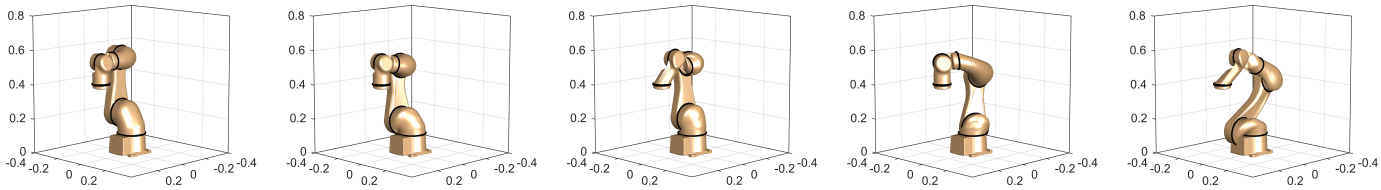
2023 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)
June 28-30, 2023. Seattle, Washington, USA

Towards Task Tailored Articulated Robot Designs

Tom Lefebvre*, Jolan Wauters, Frederik Ostyn and Guillaume Crevecoeur



***Abstract*—The objective of this research is to demonstrate that conventional robot designs, typically characterised by orthogonal axis placement, can be improved, tailoring the design to a task or task ensemble. For that purpose we develop a co-optimization pipe-line specialized to the combined optimization of design and task execution. Our pipe-line pursues high realism by proposing a link deformation strategy that preserves manufacturability of the deformed links, specifically leaving the actuators unaffected. We demonstrate the approach on a proprietary industrial robot, tailoring the design to two time optimal task definitions. We demonstrate that task execution times can be reduced up to 15%.**

***Index Terms*—Design Optimization in Mechatronics, Rapid Prototyping, Modelling and Design of Mechatronic Systems**

I. Introduction

Robotic manipulators are widely deployed in production environments to automate laborious and dangerous tasks such as assembly, welding, painting, material handling, etc. Optimizing the robot trajectory in order to minimize features such as energy consumption, actuator loading, or execution time is standard practice to economize production. The design of the robot itself however, in particular the geometry of its links, is typically not considered in this process. Instead, standard, "off-the-shelf" manipulators from robot manufacturers are used. These designs, most oftenly characterized by orthogonal consecutive joint axes, have been proven effective to accommodate general practices but are suboptimal for specific tasks or task ensembles. By tailoring the robot design to its intended task, functional performance improvements can still be achieved.

Most industrial manipulators consist of a kinematic chain of revolute and/or prismatic joints [1]. Two common topologies are the articulated robot, consisting solely from revolute joints and the SCARA robot which has one prismatic and two revolute joints. Articulated robots are typically equipped with a spherical wrist, meaning the last three joints intersect in a common point. The other joints are usually organised in an orthogonal fashion. This particular design principle simplifies the kinematic analysis [2]. Revolute joints can be realized placing the motor off-joint, transmitting torque to axis via complicated drives. Contemporary, motors are typically placed in joints for easy construction and cabling. A key challenge is thus to realize high torque density. Therefore high gear ratio transmissions such as strain wave gears are a common choice [3]. Overload clutches are also integrated to prevent mechanical failure after collision [4]. Evidently a lot of research goes into designing an actuator that is both performant, compact and light. By consequence, the actuator is a component whose design is not easily integrated in a co-optimization framework.

The remark above does not hold for the mechanical links whose sole purpose is to connect the joints. Considering modern manufacturing practices and standards, it seems possible to deviate from standard designs without provoking unnecessary high manufacturing costs. This would allow to customize the link morphology depending on the indented task that the robotic manipulator will execute. Furthermore the corresponding functionality improvement (energy consumption, execution time) would easily outweigh the increased manufacturing or purchase cost if the robot is to repeat its task indefinitely.

When developing a new system, usually the process moves from concept to product through an iterative loop of design, implementation, and testing (the well-known V-diagram). Model-based design attempts to exploit behavioural system models in the development cycle. The paradigm pursues *first time right* in the sense that the number of iterations and associated costs are reduced leading to innovative systems with improved and sometimes even new functionalities. Historically, the design and control were optimized separately. The theory of combined control and design, or co-optimization [5], links the design with e.g. motion optimization approaches and treats them simultaneously. The resulting extended design space may lead to coupled design and control solutions that are not obtainable through conventional sequential approaches where the design is determined first, based on generic guidelines and design principles, and only afterwards a suitable control is developed to improve its functionalities [5, 6, 7]. Although model based co-optimization is a promising design approach, it is entirely reliant on the presence of accurate models, fast simulation and efficient optimization algorithms [8]. Conveniently the first conditions are satisfied with industrial robots.

A considerable body of earlier work has investigated this direction. Co-optimization approaches for robots with flexible links have been documented as early as the nineties [9, 10]. Subsequently the idea spread to more standard design

*Corresponding author.
T. Lefebvre, J. Wauters, F. Ostyn and G. Crevecoeur are with the Dynamic Design Lab (D^{2}LAB) of the Department of Electromechanical, Systems and Metal Engineering, Ghent University, B-9052 Ghent, Belgium. E-mail: {tom.lefebvre,jolan.wauters,frederik.ostyn,guillaume.crevecoeur}@ugent.be.
T. Lefebvre, J. Wauters, F. Ostyn and G. Crevecoeur are member of core-lab MIRO, Flanders Make, Belgium.

though the majority of the literature is dominated by planar designs and/or low dimensional control parametrizations [11, 12, 13, 14]. More recent studies consider co-design with policy learning [15, 16]. One notable study [17] explores 3D motion using point masses to parametrize the links. This approach leads automatically to counterbalancing principles which have also been considered for co-optimization [18]. The only study to consider full rigid motion deformation, including non-orthogonal axes, is [19].

In this work we document a specialized co-optimization framework for simultaneous articulated robot design and time optimal motion planning. We initialize the design cycle with a proprietary 6 DOF articulated robot [20]. Our link deformation approached is closely related to [19] however significantly improves realism. In particular our approach preserves manufacturability of the design by leaving the actuators and mounting geometry unaffected. We demonstrate the approach on two time optimal motion optimization scenarios and document reduction in execution time up to 15%.

II. Methodology

A. Modelling

We start our discussion by reviewing the quantitative models required to set-up the simulation models and motivate some of the choices that are made along the way.

In this work we initialize a design with the nominal design of a given industrial manipulator. We then develop an approach to parametrise the robot design by kinematically deforming the individual links. In our discussion therefore we first elaborate in great detail how an isolated deformed link is parametrized. Once this model component has been established, the other modelling components are straightforward.

All joints and links are labelled with an index i running from 1 to n with n the total degree-of-freedom. The base frame, which is not actuated, is labelled with the index 0. Further, deformed quantities will be highlighted with a marker, '$\star$'.

1) Deformed links: The physical characteristics of an isolated link are determined by its kinematic and inertial properties. Parametrization of deformed kinematic properties is relatively straightforward. The more challenging part is to translate the kinematic link deformation into a physically accurate deformation of the link's inertial properties.

a) Geometric properties: It is common knowledge that the kinematic properties of serial linkage manipulators can be efficiently modelled relying on the decomposition properties of homogenous transformation matrices. From a kinematic perspective a link can thus be described by a homogenous transformation matrix $\mathrm{Q}(q) \in \mathfrak{SE}(3)$ linking the frame of reference attached to the present link to the frame of reference attached to the following link. Traditionally this transformation matrix is parametrized through the Denavit-Hartenberg convention. The actuation of the i-th joint is encoded in the i-th transformation matrix through a variable rotation about the local z-axis. The transformation from the intermediate rotated frame of reference to the frame of reference attached to the next link is then determined by 4 parameters describing the associated rigid motion. In this work we generalise this rigid motion to the homogenous transformation Q and define a geometric link deformation by introducing a secondary rigid motion, $\Delta \in \mathfrak{SE}(3)$. A complete parametrized kinematic link is then described by the homogenous transformation matrix, $\mathrm{Q}^\star(q;\delta)$, with Q the nominal transformation matrix.

$$\mathrm{Q}^\star(q;\delta) = \mathrm{Rot}_z(q) \cdot \mathrm{Q} \cdot \Delta(\delta) \tag{1}$$

To describe the deformation, Δ, we make use of the exponential map, parameterizing the rigid motion using a vector $\delta = (\delta_v, \delta_\omega) \in \mathcal{D} \subset \mathbb{R}^6$. We make use of the hat operator to represent the cross product as a transformation matrix.

$$\Delta(\delta) = \exp\left(\begin{bmatrix} \hat{\delta}_\omega & \delta_v \\ 0^\top & 0 \end{bmatrix}\right) \tag{2}$$

b) Inertial properties: The inertial properties of a rigid body with respect to a body fixed local frame of reference are determined by its mass, $m \in \mathbb{R}_{\geq 0}$, its centre of mass, $r \in \mathbb{R}^3$, and its inertial tensor, $\mathrm{I} \in \mathbb{S}^3$, where $\mathbb{S}^n$ denotes the set of positive definite matrices with dimension $n \times n$. The goal is here to determine the inertial properties of the deformed link as a function of the geometric deformation parameter Δ or the parameters δ. We assume the inertial properties of the nominal design are known, for example obtained through CAD software. Second, we assume access to a mesh of the geometry of the nominal link as are commonly available for visualization purposes (Fig. 1a). Such a mesh is determined by N position coordinates, $\{p_j\}$, expressed in the local frame of reference.

We then propose the following calculation procedure. First we associate a point mass, m_j, to each of the individual points in the mesh. We determine the individual point masses m_j using the quadratic program given below, penalizing deviation from the mean mass $\bar{m} = m/N$ and constraining the mass distribution to satisfy the given nominal inertial properties. We obtain the weighted mesh, $\{m_j, p_j\}$. The inertial properties of the deformed link can be determined by deforming the individual point masses from the mesh, (see Fig. 1b)

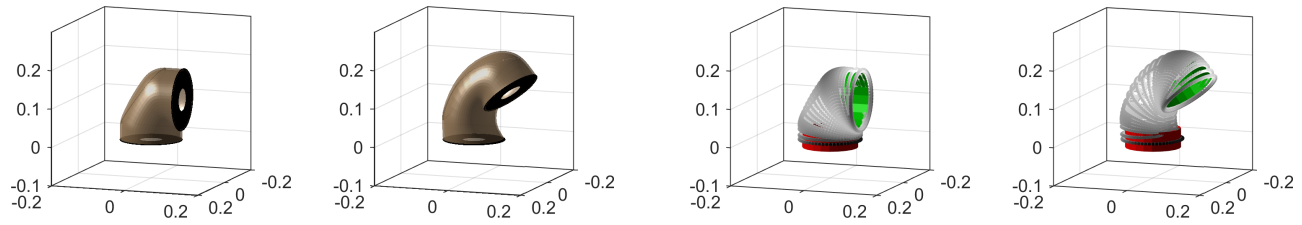
$$\begin{aligned} \{m_j\} &= \arg\min_{\{m_j\}} \tfrac{1}{2} \sum\nolimits_j (m_j - \bar{m})^2 \\ \text{s.t. } & \begin{cases} m = \sum_j m_j \\ m \cdot r = \sum_j m_j p_j \\ \mathrm{I} = \sum_j m_j \left(p_j^\top p_j \mathrm{E} - p_j p_j^\top\right) \end{cases} \end{aligned} \tag{3}$$

with E the identify matrix.

To determine the position coordinates of the point masses after deforming the link, we propose the following strategy. This strategy applies to any point p expressed in the *local frame of reference*. Here P denotes the homogeneous coordinate representation of the position coordinate p. The parameter α is a local deformation factor that determines to what extent a point experiences the full rigid motion deformation $\Delta(\delta)$.

$$P^\star(\delta) = \mathrm{Q} \cdot \Delta(\alpha \cdot \delta) \cdot \mathrm{Q}^{-1} \cdot P \tag{4}$$

Our strategy to determine the local deformation factor α is based on the construction of two flanges associated to the input and output shaft of the individual links. These flanges are determined by the region between two concentric circles perpendicular to the input, e_1, and output axis, e_2, and whose



(a) Illustration of nominal and deformed geometries. We associate an annulus perpendicular to the input and output shafts. Our deformation strategy is so that any points onto the input or output circumference is either unaffected by the deformation, or, experiences the full deformation, respectively. Thickness is exaggerated for illustration.

(b) Demonstration of the weighted mesh. The mass distribution is exemplified by greyscale, the heavier points presented in black, the lighter points in white. The input rotor and output stator are visualized in red and green respectively. The rotor and stator position is either unaffected by the deformation, or, experiences the full deformation.

Fig. 1: Illustration of the proposed link deformation strategies. Geometrical deformation strategy (left) and inertial deformation strategy (right).

centres, r_1 and r_2, match the centres of the local frame of references associated to the links. Their outer radii, R_1 and R_2 are chosen so to coincide with the outer hull of the geometric links. Their inner radii are chosen to match the thickness of the output flange. For convenience we parametrize the input and output thickness, t_1 and t_2, rather than the inner radii. We refer to Fig. 1a for an example. We now wish to determine α so that any point on the input circumference experiences no deformation ($\alpha = 0$), and, any point on the output circumference experiences the full rigid motion Δ ($\alpha = 1$). Then after deforming the links, the circumferences of subsequent links should still match. So for any nominal point p, we determine points, p_1 and p_2, closest to the input and output flanges. We determine α as the normalized projection of the vector, $p - p_1$, onto, $p_2 - p_1$. By definition for any point, q, on the input circumference, we have $p_1 = q$. Analogously, for any point, q, on the output circumference, we have $p_2 = q$.

$$\begin{aligned} \alpha(p; \{e_i, r_i, R_i, t_i\}) &= \frac{(p - p_1)^\top (p_2 - p_1)}{\|p_2 - p_1\|^2} \\ p_i &= p^\bullet(p; e_i, r_i, R_i, t_i), \;\; i \in \{1, 2\} \end{aligned} \tag{5}$$

The function $p^\bullet$ can be written as a quadratic program.

$$\begin{aligned} p^\bullet(p; e, r, R, t) &= \arg\min_{p^*} \tfrac{1}{2} \|p^* - p\|^2 \\ \text{s.t.} \; &\begin{cases} 0 = e^\top (p^* - r) \\ R - t \leq \|p^* - r\|^2 \leq R \end{cases} \end{aligned} \tag{6}$$

The desired properties of α follow. This deformation strategy attempts to preserve manufacturability. Determination of the deformation, α, factor by other projection strategies is also possible. E.g. using the exact stator and rotor hull geometry. This approach can be used to determine the deformed weighted mesh, $\{m_j, p_j^\star\}$. In turn this can be used to determine the inertial properties, $\{m, r^\star, \mathrm{I}^\star\}$, of the deformed link using the calculation presented in the constraint of problem (3). The mesh can also be used for visualization purposes, see Fig. 2.

To improve the physical accurateness of the inertial properties of the deformed link, we further identify the three physical objects that contribute to the inertial properties of the nominal link: (1) the rotor and gear of the i-joint, (2) the mechanical link itself and (3) the stator of the $i+1$-th joint. Rather than accounting for the rotor and stator in the weighted mesh, we impose that their geometry remains unaffected by the deformation and account separately for

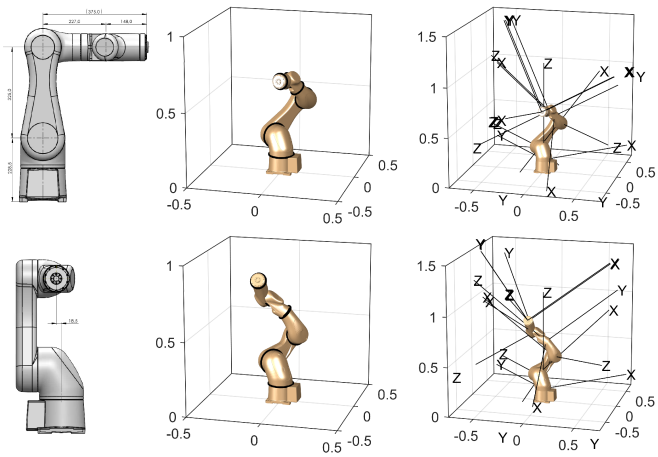


Fig. 2: Illustration of link deformation strategy applied on the full robot design with nominal (left-top) and deformed (bottom) geometries. The nominal meshes are deformed using the procedure described in (4). The deformation strategy pursues physical accurateness and manufacturability.

their respective contributions. Here $\{m_l, r_l^\star, \mathrm{I}_l^\star\}$ denote the inertial properties of the deformed mechanical link. These are calculated according the mesh based strategy detailed above. The values $\{m_r, r_r, \mathrm{I}_r\}$ denote the inertial properties of the rotor. It is assumed that the rotor is unaffected by the deformation. Finally $\{m_s, r_s^\star, \mathrm{I}_s^\star\}$ denote the inertial properties of the stator after deformation. It is assumed that the entire stator experiences the full rigid motion Δ. The benefit of this specific deformation strategy is that the manufacturability of the design is preserved with the possible exception of the mechanical link itself, see Fig. 1b.

$$\begin{aligned} m &= m_r + m_l + m_s \\ m \cdot r^\star &= m_r \cdot r_r + m_l \cdot r_l^\star + m_s \cdot r_s^\star \\ \mathrm{I}^\star &= \mathrm{I}_r + \mathrm{I}_l^\star + \mathrm{I}_s^\star \end{aligned} \tag{7}$$

We recognize that the physical accurateness can still be improved, although much improvement has been made with regard to earlier and similar study [19].

2) Kinematics and dynamics: The forward kinematics, $\mathcal{K} : \mathbb{R}^n \times \mathcal{D}^n \mapsto \mathfrak{SE}(3)$, expressing the end-effector configuration as a function of the joint vector and design parameters follow directly from the decomposition properties of homogeneous transformation matrices. The inverse kinematics, $\mathcal{K}^{-1} : \mathfrak{SE}(3) \times \mathcal{D}^n \mapsto \mathbb{R}^n$, determine the joint values, q^*,

corresponding a desired end-effector configuration represented as a homogenous transformation matrix, T^*. Unfortunately the deformation strategy prevents an analytical solution strategy (as opposed to conventional industrial robots equipped with a wrist joint). Therefore we use a numerical strategy instead [21]. Particularly we use the pseudo-inverse Jacobian methods with Levenberg–Marquardt regularization. Finally we use the Recursive Newton-Euler (RNE) algorithm to calculate the inverse dynamics, $\mathcal{D}^{-1} : \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}^6 \times \mathcal{D}^n \mapsto \mathbb{R}^n$. The inverse dynamics map the instantaneous motion, $(q, \dot{q}, \ddot{q})$, the external wrench, f, exerted to the end-effector, and, the present design to the corresponding actuator torques, τ [22, 23]. Friction is accounted for by superposing a friction torque on the associated joint torque, τ_i. Here we adopt a simple viscous friction model proportional to the joint velocity, $\dot{q}_i$.

B. Co-Optimization framework

1) Motion optimization: We consider two essential motion optimization problems. The first problem considers time-optimal trajectory planning as a subproblem of the full motion planning problem. Here we assume that a desired path is given and the subsequent time-optimal trajectory planning problem aims to realize as high as possible a velocity along the path [24]. The second problem considers the full scale motion planning problem. In both problems our goal is to minimize the total time of execution without violating actuator constraints.

a) Time-Optimal Trajectory Planning: Consider a pre-defined joint trajectory, $q^*(s) : [0, 1] \mapsto \mathbb{R}^n$. Here $s \in [0, 1]$ represents the path coordinate. The dynamics of the trajectory implied by the path then depend on the relation between the path coordinate, $s \in [0, 1]$, and time, $t \in [0, T]$. This problem solves for the optimal path coordinate signal $s^*(t)$ and T^*.

$$
\begin{aligned}
f(s^*(t);\delta) = &\min_{s(t),T} T \\
\text{s.t. } &\begin{cases} s(0) = 0 \\ s(T) = 1 \\ \forall t \in [0,T] : \begin{cases} \dot{s}(t) > 0 \\ |q| \le \overline{q} \\ |\dot{q}| \le \overline{\dot{q}} \\ |\mathcal{D}^{-1}(q,\dot{q},\ddot{q},f;\delta)| \le \overline{\tau} \end{cases} \end{cases}
\end{aligned} \tag{8}
$$

where $\dot{q} = q'\dot{s}$ and $\ddot{q} = q'\ddot{s} + q''\dot{s}^2$.

b) Motion planning: Consider a sequence of reference configurations, $\{\mathrm{T}_j\}$, and, loading signals, $\{f_j(t)\}$. The goal is to find a set of optimal visiting trajectories, $\{q_j^*(t), t \in [0, T_j]\}$, without violating the actuator constraints.

$$
\begin{aligned}
f(\{q_j^*(t)\};\delta) = &\min_{\{q_j(t)\},\{T_j\}} \sum\nolimits_j T_j \\
\text{s.t. } \forall j : &\begin{cases} \mathrm{T}_j = \mathcal{K}(q_j(0)) \\ \mathrm{T}_{j+1} = \mathcal{K}(q_j(T_j)) \\ \forall t \in [0,T_j] : \begin{cases} |q| \le \overline{q} \\ |\dot{q}| \le \overline{\dot{q}} \\ |\mathcal{D}^{-1}(q,\dot{q},\ddot{q},f;\delta)| \le \overline{\tau} \end{cases} \end{cases}
\end{aligned} \tag{9}
$$

2) Design optimization: The co-optimization problem can now be formulated as follows

$$
\begin{aligned}
&\min_{\delta} g(\theta^*;\delta) \\
&\text{s.t. } \theta^* = \arg\min f(\theta;\delta)
\end{aligned} \tag{10}
$$

We overload notation and allow $\delta \in \mathbb{R}^{6n}$ to denote the full set of design variables. The variable θ denotes any variables used to parametrize the motion optimization problem (see later).

Some additional remarks are in place. First of all the present formulation is an example of a nested co-design architecture. A simultaneous co-design architecture would solve for all optimization variables simultaneously. Nested co-design has two major benefits over simultaneous co-design. The functions g and f are not necessarily the same, meaning that additional features can be expressed in the design optimization other than those expressed in the motion optimization. Second each optimization sub problem can be treated using a dedicated solver. Typically the motion optimization problem is solved using a local gradient based optimizer capable of exploiting the temporal locality of the parameter θ. The co-design optimization problem is preferably solved using a global and often derivative free optimizer.

C. Numerical solution

1) Motion planning: For the motion optimization problems we rely on arguably rough optimization strategies whilst more precise and expressive optimization methods are described by the literature. However we argue that the present discretization techniques are sufficient for their intended purpose in this paper. For the optimization of the corresponding Nonlinear Programs (NLPs) we rely on the SQP implementation of MATLAB. Gradients were computed using numerical forward finite differencing with step size 10^{-6}.

a) Time-optimal trajectory planning: We assume the path is determined by a sequence of joint coordinates $\{q_k\}$ (or equivalently by work space configuration $\{\mathrm{T}_k\}$) and optimize for the associated time grid, $\theta = \{t_k\}$. The dynamic feasibility of the trajectory is verified (only) on the same time grid. The velocity and acceleration are approximated using central differencing on the arbitrary time grid. This means that we need to evaluate the velocity, acceleration and torque for the same time grid. To that end we approximate the vicinity of the trajectory $q(t_k)$ by fitting a quadratic function through the present, preceding and following point and evaluate the velocity and acceleration by differentiating the local quadratic approximation. This approach yields the following approximate evaluations of the velocity and acceleration (equivalent to central differencing when a fixed time step is used)

$$
\begin{aligned}
\dot{q}_j \approx &\frac{\Delta q_{k+1,k} t_{k-1}^2 + \Delta q_{k+1,k} t_{k-1}^2 + \Delta q_{k+1,k} t_{k-1}^2}{\Delta t_{k+1,k-1} \Delta t_{k,k-1} \Delta t_{k+1,k}} \\
&- 2\frac{\Delta q_{k+1,k} t_{k-1} t_i + \Delta q_{k,k-1} t_k t_{k+1}}{\Delta t_{k+1,k-1} \Delta t_{k,k-1} \Delta t_{k+1,k}} \\
\ddot{q}_i \approx &\, 2\frac{\Delta q_{k+1,k} t_{k-1} + \Delta q_{k+1,k} t_{k-1} + \Delta q_{k+1,k} t_{k-1}}{\Delta t_{k+1,k-1} \Delta t_{k,k-1} \Delta t_{k+1,k}}
\end{aligned} \tag{11}
$$

where $\Delta x_{i,j}$ is shorthand for $x_i - x_j$. The corresponding joint torques follow from evaluation of the inverse dynamics.

b) Motion planning: We make use of the B-spline framework to parametrize the trajectories $q_j(t)$. A B-spline of order d refers to a function basis where each of the basis functions is represented by a piece-wise polynomial $B_{k,d}$. The basis functions are further characterised by a knot-vector collecting a non-decreasing set of time instants $\{t_0, \ldots, t_k, \ldots, t_m\}$. By choosing $t_0 = 0$ and $t_m = T_j$, the coverage of the B-spline coincides with the time interval $[0, T_j]$. The number of basis functions equals $m+d-1$. For details on construction we refer to [25]. We define the trajectories as a linear combination of the B-spline basis functions and a set of coefficients $\{\theta_{j,k}\}$.

$$q_j(t; \{\theta_{j,k}\}, T_j) = \sum\nolimits_i B_{k,d}\,(t; T_j)\,\theta_{j,k} \tag{12}$$

Due to the nature of problem (9) we can further decompose the full motion planning problem in a number of separated motion planning problems that solve for trajectories in between workspace configuration T_j and T_{j+1} minimizing for the transition time T_j. The inequality constraints are discretised by evaluation on a dense time grid. The decomposition criteria means that we can invoke parallelization techniques.

2) Co-optimization: Solution of the co-optimisation problem depends on the specific motion optimization problem. For the time-optimal trajectory planning it is possible to adopt a simultaneous approach adding the parametric design vector δ to the grid $\theta = \{t_k\}$ used to parametrize the problem. Solution of the full motion planning problem using a simultaneous approach is somewhat more troublesome, specifically because we rely on a decomposition strategy to solve the problem. This means that it is not straightforward to update the design vector simultaneously. Therefore we adopt a nested approach.

To account for the computational cost that comes with the nested formulation the inclusion of Bayesian Optimization (BO) in the outer loop is examined [8]. BO relies on the replacement of the expensive-to-evaluate objective by a cheap-to-sample surrogate. The surrogate is updated iteratively with designs that most strongly contribute to the minimization of the objective and the improvement of the accuracy of the surrogate through the optimization of an infill criterion. Because for certain designs the kinematic constraints cannot be satisfied, we use the approach described in [26] which is able to deal with failed evaluations.

III. Experiments

A. Description of nominal design

In all our experiments we use a 6 DOF industrial manipulator with a proprietary design. The industrial robot arm was developed entirely in our lab [20]. The specifics of the design are not yet published in paper [20] but are comparable to those of the KR 6 R700-2. The nominal design can be verified in Fig. 2. Some of the specifics per link are presented in Table I. The inertial properties were determined using CAD software. The nominal design has overload clutches [4], explaining the slightly heavier and larger actuators (see Table I, m_r and m_s).

TABLE I: Per joint specifications of default design. The rotor of the previous joints and the stator of the next joint are attributed to the present link.

n	m_r [kg]	m_l [kg]	m_s [kg]	$\bar{\dot{q}}$ [rad s^{-1}]	$\bar{\tau}$ [N m]
1	3.5	1.8	4.5	8.4	137
2	3.5	2	2.7	8.4	137
3	1.1	0.5	1.1	10	75
4	0.7	0.4	1.1	13.6	19
5	0.7	0.5	1.3	13.6	19
6	0.7	0.2		13.6	8

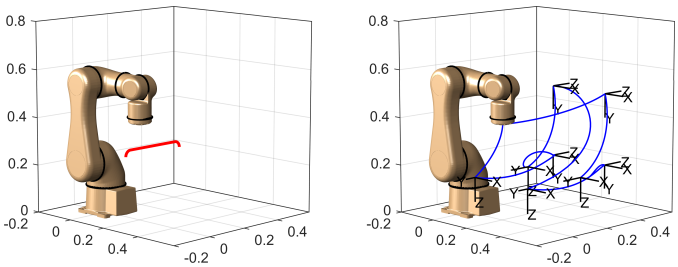


Fig. 3: Illustration motion optimization scenarios (a) and (b). The ISO 9283 trajectory is visualized on the left. The pick-and-place motion optimization problem is visualized on the right.

B. Definition of scenarios

In this work we consider two motion optimization scenarios. Both scenarios are illustrated in Fig. 3.

1) (a): The first scenario considers time-optimal trajectory planning on the reference trajectory described in the ISO 9283 with a $1\,\mathrm{kg}$ payload. The trajectory is composed of a straight line of $305\,\mathrm{mm}$ with downwards rounded beginning and end point with a radius of $25\,\mathrm{mm}$ (Fig. 3, left). The goal is to determine a design that minimizes the execution time. For comparison, the documented cycle time for the KR 6 R700-2 is roughly $435\,\mathrm{ms}$ using a trapezoidal velocity profile.

2) (b): The second scenario considers a pick-and-place task where 4 objects of $10\,\mathrm{kg}$ are picked from a horizontal pallet and are placed in a vertical pallet. The robot has to start and end in a fixed neutral position. The total number of individual pick positions and place positions is both 4. The positions are distributed over a horizontal and vertical square respectively (Fig. 3, right). The total number of visiting trajectories is 9.

C. Results

We practice our co-optimization pipe-line for both scenarios deforming the first 3 and 5 links, leading to 18 or 30 design variables respectively. We will refer to the first as design (A) and to the latter as design (B). The benefit of design (A) over (B) is that the last three joints are unaffected preserving the wrist joint. As a result the inverse kinematics need only to be solved numerically for the position. Clearly design (B) has more degrees of freedom to exploit and is expected to outperform (A). We do not deform the final link due to its negligible dimensions reducing to a final rotation.

The resulting designs are illustrated on the title page. From left to right we present the default design, design (a-A), design (a-B), design (b-A) and finally design (b-B). In Table II an overview is given of the total execution time per scenario and design. Fig. 4 represents the time optimal signals associated to scenario (a). In Fig. 5 we have visualized the optimal motion represented in the work space for scenario (b).

As could be expected design (A) outperforms the default design for either scenario whilst design (B) outperforms design (A) for both scenarios. For scenario (a), a maximum reduction

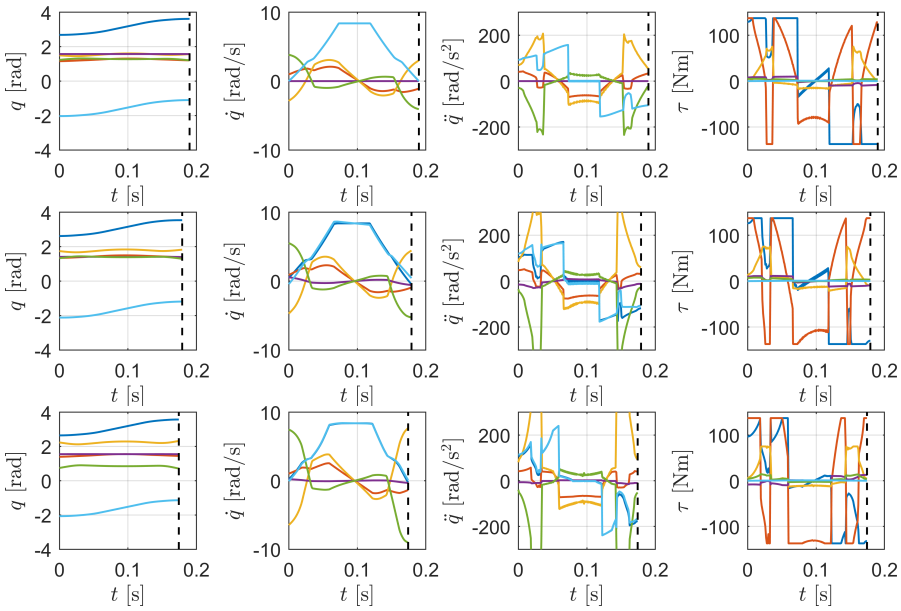


Fig. 4: Comparison of joint, velocity, acceleration and torque signals for the default design (top), design A (middle) and design B (bottom).

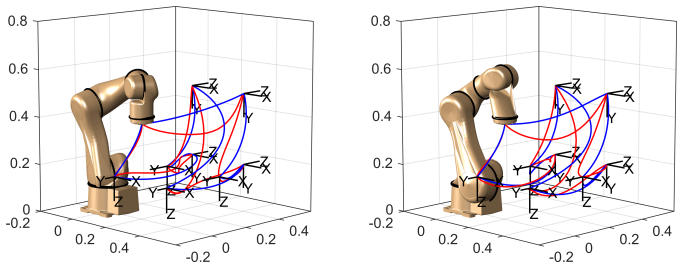


Fig. 5: Nominal and optimal motion for design (A) (left) and (B) (right)

of about 8% was achieved. For scenario (b), the total execution time was reduced with roughly 15%. These gains may appear modest however may have considerable impact when these tasks have to be repeated indefinitely.

Some additional remarks can be made. In scenario (a) one may observe that the optimal design allows to drive the second actuator (see the red curves in Fig. 4) almost perpetually on its maximum allowed continuous torque explaining the reduction in execution time. This effect strengthens gradually with increasing freedom of design. Clearly the co-optimization framework is capable of identifying a design that transmits and distributes the payload optimally over the different joints. This is possible by releasing the constraint on orthogonal axes. Similar observations apply to the result for scenario (b). The comparison in Fig. 5 illustrates that the optimal motion for either design (A) and (B) are quite similar, redirecting the heavily load inwards during transition and lashing out to reach the target position in time. This strategy reduces the effective inertia and hence reduces actuator loading.

IV. Discussion

As illustrated by the present case study, co-optimization of articulated robots shows interesting potential as a general research direction. It was proven that a significant performance

TABLE II: Optimal execution time for different scenarios and designs.

		default	design A	design B
scenario a	[ms]	189.8	179.1	174.1
	[%]	100	94.36	**91.73**
scenario b	[s]	2.29	2.02	1.93
	[%]	100	88.17	**84.14**

improvement could be achieved by adapting the geometry of the robot links, supplementary to the performance improvement resulting standard motion optimization. These results were obtained for the specific task of optimizing the total execution time associated to a task which naturally attempts to drive the actuators on their maximum allowed continuous torque. Hence the task of the co-optimization is to allow redistribution of the loading of the robot so that all actuators can be fully excited simultaneously. This observation implies that the use of alternative objectives, e.g. integrated squared torque, could possible result in more exaggerated robot deformations.

We recognize this study is still incomplete in several aspects. The deformation strategy can be improved. Some ideas were given throughout the text. Furthermore we should check for inherent self collision as can be noticed with the first link in some of the optimal designs. The motion optimization can also be further improved and should account for dynamic self collision which, by coincidence, was not an issue for the considered scenarios. Ultimately our goal is to execute these design principles on a practical set-up. First on a system with servo actuators using 3D printed links. Second we intend to apply this approach on the proprietary industrial robot design.

Acknowledgements

This research received funding from the Flemish Government (AI Research Program). The computational resources and services used in this work were provided by the VSC (Flemish Supercomputer Center).

References

[1] Mark W Spong, Seth Hutchinson, Mathukumalli Vidyasagar, et al. *Robot modeling and control*, volume 3. Wiley New York, 2006.

[2] Khaled Elashry and Ruairi Glynn. An approach to automated construction using adaptive programing. In *Robotic Fabrication in Architecture, Art and Design 2014*, pages 51–66. Springer, 2014.

[3] Pablo López García, Stein Crispel, Elias Saerens, Tom Verstraten, and Dirk Lefeber. Compact gearboxes for modern robotics: A review. *Frontiers in Robotics and AI*, 7:103, 2020.

[4] Frederik Ostyn, Tom Lefebvre, Bram Vanderborght, and Guillaume Crevecoeur. Overload clutch design for collision tolerant high–speed industrial robots. *IEEE Robotics and Automation Letters*, 6(2):863–870, 2021.

[5] Mario Garcia-Sanz. Control co-design: an engineering game changer. *Advanced Control for Applications: Engineering and Industrial Systems*, 1(1):e18, 2019.

[6] Anand P Deshmukh and James T Allison. Multidisciplinary dynamic optimization of horizontal axis wind turbine design. *Structural and Multidisciplinary Optimization*, 53(1):15–27, 2016.

[7] Daniel R Herber and James T Allison. Nested and simultaneous solution strategies for general combined plant and control design problems. *Journal of Mechanical Design*, 141(1), 2019.

[8] Jolan Wauters, Tom Lefebvre, and Guillaume Crevecoeur. Comparative study of co-design strategies for mission-specific design of quadcopters using differential flatness and bayesian optimization. In *2022 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*, pages 703–709. IEEE, 2022.

[9] Haruhiko Asada, J-H Park, and S Rai. A control-configured flexible arm: integrated structure control design. In *Proceedings. 1991 IEEE International Conference on Robotics and Automation*, pages 2356–2357. IEEE Computer Society, 1991.

[10] Guaraci Bastos Jr and Olivier Brüls. An integrated control-structure design for manipulators with flexible links. *IFAC-PapersOnLine*, 48(11):156–161, 2015.

[11] Hong-Sen Yan and Guo-Jhih Yan. Integrated control and mechanism design for the variable input-speed servo four-bar linkages. *Mechatronics*, 19(2):274–285, 2009.

[12] WJ Zhang, Q Li, and LS Guo. Integrated design of mechanical structure and control algorithm for a programmable four-bar linkage. *IEEE/ASME transactions on mechatronics*, 4(4):354–362, 1999.

[13] J-H Park and Haruhiko Asada. Integrated structure/control design of a two-link nonrigid robot arm for high-speed positioning. In *Proceedings 1992 IEEE International Conference on Robotics and Automation*, pages 735–736. IEEE Computer Society, 1992.

[14] Ricard Sanchís, Salvador Cardona, and Jordi Martínez. Determination of the vertical vibration of a ballasted railway track to be used in the experimental detection of wheel flats in metropolitan railways. *Journal of Vibration and Acoustics*, 141(2), 2019.

[15] Tianjian Chen, Zhanpeng He, and Matei Ciocarlie. Co-designing hardware and control for robot hands. *Science Robotics*, 6(54):eabg2133, 2021.

[16] Lucy Jackson, Celyn Walters, Steve Eckersley, Pete Senior, and Simon Hadfield. Orchid: Optimisation of robotic control and hardware in design using reinforcement learning. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4911–4917. IEEE, 2021.

[17] Vinay Gupta, Subir Kumar Saha, and Himanshu Chaudhary. Optimum design of serial robots. *Journal of Mechanical Design*, 141(8), 2019.

[18] James T Allison. Plant-limited co-design of an energy-efficient counterbalanced robotic manipulator. *Journal of Mechanical Design*, 135(10):101003, 2013.

[19] Marc Toussaint, Jung-Su Ha, and Ozgur S Oguz. Co-optimizing robot, environment, and tool design via joint manipulation planning. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6600–6606. IEEE, 2021.

[20] Ostyn, Frederik. A clutch, industrial robot and the use thereof, 2022.

[21] Andreas Aristidou, Joan Lasenby, Yiorgos Chrysanthou, and Ariel Shamir. Inverse kinematics techniques in computer graphics: A survey. In *Computer graphics forum*, number 6, pages 35–58. Wiley Online Library, 2018.

[22] John Luh, Michael Walker, and Richard Paul. On-line computational scheme for mechanical manipulators. 1980.

[23] Alessandro De Luca and Lorenzo Ferrajoli. A modified newton-euler method for dynamic computations in robot fault detection and control. In *2009 IEEE International Conference on Robotics and Automation*, pages 3359–3364. IEEE, 2009.

[24] Diederik Verscheure, Bram Demeulenaere, Jan Swevers, Joris De Schutter, and Moritz Diehl. Time-optimal path tracking for robots: A convex optimization approach. *IEEE Transactions on Automatic Control*, 54(10):2318–2327, 2009.

[25] Florin Stoical, Vlad-Mihai Ivănuşcă, Ionela Prodan, and Dan Popescu. Obstacle avoidance via b-spline parametrizations of flat trajectories. In *2016 24th Mediterranean Conference on Control and Automation (MED)*, pages 1002–1007. IEEE, 2016.

[26] François Bachoc, Céline Helbert, and Victor Picheny. Gaussian process optimization with failures: classification and convergence proof. *Journal of Global Optimization*, 78(3):483–506, 2020.