in Fig. 2.

4 Inverse kinematics algorithms

In this section, standard inverse kinematics (IK) algorithms are presented in order to derive poses $\mathbf{q}_i \in \mathbb{R}^f$ in joint space coordinates from points $\mathbf{p}_i \in \mathbb{R}^p$ in cartesian coordinates that fulfill equation (3.1) at each time step t_i. The presented methods serve as a reference and are applied to the system without additional kinematics. Due to redundancy and thus the existence of multiple solutions, the inverse kinematics calculation is often formulated as a

constrained optimization problem

$$\underset{\dot{\mathbf{q}}}{\operatorname{argmin}} \quad c(\dot{\mathbf{q}}) = \frac{1}{2}\dot{\mathbf{q}}^T \mathbf{W}\dot{\mathbf{q}} \tag{4.1}$$

$$\text{s.t.} \quad \dot{\mathbf{p}} = \mathbf{J}(\mathbf{q})\,\dot{\mathbf{q}}, \tag{4.2}$$

as in [16]. In this optimization problem the cost function $c(\dot{\mathbf{q}})$ emphasizes to find the minimal joint velocities $\dot{\mathbf{q}}$ weighted by positive definite matrix $\mathbf{W}$, that also satisfy constraint equations (4.2). Here, the joint velocities are mapped onto the end effector velocity using the kinematic relationship from equation (2.4) and are then set to follow the path velocity $\dot{\mathbf{p}}$. Equation (4.1) is a linear optimization problem with quadratic cost, also referred to as quadratic program (QP), hence an analytical solution

$$\dot{\mathbf{q}} = \underbrace{\mathbf{W}^{-1}\mathbf{J}^T(\mathbf{q}_k)\left[\mathbf{J}(\mathbf{q}_k)\,\mathbf{W}^{-1}\mathbf{J}^T(\mathbf{q}_k)\right]^{-1}}_{\mathbf{J}_W^{\dagger}}\dot{\mathbf{p}} \tag{4.3}$$

can be derived yielding the weighted Moore-Penrose pseudo inverse $\mathbf{J}_{\mathbf{W}}^{\dagger}$ with weighting matrix $\mathbf{W}$ that is often chosen to be the identity matrix $\mathbf{I}_f$. In the sense of inverse kinematics, large weights $w_{i,i}$ result in a reduced movement of the corresponding degree of freedom q_i.

In order to obtain the joint position $\mathbf{q}$ from the solution of equation (4.3), a quasi-Euler integration

$$\mathbf{q}_{k+1} = \mathbf{q}_k + \alpha_0\,\mathbf{J}_W^{\dagger}\mathbf{e}_k, \tag{4.4}$$

is performed as an update for the current solution $\mathbf{q}_k$ at time t_i, with $\alpha_0 \in (0, 1]$ not representing a time interval but a selectable step size. Here, $\dot{\mathbf{p}}$ is substituted by the residual error

$$\mathbf{e}_k = \mathbf{p}_i - \mathbf{h}(\mathbf{q}_k) \tag{4.5}$$

of cartesian reference point $\mathbf{p}_i$ at time t_i and forward kinematics $\mathbf{h}(\mathbf{q}_k)$ as in equation (2.3) with respect to the robot's end effector. The update is performed iteratively $k = 1, \ldots, k_{\max}$ and for each time step t_i until the residual error converges to a threshold or a maximum number of iterations $k_{\max}$ is reached. In order to obtain a continuous solution and fast convergence, the solution from the last time step t_{i-1} is used as the initial value for the current time step t_i.

Figure 3 shows the robot system from Sect. 2 without additional kinematics following the path from Sect. 3. The corresponding poses are calculated according to equation (4.4) and this solution to the inverse kinematics problem is denoted as 'A'. In addition, Fig. 3 shows another solution 'B' to the inverse kinematics problem, visualizing the possibilities due to the redundancy of the system.

Even without additional kinematics, $n < f$ holds and the robot's extra degree of freedom can be utilized to perform subtasks. In [17] for example, it is motivated to use redundancy for finding a maximum stiff positioning of a machining robot. Similarly, a method to obtain the energetically beneficial poses of solution 'B' is presented in the following.

According to [16] internal joint motion of a robot can be used to perform subtasks by extending equation (4.3)

$$\dot{\mathbf{q}} = \mathbf{J}_{\mathbf{W}}^{\dagger}\dot{\mathbf{p}} + \underbrace{\left(\mathbf{I}_{n_q} - \mathbf{J}_{\mathbf{W}}^{\dagger}\mathbf{J}\mathbf{W}\right)}_{\mathbf{P}}\dot{\mathbf{q}}_0 \tag{4.6}$$

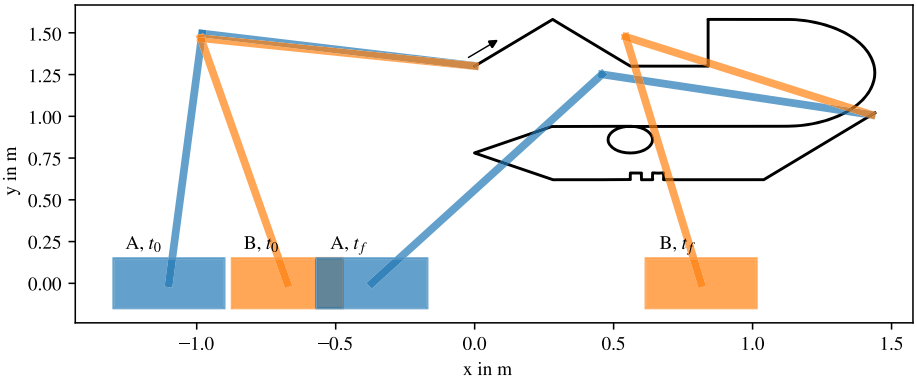

Fig. 3 The robot system without additional kinematics following the same reference trajectory. System 'A' for a normal inverse kinematics solution and system 'B' with energetically beneficial poses

with projection matrix $\mathbf{P}$. Hereby $\mathbf{P}$ projects any arbitrary joint velocity $\dot{\mathbf{q}}_0$ onto the null space of $\mathbf{J}$, hence allowing for additional joint motion $\dot{\mathbf{q}}_0$ that is not affecting the end effector's position. Generally, the internal joint motion $\dot{\mathbf{q}}_0$ can be represented as

$$\dot{\mathbf{q}}_0 = k_0 \frac{\partial w(\mathbf{q})}{\partial \mathbf{q}}, \tag{4.7}$$

the gradient of a newly introduced cost function $w(\mathbf{q})$ that is minimized along the original cost $c(\dot{\mathbf{q}})$ in equation (4.1). There are various options for a suitable choice of $w(\mathbf{q})$, to ensure maximum maneuverability, avoid singularities or maintain joint limits, for instance. In this matter, the goal is to reduce the system's energy consumption, which goes along with minimizing the drives' power output.

Following [18], the voltage $\mathbf{v}$ and current $\mathbf{i}$ for electrical drives are given by

$$\mathbf{v} = \mathbf{R}\mathbf{i} + \mathbf{K}_{emf}\dot{\mathbf{q}}, \tag{4.8}$$

$$\mathbf{i} = \mathbf{K}_t^{-1}\boldsymbol{\tau}, \tag{4.9}$$

with resistance $\mathbf{R}$, back emf constant $\mathbf{K}_{emf}$, and drive torque constant $\mathbf{K}_t$. Equations (4.8) and (4.9) correspond to the steady state modeling of the drive, a common simplification in which, for example, the effect of armature inductance $\mathbf{v_a} = \mathbf{L_a}\frac{d\mathbf{i}}{dt}$ is neglected.

For the electrical power calculation P_w, this results in

$$P_w = \mathbf{i}^T\mathbf{v} \tag{4.10}$$

$$= \boldsymbol{\tau}^T\mathbf{K}_t^{-T}\mathbf{R}\mathbf{K}_t^{-1}\boldsymbol{\tau} + \boldsymbol{\tau}^T\mathbf{K}_t^{-T}\mathbf{K}_{emf}\dot{\mathbf{q}} \tag{4.11}$$

$$= \boldsymbol{\tau}^T\mathbf{K}_a\boldsymbol{\tau} + \boldsymbol{\tau}^T\mathbf{K}_b\dot{\mathbf{q}} \tag{4.12}$$

with drive specific constants $\mathbf{K}_a = \mathbf{K}_t^{-T}\mathbf{R}\mathbf{K}_t^{-1}$ and $\mathbf{K}_b = \mathbf{K}_t^{-T}\mathbf{K}_{emf}$ when no gear reduction is present. For most conventional drive systems the back emf constant $\mathbf{K}_{emf}$ and torque constant $\mathbf{K}_t$ are the same, when using SI-units, thus leading to $\mathbf{K}_b = \mathbf{I}$ being the identity matrix.

However, the drives' power P_w cannot be used as a cost function $w(\mathbf{q})$ for the inverse kinematics calculation, since it is not exclusively a function of joint variables $\mathbf{q}$ and a gradient may not be available. Therefore, a different cost function resembling the drives' power

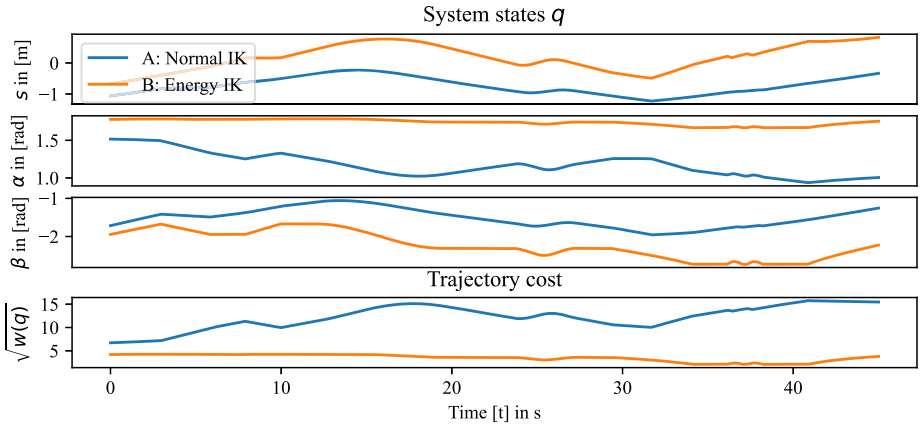

Fig. 4 The system states for the entire trajectory for both the solution to the normal inverse kinematics problem (A) as well as for the inverse kinematics problem utilizing redundancy (B). The trajectory cost respectively the gravitational load is shown for each time step t_i

P_w has to be considered. Equation (4.12) illustrates that both low drive torques $\boldsymbol{\tau}$ as well as drive velocities $\dot{\mathbf{q}}$ result in power reduction. In order to account for low drive torques

$$w\left(\mathbf{q}\right) = \mathbf{g}\left(\mathbf{q}\right)^T \mathbf{g}\left(\mathbf{q}\right) \tag{4.13}$$

is chosen as additional cost function $w(\mathbf{q})$. Here, $\mathbf{g}\left(\mathbf{q}\right)$ describes the gravitational load acting on the drives, see equation (2.1), which accounts for the main part and in the static case ($\ddot{\mathbf{q}} = \dot{\mathbf{q}} = 0$) for the entire part of the generated torques $\boldsymbol{\tau}$. In order to also account for the joint velocity component in equation (4.12), the weighting matrix $\mathbf{W}$ in equation (4.3) is substituted for the inertia matrix $\mathbf{M}\left(\mathbf{q}\right)$ from equation (2.1). This results in the kinetic energy

$$T = \frac{1}{2}\dot{\mathbf{q}}^T \mathbf{M}(\mathbf{q})\dot{\mathbf{q}} \tag{4.14}$$

as expression for the original cost function $c(\dot{\mathbf{q}})$, which favours the movement of low inertia elements accordingly. Together with equations (4.6) and (4.7) this yields

$$\mathbf{q}_{k+1} = \mathbf{q}_k + \alpha_0 \, \mathbf{J}^{\dagger}_{\mathbf{M}(\mathbf{q}_k)} \mathbf{e}_k + k_0 \mathbf{P} \frac{\partial w\left(\mathbf{q}\right)}{\partial \mathbf{q}} \tag{4.15}$$

as iterative update for energetically beneficial poses $\mathbf{q}_k$. Hereby k_0 is a tunable parameter that adjusts the extent to which the additional cost $w(\mathbf{q})$ is considered. Both, the solutions for 'A' the normal inverse kinematics (IK) calculated according to equation (4.4) and 'B' for the inverse kinematics utilizing redundancy according to equation (4.15), denoted as 'Energy IK' from now on, are shown in Fig. 4. They represent the systems also shown in Fig. 3, but for the entire trajectory. A difference for the cart position s and arm angles α and β is clearly visible. In addition, the trajectory cost $w(\mathbf{q})$ according to (4.13), i.e. the resulting gravitational load acting on the joints, is significantly reduced for case 'B'. This is achieved by reducing the lever of the center of gravity, as it can be observed in Fig. 3.

Despite the promising outcome, this method has proven to be numerically challenging, demanding for a tuning effort of parameters α_0 as well as k_0 and thus resulting in normalizing the gradient in equation (4.7) or very small k_0. Eventually, the goal is to incorporate

additional kinematics into the cart-pole system. In particular, an additional point mass in reference to the hexapod depicted in Fig. 1, that can move in x- and y-direction of the reference path. The additional kinematics add to the system's redundancy, but are limited in movement due to its workspace, though. Thus, advanced methods are needed for the inverse kinematics calculation of such systems that might also account for joint limits.

5 Model predictive pose optimization

The inverse kinematics calculation was motivated as the solution of an optimization problem in the previous section. In order to find an energetically advantageous pose, a cost function containing the kinetic energy as well as the gravitational load acting on the drives was minimized. The former is solved for each time step independently, thus corresponding to a very local optimum with respect to the overall trajectory.

At this point, the same task is formulated as optimal control problem (OCP). OCPs are used for path planning or trajectory optimization in various fields, see for example [19]. Similarly, in [20], an optimal control problem is used as a path planner for a robot and Bayesian optimization is applied to learn the hyper parameters of the OCP's cost function with a focus on minimum energy consumption. The results are trajectories deviating from the reference path by rounding or smoothing corners in particular. In machining, however, path accuracy is essential. Thus, one goal of this work is to analyze whether adding additional kinematics can be beneficial to the energy efficiency.

The compensating movements of the additional kinematics allow the robot to deviate from the reference path as long as this is still within the working space of the additional kinematics. This enables the robot to round corners and hence, less abrupt changes in direction for the high inertia part of the system are needed. Moreover, while the inverse kinematics algorithms in Sect. 4 only calculate the pose at each time step, a model predictive approach is used in this section with the expectation that including further time steps in the IK calculation enables an early response to corners or major reorientation while following the trajectory. Such a solution to an OCP over a finite time horizon corresponds to a more global solution to the overall problem.

In general a OCP is defined as constrained nonlinear optimization problem (NLP)

$$\underset{\mathbf{x}(t),\mathbf{u}(t)}{\operatorname{argmin}} \quad c(\mathbf{x}(t_f)) + \int_{t_0}^{t_f} c\left(\mathbf{x}\left(t\right), \mathbf{u}\left(t\right)\right) dt \tag{5.1}$$

$$\text{s.t.} \quad \dot{\mathbf{x}}\left(t\right) = \mathbf{f}\left(\mathbf{x}\left(t\right), \mathbf{u}\left(t\right)\right) \tag{5.2}$$

$$\mathbf{p}(t) = \mathbf{h}\left(\mathbf{x}(t)\right) \tag{5.3}$$

$$\mathbf{x}\left(t\right) \in \mathcal{X}, \mathbf{u}\left(t\right) \in \mathcal{U} \tag{5.4}$$

in order to reduce a cost function $c\left(\mathbf{x}, \mathbf{u}\right)$ of system states $\mathbf{x} = \left[\mathbf{q}, \dot{\mathbf{q}}\right]^T$ and inputs $\mathbf{u} = \boldsymbol{\tau}$ for time interval $t \in \left[t_0, t_f\right]$. The goal for a given task is formulated within the cost function, and is reached by minimizing it. In order to guarantee physicality, the system's dynamics are included in constraint equations (5.2). Moreover, limitations in the system's states or inputs are also maintained by equations (5.4). To ensure path accuracy at all cost, the forward kinematics and thus the system's output from equation (2.3) is represented in the OCP formulation by equation (5.3), where $\mathbf{p}(t)$ corresponds to $\mathbf{y}(t)$ of equation (2.3).

The combination of ordinary differential equations (5.2) (ODE) and algebraic equations (5.3) within the constraints form a system of differential algebraic equations (DAE),

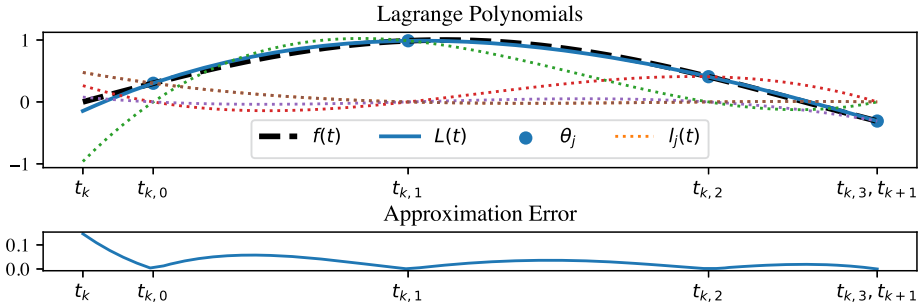

Fig. 5 Example for the approximation of function $f(t)$ by Lagrange Polynomial $L(t)$ and marked collocation points at $t_{k,j}$

that are typically hard to solve numerically. Therefore, a collocation method is used to solve the OCP in equation (5.1), see e.g. [21] or [22]. Collocation is based on polynomials or other basis functions that piecewise approximate the solution of an ODE or DAE on a section $x_k(t) : x \in (x_k, x_{k+1}]$ at discretized time intervals $t_k = kh$, $k = (0, \ldots, N_h)$ with step size h over a horizon N_h. One possible choice for such basis functions are Lagrange Polynomials

$$L(t) = \sum_{j=0}^{n_c} \theta_j l_j(t) \tag{5.5}$$

$$l_j(t) = \prod_{\substack{0 \le m \le n_c \\ m \neq j}} \frac{t - t_m}{t_j - t_m} \tag{5.6}$$

of order n_c defined by θ_j at collocation points $t_{k,j} \in [t_k, t_{k+1}]$. Figure 5 shows an arbitrary function $f(t)$ represented by Lagrange polynomials $L(\theta, t)$. Also depicted in Fig. 5 is the approximation error. That is, the distance between the actual function $f(t)$ and the approximation $L(\theta, t)$. By definition, the approximation error at the collocation points $t_{k,j}$ is zero, i.e. $L(\theta, t_{k,j}) = f(t_{k,j})$, and scales with time step h in the interpolated intervals.

Furthermore, this also applies to the piecewise approximation of $\mathbf{x}_k(t) = \begin{bmatrix} \mathbf{q}_k(t) & \dot{\mathbf{q}}_k(t) \end{bmatrix}^T$ by Lagrange Polynomials, and the property

$$\mathbf{x}_k(t = t_{k,j}) = \mathbf{L}(\boldsymbol{\theta}_k, t_{k,j}) = \boldsymbol{\theta}_{k,j} \tag{5.7}$$

with $\boldsymbol{\theta}_k = \begin{bmatrix} \theta_{k,0}, \ldots, \theta_{k,n_c} \end{bmatrix}$ holds. Thus, reducing the problem in equation (5.1) to a discretized optimization

$$\underset{\boldsymbol{\theta}, \boldsymbol{\tau}_0, \ldots, \boldsymbol{\tau}_{N_h - 1}}{\operatorname{argmin}} \quad c\left(\mathbf{x}_{N_h}(t = t_{N_h, n_c})\right) + \sum_{k=0}^{N_h - 1} c\left(\mathbf{x}_k(t = t_k), \boldsymbol{\tau}_k\right) \tag{5.8}$$

$$\text{s.t.} \quad \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{M}\left(\mathbf{x}_k(t_{k,j})\right) \end{bmatrix} \dot{\mathbf{x}}_k(t_{k,j}) + \begin{bmatrix} \mathbf{0} \\ \mathbf{k}\left(\mathbf{x}_k(t_{k,j})\right) \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{g}\left(\mathbf{x}_k(t_{k,j})\right) \end{bmatrix} + \begin{bmatrix} \dot{\mathbf{q}}_k(t_{k,j}) \\ \boldsymbol{\tau}_k \end{bmatrix} \quad \text{for } k = 0, \ldots, N_h - 1 \tag{5.9}$$

$$\mathbf{p}(t_k) = \mathbf{h}\left(\mathbf{x}_k(t_k)\right) \qquad j = 0, \ldots, n_c \tag{5.10}$$

$$\mathbf{x}_0(0) = \mathbf{x}_0 \tag{5.11}$$

$$\mathbf{x}_{k+1}(t_k) = \boldsymbol{\theta}_{k,n_c} \tag{5.12}$$

$$\mathbf{x}_{\min} \leq \mathbf{x}_k(t_{k,j}) \leq \mathbf{x}_{\max} \tag{5.13}$$

$$\boldsymbol{\tau}_{\min} \leq \boldsymbol{\tau}_k \leq \boldsymbol{\tau}_{\max} \tag{5.14}$$

for parameters $\boldsymbol{\theta} = \left[\theta_{0,0}, \ldots, \theta_{0,n_c}, \ldots, \theta_{k,j}, \ldots, \theta_{N_h,n_c}\right]^T$ and $\tau_0, \ldots, \tau_{N_h-1}$ over a prediction horizon N_h. Equations (5.12) ensure continuity of the solution since t_k is not a collocation point as depicted in Fig. 5. The forward kinematics in equation (5.10) act as servo constraints by enforcing the TCP to follow a reference trajectory $\mathbf{p}(t)$. Joint limits as well as maximum velocities or torques are maintained by equations (5.13) and (5.14).

In order to improve the approximation accuracy and hence the overall solution using Lagrange Polynomials, the number of collocation points or the polynomial order can be increased, respectively. Also, the approximated time interval can be reduced. This is an important factor for obtaining numerically efficient but also precise solutions for equation (5.8). The stability and thus convergence is influenced by the choice of collocation points $t_{k,j}$. For DAEs, as in equation (5.1), collocation points are selected according to the Radau scheme, as these provide L-stability even for numerically stiff problems, see [23].

The problem formulation of equation (5.8) as OCP allows for a more general definition of cost compared to previously defined cost functions for the inverse kinematics calculation in Sect. 4. Therefore, in order to use the full potential in the reduction of power consumption, the power output

$$c(\mathbf{x}, \boldsymbol{\tau}) = P_w = \underbrace{\boldsymbol{\tau}^T \mathbf{K}_a \boldsymbol{\tau}}_{P_{loss}} + \underbrace{\boldsymbol{\tau}^T \mathbf{K}_b \dot{\mathbf{q}}}_{P_{mech}} \tag{5.15}$$

of the drives itself, as described in equation (4.12), serves as cost $c(\mathbf{x}, \tau)$ for the pose optimization.

Since the full power output is used at this point, a more concise consideration of the applicability is provided. Equation (5.15) consists of two terms, the power loss P_{loss} caused in the armature depending on the torque $\boldsymbol{\tau}$ and the system's mechanical power P_{mech}, with $\mathbf{K}_b = \mathbf{I}$ and usually $\mathbf{K}_a \leq \mathbf{I}$. The consideration of gear ratios $\mathbf{K}_g$ yields

$$\boldsymbol{\tau}_m = \mathbf{K}_g^{-1} \boldsymbol{\tau} \tag{5.16}$$

$$\dot{\mathbf{q}}_m = \mathbf{K}_g \dot{\mathbf{q}} \tag{5.17}$$

for the drive torque $\boldsymbol{\tau}_m$ and drive velocities $\dot{\mathbf{q}}_m$, respectively. Applying this to equation (5.15) yields

$$c(\mathbf{x}, \boldsymbol{\tau}) = P_w = \underbrace{\boldsymbol{\tau}^T \mathbf{K}_g^{-T} \mathbf{K}_a \mathbf{K}_g^{-1} \boldsymbol{\tau}}_{P_{loss}} + \underbrace{\boldsymbol{\tau}^T \mathbf{K}_g^{-T} \mathbf{K}_b \mathbf{K}_g \dot{\mathbf{q}}}_{P_{mech}}, \tag{5.18}$$

where the gear ratios $\mathbf{K}_g$ are cancelled out in the mechanical power P_{mech} calculation and the power losses P_{loss} are reduced quadratically. It is noted, that with high gear ratio the power losses P_{loss} basically vanish. That is, the mechanical power P_{mech} turns to be the most dominant term in the cost function and holding torques, e.g. due to the generalized gravitational forces $\mathbf{g}(\mathbf{q})$ when the system is at rest, become less imminent in the energy consumption. Moreover, in that case a regularization of $\boldsymbol{\tau}$ is non-existent in cost function (5.18), which

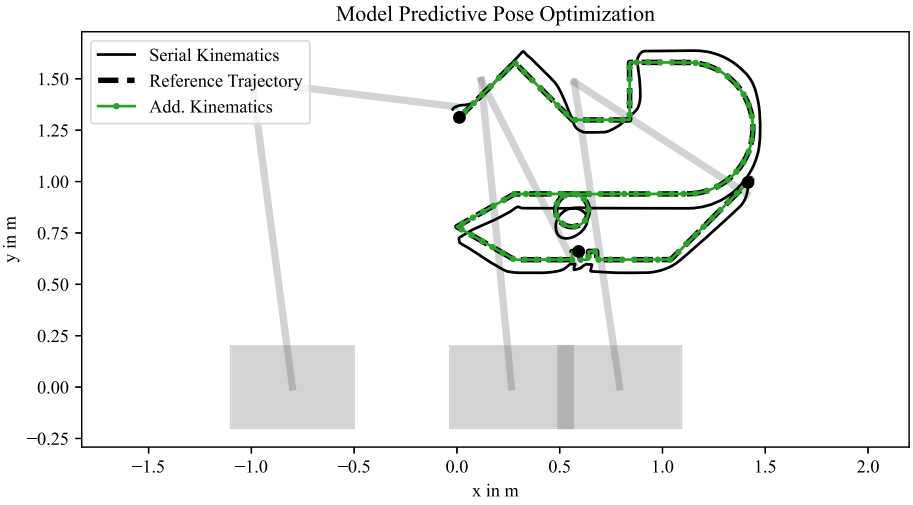

Fig. 6 The system with additional kinematics following the reference trajectory. The poses are calculated using model predictive pose optimization (MPPO)

can lead to infeasible solutions to the overall OCP in equation (5.8). For good feasibility as well as comparability to Sect. 4, where the gravitational load is the main contribution for energy reduction, no gear transmission is considered for the system in this work.

Additionally, the full dynamic model of the system is used in the OCP formulation in equation (5.2). A more refined model of the system including friction and internal gear dynamics can have a major impact from a control system point of view. However, the focus of this work is pose optimization and thus the system's dynamics are simplified to the equations of motion from equation (2.1). The solution to this problem is a joint space trajectory $\mathbf{q}(t)$, that can be used as a reference for the underlying control of each respective joint. With cascaded PID control, for example, being one of the common control strategies that are used in the industrial field.

The OCP in equation (5.8) is solved using the *Casadi* Toolbox for optimal control [24]. At this point the robot including additional kinematics is considered, and the solution 'C' denoted as 'Model Predictive Pose Optimization' (MPPO) is presented in Fig. 6. The path followed by the end effector, i.e. the additional kinematics, is depicted together with the reference trajectory.

In order to study the effect of incorporating the kinematic constraints directly into the problem formulation, the trajectory following error $e(t)$, i.e. the difference of reference trajectory to the end effector's trajectory

$$e(t) = \|\mathbf{p}(t) - \mathbf{h}\left(\mathbf{q}(t)\right)\| \tag{5.19}$$

is calculated. Accordingly, the root mean squared error (RMSE)

$$e_{rmse} = \sqrt{\frac{1}{n}\sum_{k=0}^{n-1} \|\mathbf{p}_k - \mathbf{h}\left(\mathbf{q}_k\right)\|^2} \tag{5.20}$$

is calculated for a quantitative evaluation. Both, the trajectory following error and the RMSE are depicted in Fig. 7 for two MPPO solutions using different time discretization h. The

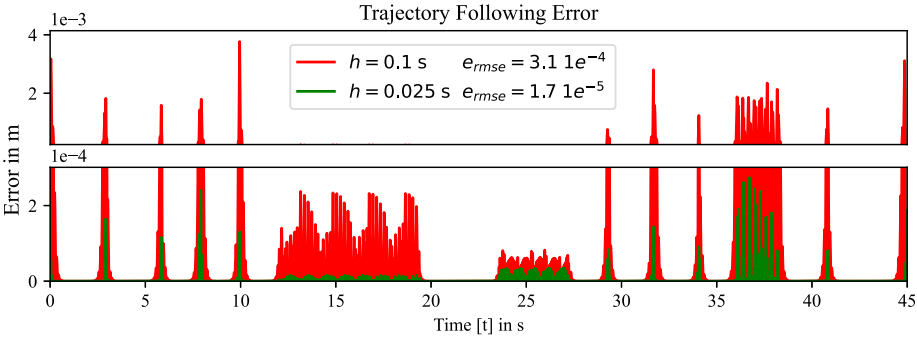

Fig. 7 Trajectory following error for the MPPO solution at two different step sizes h

error $e(t)$ is zero at time points t_k where the kinematic constraints of equation (5.10) hold. At points when sharp corners are traversed or when the reference trajectory deviates from a straight line, the error on the interpolated intervals $t \in (t_k, t_{k+1})$ is greatest. This follows from the trajectory planning in Sect. 3. Here, the waypoints of the reference path are interpolated by B-splines, which can lead to sharp corners. In conventional machining, the reference path is generated in CAD/CAM software, that incorporates the tool geometry for a feasible reference path that is interpolated by polynomials of order $n = 3$ at least. Overall, the OCP formulated as DAE with kinematic constraints leads to a guaranteed trajectory following at each time step t_k and small deviations along the interpolated intervals. Moreover, by quartering the step size h, the RMSE is reduced by a magnitude.

Figure 6 also shows the path followed by the serial kinematics part, i.e. the robot's arm. It can indeed be observed that the additional kinematics are used to enable the robot to round corners of the robot arm trajectory while following the TCP trajectory and thus reduce the consumed energy of the system's high inertia part. The overall trajectory cost is calculated by integration of the drives' power output

$$E(t) = \int_{t_0}^{t_f} P_w dt, \tag{5.21}$$

i.e. calculating the total energy used for the task. In order to compare the consumed energy to the results from the previous Sect. 4, equation (5.21) needs to be calculated for the inverse kinematics solution 'B'. Therefore, the poses $\mathbf{q}$ obtained through equation (4.15) are differentiated numerically to obtain the joint velocities $\dot{\mathbf{q}}$ and accelerations $\ddot{\mathbf{q}}$. By rearranging and inserting these into equation (2.1), the inverse dynamics $\boldsymbol{\tau}$ are calculated. With this at hand, the overall trajectory cost for the Energy IK solution is obtained and presented along the system states in Fig. 8. It shows, that a further reduction in energy consumption is possible by utilizing more advanced algorithms and exploiting the extra degrees of freedom of the additional kinematics.

6 Model predictive pose and trajectory optimization

To build on the results for the model predictive pose optimization, it is further analyzed to optimize the overall trajectory, by adapting the timing law $s(t)$ as mentioned in Sect. 3. This approach is also followed in [20] and generally known as time scaling, see [12]. In terms

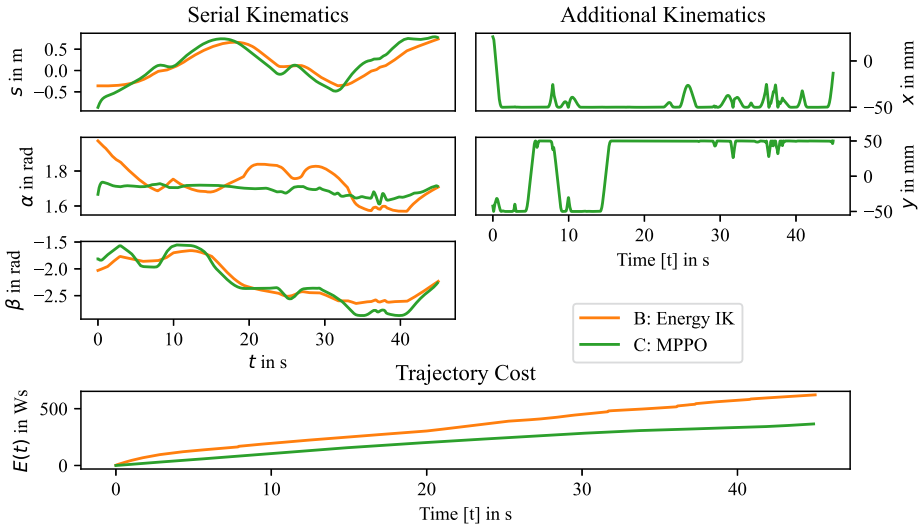

Fig. 8 The system states for the entire trajectory for both the solution to the presented model predictive pose optimization (C: MPPO) as well as for the inverse kinematics problem utilizing redundancy (B: Energy IK) as reference. The trajectory cost refers to the consumed Energy $E(t)$ of the system

of machining, the path velocity $\dot{s}(t)$ is referred to as feed rate. It determines the cutting velocity, e.g. in a milling process, and has a significant impact on the component quality. The efficiency of a machining process is often measured in the time required, thus the highest possible feed rate can increase efficiency but is limited to process restrictions. In the context of this work it is analyzed whether faster machining not only increases efficiency but also reduces energy consumption. Although the instantaneous power output is temporarily higher with a faster feed rate, it is applied over a shorter period of time. Therefore, the timing law $s(t)$ is added as additional degree of freedom, i.e. as optimization variable,

$$\underset{\boldsymbol{\theta},\boldsymbol{\theta}^s,\boldsymbol{\tau}_0,\ldots,\boldsymbol{\tau}_{N_h-1}}{\operatorname{argmin}} \quad c\left(\mathbf{x}_{N_h}(t=t_{N_h,n_c}), s(t_{N_h,n_c})\right) + \sum_{k=0}^{N_h-1} c\left(\mathbf{x}_k(t=t_k), \boldsymbol{\tau}_k, s(t_k)\right) \tag{6.1}$$

$$\text{s.t.} \quad \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{M}\left(\mathbf{x}_k(t_{k,j})\right) \end{bmatrix} \dot{\mathbf{x}}_k(t_{k,j}) + \begin{bmatrix} \mathbf{0} \\ \mathbf{k}\left(\mathbf{x}_k(t_{k,j})\right) \end{bmatrix}$$

$$= \begin{bmatrix} \mathbf{0} \\ \mathbf{g}\left(\mathbf{x}_k(t_{k,j})\right) \end{bmatrix} + \begin{bmatrix} \dot{\mathbf{q}}_k(t_{k,j}) \\ \boldsymbol{\tau}_k \end{bmatrix} \qquad \text{for } k = 0, \ldots, N_h - 1 \tag{6.2}$$

$$\mathbf{p}(s(t_k)) = \mathbf{h}\left(\mathbf{x}_k(t_k)\right) \qquad j = 0, \ldots, n_c \tag{6.3}$$

$$\mathbf{x}_0(0) = \mathbf{x}_0 \tag{6.4}$$

$$\mathbf{x}_{k+1}(t_k) = \boldsymbol{\theta}_{k,n_c} \tag{6.5}$$

$$s_{k+1}(t_k) = \boldsymbol{\theta}^s_{k,n_c} \tag{6.6}$$

$$\mathbf{x}_{\min} \leq \mathbf{x}_k(t_{k,j}) \leq \mathbf{x}_{\max} \tag{6.7}$$

$$s_{\min} \leq s_k(t_{k,j}) \leq s_{\max} \tag{6.8}$$

$$\boldsymbol{\tau}_{\min} \leq \boldsymbol{\tau}_k \leq \boldsymbol{\tau}_{\max} \tag{6.9}$$

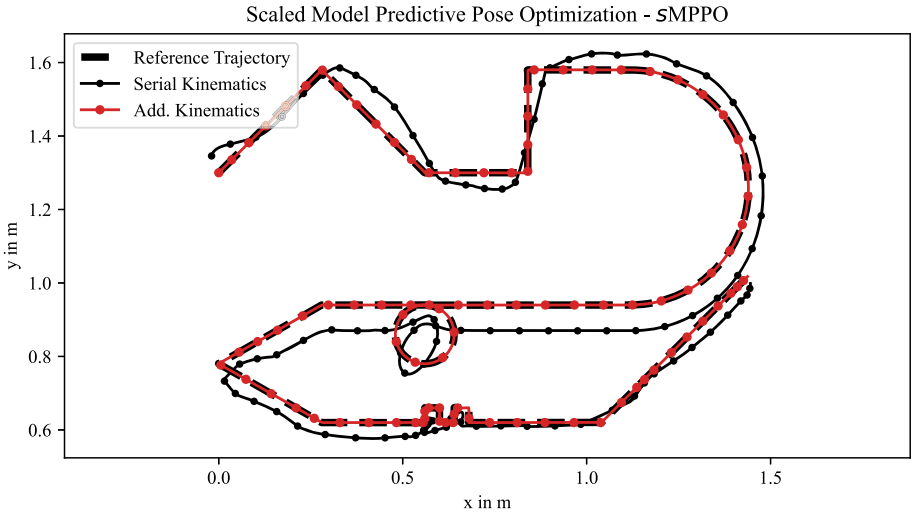

Fig. 9 The system with additional kinematics following the reference trajectory. The poses are calculated using model predictive pose optimization including time scaling (*s*MPPO)

to the problem in equation (5.8). Hereby, also the timing law $s(t)$ is interpolated by collocation parameters $\boldsymbol{\theta}^s$ and the cost function is extended to

$$c(\mathbf{x}, \boldsymbol{\tau}, s(t)) = P_w - \lambda \cdot s(t)^2 \tag{6.10}$$

with weighting factor $\lambda : \lambda > 0$, in order to motivate progress along the trajectory. At this point, due to a beforehand unknown time horizon, problem (6.1) is solved in a model predictive manner. That is, only the first time step t_k of the solution to the OCP (6.1) is used. The OCP is then solved for the next time step t_{k+1} with the previous solution as initial condition. Hereby, the prediction horizon is set to $N_h = 65$ and Lagrange Polynomials of order $n_c = 4$ are chosen for the interpolation of each time step with length $h = 0.025$ s. The results for the Model Predictive Pose Optimization including time scaling (*s*MPPO) are shown in Fig. 9. Likewise to Fig. 6, it shows the trajectory followed by the additional kinematics, i.e. the TCP, as well as the trajectory followed by the serial kinematics part, respectively. The markers in Fig. 10 are equidistant in time, meaning that short distances resemble a slow path velocity and vice versa. Consequently, deceleration can be observed especially before sharp turns. The utilization of the additional kinematics is even stronger compared to the MPPO solution in Fig. 6 and can be observed particularly when driving corners. Here, the majority of the movement is performed by the additional kinematics and the serial kinematics follows a smoothed, rounded contour. Through time scaling, the trajectory is followed nearly twice as fast compared to the linear reference from Sect. 3. This is shown in Fig. 10, where acceleration and deceleration is clearly visible compared to the timing law $s_{\text{ref}}(t)$ of all previous solutions. Also the overall energy consumption, as shown in Fig. 11, is reduced by time scaling the reference trajectory and is discussed further in the following.

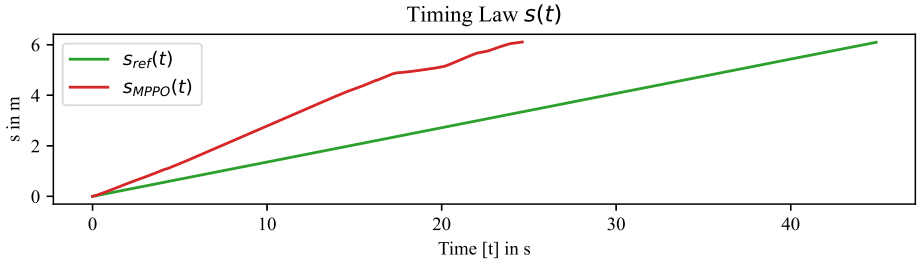

Fig. 10 The adapted timing law $s_{\mathrm{MPPO}}(t)$ compared to s_{ref} of all previous solutions

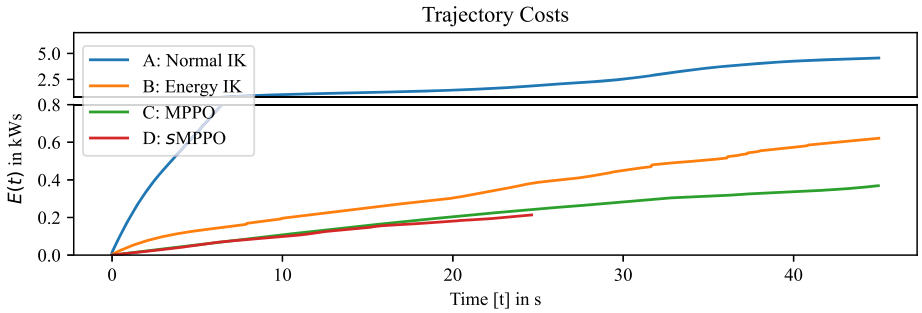

Fig. 11 The consumed energy $E(t)$ for all methods presented in this work

7 Discussion

The results obtained in this work show that even a small change in the robot's pose can significantly minimize the energy consumed when following a machining path. Figure 11 lists all the methods analyzed and their energy consumption for comparison.

In the first attempt, relating to the standard inverse kinematics algorithm versus the algorithm utilizing redundancy for energy minimization of Sect. 4, the consumed energy is reduced from roughly 4.6 kWs down to 0.62 kWs. This corresponds to a consumption of 13.5% compared to the original energy for the standard case and is mainly achieved by using the linear rail or the cart of the simplified system for an energetically advantageous repositioning. By including additional kinematics to the system, the energy consumption is further minimized to circa 0.37 kWs. Finally, further energy is saved by also optimizing the trajectory through the adaption of timing law $s(t)$. Hereby, the entire trajectory is completed in 25.1 seconds, almost half the time of the reference trajectory at 45 seconds. This in turn results in a nearly halved energy consumption of 0.22 kWs. Compared to the initial consumption, this corresponds to a consumption of only 4.78%, i.e. a total energy saving of 95.22%.

It should be noted that such a drastic effect is only possible due to a simplified model of the drives and the general system, e.g. by not taking gear ratios into account, as discussed in Sect. 5. Figure 12 shows the total consumed energy when only taking into account the mechanical power P_{mech} in equation (5.21), i.e. for a system with very high gear ratio. All solutions perform better when compared to the normal IK solution, yet the methods do not deviate as much. The scaled MPPO solution continues to stand out, as it is more time, and at the same time, energy efficient. Overall, it can be summarized that it is worth exploiting

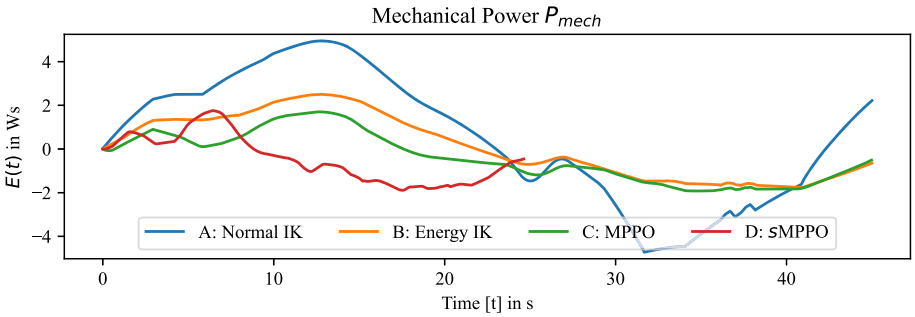

Fig. 12 The mechanical power P_{mech} part of the total consumed energy $E(t)$

the excess degrees of freedom for subtasks like energy efficiency and, if possible, even increasing the existing redundancy by adding extra kinematics.

8 Conclusion

In this work a novel approach for pose optimization of hybrid kinematic robots in machining has been developed. Pose optimization is done by alternating the inverse kinematics calculation. First, a standard inverse kinematics algorithm is presented, which is then extended to exploit redundancy for energy minimization. In order to also address the additional degrees of freedom of hybrid kinematic robots, a model predictive pose optimization (MPPO) is presented. Here, the full drives' power output is used as cost function, which is then minimized by solving a nonlinear optimization problem in DAE formulation. This method is then extended to also optimize the trajectory by adapting the timing law $s(t)$ to increase efficiency.

While in literature often pick and place tasks serve as application for robot's trajectory optimization, this approach focuses on energy optimality in machining tasks. Thereby a reference path is provided, and path accuracy is crucial. Thanks to the DAE formulation, path accuracy can be guaranteed. This is not necessarily the case in learning based methods. In addition, in similar optimization problems for trajectory optimization, the trajectory deviation is often included in the cost function along with other objectives. This results in a multi-objective problem, where one objective is mostly achieved at the expense of the other. Compared to approaches in which the jerk and acceleration of the system are explicitly limited in order to avoid high torques when traversing corners, this is implicitly achieved with the method presented here. Due to the hybrid kinematic structure of the system, high jerk and acceleration can be driven by the lower inertia of the additional kinematics part. Thanks to additional time scaling, path velocity, acceleration and jerk are adapted for energy optimal traversing while also increasing efficiency.

From an industrial point of view, conventional CNC machines typically do not possess any extra degrees of freedom and therefore these are not yet taken into account in conventional CAD/CAM programs that are used for path generation in machining. So far, robot controls are not equipped with an "energy-saving mode" either. Especially for production systems this can have a positive effect for a company's environmental footprint and thus contribute to CO_2 reduction and national climate goals.

Finally it is noted, that Model Predictive Pose and Trajectory Optimization can be easily adapted to other subtasks, such as minimum process forces, by adjusting the cost function.

Acknowledgements The majority of the presented work was conducted with funding of the Federal Ministry of Economic Affairs and Climate Action and supervision of the Programme Management Agency for Aviation Research, German Aerospace Center (DLR) within the project MöWE - "Maximal Ökoeffiziente Wing Entwicklung" (FKZ:20D2210C) and ADMAS – "Advanced Machining And Sealing" (FKZ: 20W1901H). The authors are grateful for the valuable expert information provided by the project partners and every member of the project team Integrated Production Systems, as well as the head of department Dr. Dirk Niermann at Fraunhofer IFAM Stade.

Author contributions P.C. wrote the manuscript text, prepared the figures and is responsible for the main contribution of all data and analysis. R.S. supervised the writing process and is responsible for the conceptional design. C.M. and C.B. assisted in writing and working on the related project on which the paper is based (conceptual work, review and discussions). All authors reviewed the manuscript.

Funding information Open Access funding enabled and organized by Projekt DEAL.

Data availability No datasets were generated or analysed during the current study.

Declarations

Competing interests The authors declare no competing interests.

References

1. Pan, Z., Zhang, H., Zhu, Z., Wang, J.: Chatter analysis of robotic machining process. J. Mater. Process. Technol. **173**(3), 301–309 (2006)
2. Cordes, M., Hintze, W.: Offline simulation of path deviation due to joint compliance and hysteresis for robot machining. Int. J. Adv. Manuf. Technol. **90**(1), 1075–1083 (2017)
3. Verl, A., Valente, A., Melkote, S., Brecher, C., Ozturk, E., Tunc, L.T.: Robots in machining. CIRP Ann. **68**(2), 799–822 (2019)
4. Möller, C., Schmidt, H.C., Koch, P., Böhlmann, C., Kothe, S.M., Wollnack, J., Hintze, W.: Machining of large scaled CFRP-parts with mobile CNC-based robotic system in aerospace industry. Proc. Manuf. **14**, 17–29 (2017)
5. Hansen, S., Hamann, T., Boehlmann, C., Moeller, C., Hintze, W.: Improved dynamic behaviour of industrial robots through hybrid drives. In: ISR Europe 2023; 56th International Symposium on Robotics, pp. 390–397. VDE (2023)
6. Muth, M., Carstensen, P., Hintze, W., Gnadt, S., Brillinger, C., Möller, C., Böhlmann, C.: Sensor-guided machining of large-scale CFRP components based on resin transfer molded features. In Optical Measurement Systems for Industrial Inspection XIII (Vol. 12618, pp. 45–61). SPIE Syst. Dyn. **1**, 149–188 (2023)
7. Chemnitz, M., Schreck, G., Krüger, J.: Analyzing Energy Consumption of Industrial Robots. In: ETFA2011, pp. 1–4. IEEE (2011)
8. Palomba, I., Wehrle, E., Carabin, G., Vidoni, R.: Minimization of the energy consumption in industrial robots through regenerative drives and optimally designed compliant elements. Appl. Sci. **10**(21), 7475 (2020)
9. Meike, D., Ribickis, L.: Energy efficient use of robotics in the automobile industry. In: 2011 15th International Conference on Advanced Robotics (ICAR), pp. 507–511. IEEE (2011).
10. Paes, K., Dewulf, W., Vander Elst, K., Kellens, K., Slaets, P.: Energy efficient trajectories for an industrial ABB robot. Proc. CIRP **15**, 105–110 (2014)

11. Boscariol, P., Caracciolo, R., Richiedei, D., Trevisani, A.: Energy optimization of functionally redundant robots through motion design. Appl. Sci. **10**(9), 3022 (2020)
12. Carabin, G., Wehrle, E., Vidoni, R.: A review on energy-saving optimization methods for robotic and automatic systems. Robotics **6**(4), 39 (2017)
13. Schiehlen, W., Eberhard, P.: Applied Dynamics, vol. 57. Springer, Berlin (2014)
14. Manipulating industrial robots - Performance criteria and related testmethods: ISO 9283 : 1998, DIN (1999)
15. Morozov, M., Riise, J., Summan, R., Pierce, S.G., Mineo, C., MacLeod, C.N., Brown, R.H.: Assessing the accuracy of industrial robots through metrology for the enhancement of automated non-destructive testing. In: 2016 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI), pp. 335–340. IEEE (2016).
16. Siciliano, B., Sciavicco, L., Villani, L., Oriolo, G.: Robotics: Modelling, Planning and Control. Springer (2010)
17. Schneider, U., Posada, J.R.D., Verl, A.: Automatic pose optimization for robotic processes. In: 2015 IEEE International Conference on Robotics and Automation (ICRA), pp. 2054–2059. IEEE (2015).
18. Park, J.S.: Motion profile planning of repetitive point-to-point control for maximum energy conversion efficiency under acceleration conditions. Mechatronics **6**(6), 649–663 (1996)
19. Hargraves, C.R., Paris, S.W.: Direct trajectory optimization using nonlinear programming and collocation. J. Guid. Control Dyn. **10**(4), 338–342 (1987)
20. Holzmann, P., Pfefferkorn, M., Peters, J., Findeisen, R.: Learning energy efficient trajectory planning for robotic manipulators using Bayesian optimization. In: 2024 European Control Conference (ECC), pp. 1374–1379. IEEE (2024)
21. Levine, W.S., Grüne, L., Goebel, R., Rakovic, S.V., Mesbah, A., Kolmanovsky, I., et al.: Handbook of Model Predictive Control. Control Engineering. Birkhäuser, Cham (2018)
22. Biegler, L.T.: Nonlinear Programming: Concepts, Algorithms, and Applications to Chemical Processes. Society for Industrial and Applied Mathematics (2010)
23. Wanner, G., Hairer, E.: Solving Ordinary Differential Equations II, vol. 375, p. 98. Springer, New York (1996).
24. Andersson, J.A., Gillis, J., Horn, G., Rawlings, J.B., Diehl, M.: CasADi: a software framework for nonlinear optimization and optimal control. Math. Program. Comput. **11**, 1–36 (2019)