Singularity-Aware Design Optimization for Multi-Degree-of-Freedom Spatial Linkages

Guirec Maloisel, Espen Knoop, Bernhard Thomaszewski, Moritz Bächer, and Stelian Coros

Abstract—**We introduce a singularity-aware design optimization method for spatial multi-degree-of-freedom mechanical linkages. At the core of our approach is an adversarial sampling strategy, which actively detects singular configurations within the targeted operation range. The detection of singularities in both forward and inverse kinematics allows for two-way bijective mappings between input and output trajectories on our optimized designs, thus enabling robust control. We demonstrate our approach on a set of simulation examples and provide additional validation on physical prototypes.**

Index Terms—**Mechanism design, kinematics, optimization and optimal control.**

I. Introduction

PARALLEL robots and kinematic linkages are ubiquitous in robotics, used in robot hands, grippers, and parallel manipulators. They have been well-studied in the literature, with the seminal work by Merlet [1] providing a comprehensive overview. However, as detailed in Chap. 11 of [1], the design of such linkages remains a highly challenging task despite having received significant attention; in particular for non-planar mechanisms with multiple degrees of freedom.

In the wake of the pioneering work of Burmester and Freudenstein on four-bar linkages [2], [3], optimal design problems were systematically studied for several families of parallel mechanisms, such as planar manipulators [4], [5], spherical manipulators [6], Delta robots [7]–[9] or 6-Degree-Of-Freedom (6-DOF) platforms [10]–[13]. While notable effort has been made towards a unified methodology for such problems [7], [14], [15], manual derivations and tailoring to each type of mechanism are still largely required.

In contrast, the automatic design of arbitrary *single*-DOF linkages has recently received special focus [16]–[21]. While fewer assumptions are made on the precise type of mechanism, the restriction in mobility can be found very limiting for more general mechanisms. Extending these methods is however nontrivial, as the *multi*-DOF case introduces an additional set of challenges.

For the single-DOF case, a common design task is to provide a target trajectory that an end-effector should track. Dense sampling can then be used to ensure sufficient motion approximation and absence of singularities in the operating range. From a computational point of view, this approach quickly becomes impractical with a multi-dimensional workspace. Furthermore, the coupling between actuators along with geometric nonlinearities in the mechanism typically result in highly complex configuration spaces. Feasible regions in this space often enclose infeasible ones, and singular configurations divide the space into several kinematic branches with arbitrarily complex boundaries. The navigation of this space is often challenging, requiring singularity-aware path-planning [22]–[25]; see also [26] and [27].

Several methods exist for determining the locations and types of singularities [28] as well as the different branches [29], [30] of a given mechanism. However, none of them are readily applicable for design optimization where the configuration space topology changes as a function of the design parameters.

In this work, we develop the theoretical and algorithmic basis for optimization-driven design of spatial multi-motor mechanisms. To this end, we devise a singularity-aware inverse kinematics algorithm that promotes reachability of configurations on the boundary of the target region. We augment this formulation with a new adversarial sampling strategy that actively tracks down forward and backward singularities [31] in the operating range. The overarching goal of this approach is to generate optimized designs with two-way monotonic mappings between motor values and output variables around all configurations in their target range. These properties, in turn, guarantee that any trajectory within the target range is feasible and singularity-free, thus eliminating the need for specialized path planning.

II. Overview and Background

The input to our method is a spatial mechanical linkage consisting of rigid components whose relative motion is constrained by a set of passive joints and actuators (see Fig. 1 input). While this input is assumed to be kinematically feasible, it cannot reach

Manuscript received February 24, 2021; accepted June 17, 2021. Date of publication July 7, 2021; date of current version July 19, 2021. This letter was recommended for publication by Associate Editor M. Arsenault and Editor C. Gosselin upon evaluation of the reviewers' comments. This work was supported in part by the European Research Council (ERC) under the European Union's Horizon 2020 Research and Innovation Programme under Grant agreement No. 866480. *(Corresponding author: Guirec Maloisel.)*

Guirec Maloisel is with Disney Research, 8006 Zurich, Switzerland, and also with the Department of Computer Science, ETH Zurich, 8006 Zurich, Switzerland (e-mail: guirec.maloisel@disneyresearch.com).

Espen Knoop and Moritz Bächer are with Disney Research, 8006 Zurich, Switzerland (e-mail: espen.knoop@gmail.com; moritz.baecher@disneyresearch.com).

Bernhard Thomaszewski and Stelian Coros are with the Department of Computer Science, ETH Zurich, 8006 Zurich, Switzerland (e-mail: b.thomaszewski@gmail.com; scoros@inf.ethz.ch).

This letter has supplementary downloadable material available at https://doi.org/10.1109/LRA.2021.3095043, provided by the authors.

Digital Object Identifier 10.1109/LRA.2021.3095043

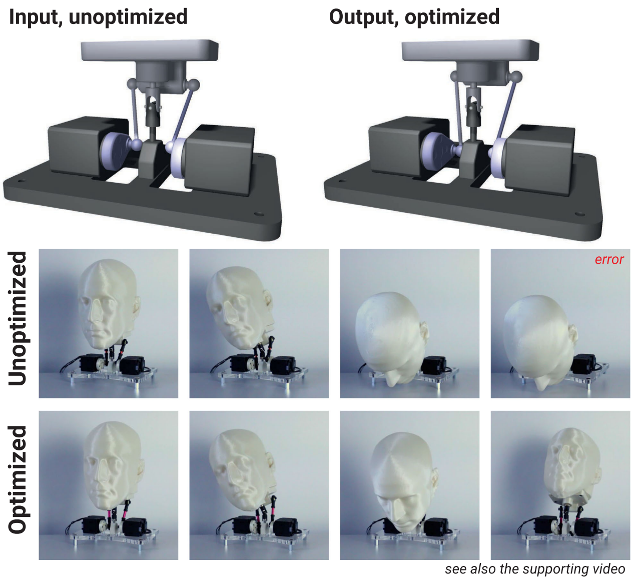


Fig. 1. Our singularity-aware design optimization pipeline takes as input a mechanical linkage and optimizes joint locations such that a target workspace is reachable and free from singularities. The unoptimized physical neck assembly fails due to a forward singularity, while the optimized assembly functions as desired. Note how subtle differences in the mechanism geometry yield drastically different behavior, stressing the need for a computational method.

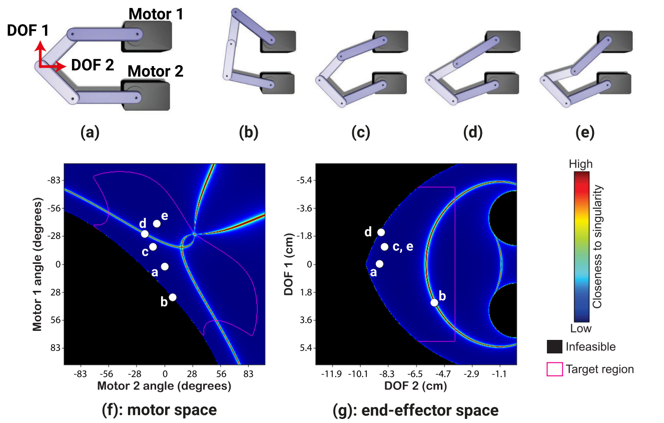


Fig. 2. **Singularities.** A five-bar mechanism, shown in its rest pose in (a), can exhibit both forward (b) and backward (d) singularities. In (f) and (g), we visualize the proximity to singularities (f_{sing}) in both motor space and end-effector space, with poses (a)-(e) as indicated. Forward singularities (b) lie at the limit of feasible configurations in motor space, and yield unbounded motor torques. Backward singularities (d) lie at the limit of the feasible region in end-effector space, and cause non-bijectivity in the input to output mapping. See indeed how states (c) and (e) are on opposite sides of singularity (d), i.e., they have the same end-effector position but different motor values. Additionally, in pose (d), the mechanism is unable to produce a non-zero force at the end effector in the direction of the aligned links.

all points in a target task space and may contain singularities in this space with respect to backward and forward kinematics. Our goal is to change the design of the linkage to ensure that all points within the task space can be reached with finite motor torques, and that the end-effector can apply forces in *all* directions at all points within this output space. Our method is agnostic to how the task-space is defined—positions, angles, or combinations thereof—as illustrated by our examples.

See Fig. 2 for an illustration of the different types of singularities, and how they introduce non-bijectivity in the mapping from motor space to end-effector space trajectories. We will show similar plots throughout the paper, visualizing proximity to a singular configuration using a color map: the brighter the color, the closer we are to a singular configuration. Black regions represent infeasible configurations, and contours shown in magenta delineate feasible parts of the target range in end-effector space (g), together with its projection onto motor space (f).

A. Forward Kinematics

To represent a linkage, we rely on a maximal coordinate formulation where the position and orientation of each component is represented with a 3D point and a unit-length quaternion, collected in a 7-coordinate state vector $\mathbf{s}_i$. For a linkage with n components, the degrees of freedom are collected in a $7n$-vector $\mathbf{s}$.

Passive joints are represented with a set of constraints $\boldsymbol{\mathcal{C}}_k(\mathbf{s}_i, \mathbf{s}_j)$ $(= 0$ if satisfied), restricting the relative motion of two connected components i and j [17], [32]. For example, for a revolute joint, we formulate five constraints: three to ensure that the joint always remains at the same location on either component, and two to restrict rotations about two of the three axes of rotation.

For active joints or motors, we have additional actuation parameters, fully restricting the relative motion between pairs of components once the parameter is set. Representing the actuation parameters with a vector $\mathbf{a}_k$, we can formulate constraints of the form $\boldsymbol{\mathcal{C}}_k(\mathbf{a}_k, \mathbf{s}_i, \mathbf{s}_j)$. For example, for a revolute actuator, $\mathbf{a}_k$ has a single entry, representing the position of the actuator about the revolute axis. The set of actuator values, collected in vector $\mathbf{a}$, span the motor space in Fig. 2.

Collecting all constraints in a vector $\boldsymbol{\mathcal{C}}_{\text{FK}}(\mathbf{a}, \mathbf{s})$, we can then solve for the state of the linkage for given actuator values $\mathbf{a}$, by minimizing the nonlinear least squares problem

$$\min_{\mathbf{s}} f_{\text{FK}}(\mathbf{a}, \mathbf{s}) \quad \text{with} \quad f_{\text{FK}}(\mathbf{a}, \mathbf{s}) = \frac{1}{2} \boldsymbol{\mathcal{C}}_{\text{FK}}^T \boldsymbol{\mathcal{C}}_{\text{FK}} \qquad (1)$$

with a standard Levenberg-Marquardt implementation [33], outputting failure if a non-zero local minimum is obtained.

To guarantee that the mechanism stays predictable, and controllable with finite actuator torques, one must keep a distance from forward singularities, by ensuring that the constraint Jacobian $\frac{\partial \boldsymbol{\mathcal{C}}_{\text{FK}}}{\partial \mathbf{s}}$ always has full rank.

B. Backward Kinematics

To evaluate reachability objectives, we introduce backward kinematics variables $\mathbf{b}$, which correspond to a non-redundant parameterization of the mechanism's output, and span the end-effector space in Fig. 2. Each variable $\mathbf{b}_k$ is either a Cartesian coordinate of a point attached to component i, or an angle at a joint connecting components i and j. For example, for the five-bar linkage in Fig. 2, it is natural to specify a Cartesian target range for the end-effector's position. We then define backward kinematics constraints as follows: we replace all active joint constraints in $\boldsymbol{\mathcal{C}}_{\text{FK}}$ with their corresponding passive joints [32]. We then add the constraint $\boldsymbol{\mathcal{C}}_k(\mathbf{b}_k, \mathbf{s}_i)$ or $\boldsymbol{\mathcal{C}}_k(\mathbf{b}_k, \mathbf{s}_i, \mathbf{s}_j)$ for positional or angular variables $\mathbf{b}_k$.[1]

The collected constraints $\boldsymbol{\mathcal{C}}_{\text{BK}}(\mathbf{b}, \mathbf{s})$ evaluate to zero when the target point $\mathbf{b}$ in end-effector space is reached, and can be plugged in the same algorithm as above to solve backward kinematics. In the physical system, this corresponds to manually driving the mechanism using its end-effector, forcing the motors to follow. The actuator values $\mathbf{a}(\mathbf{s})$ to reach the same configuration in the forward direction can be read off the output state, enabling, in the absence of singularities, the control of the mechanism directly from its end-effectors for the same cost as solving forward kinematics. Additionally, for a state satisfying forward kinematics constraints, but not backward constraints, the residual value of the objective $f_{\text{BK}} = 1/2\boldsymbol{\mathcal{C}}_{\text{BK}}^T\boldsymbol{\mathcal{C}}_{\text{BK}}$ measures the distance to the target $\mathbf{b}$.

To guarantee that an end-effector position is mapped uniquely to a motor configuration, thus enabling robust control, and to ensure that forces can be applied in arbitrary directions at the end-effector, one must stay a safe distance away from backward singularities, by controlling that the corresponding Jacobian $\frac{\partial \boldsymbol{\mathcal{C}}_{\text{BK}}}{\partial \mathbf{s}}$ has full rank for all traversed states.

C. Design Optimization

We take as input a desired range for all backward kinematics parameters $\mathbf{b}$. The mechanism is parameterized by the position and orientation of all joints in the initial configuration of the linkage. A reduced set of parameters taking into account design constraints (fixed joints, symmetries, etc.) is collected in a vector $\mathbf{p}$. We then ask for parameters that lead to a design where (1) all points in a target range can be reached, (2) there are no singularities in the target range where motor torques would go to infinity, and (3) the mapping from end-effector or task space parameters to actuator values is one-to-one around all points such that forces in arbitrary directions can be applied at the end-effector. To evaluate the latter two requirements, we next discuss a metric that measures the distance to either kind of singularity.

III. Measuring Proximity to Forward and Backward Singularities

To detect and ultimately avoid singular configurations, we can study sensitivities of the state with respect to either forward or backward kinematic parameters. Upon successful solution of the forward kinematics problem, we obtain a state $\mathbf{s}$ at which the gradient of objective f_{FK} is zero. The sensitivity of the state with respect to the actuation parameters $\mathbf{a}$ is given by the implicit function theorem

$$\frac{\mathrm{d}\mathbf{s}}{\mathrm{d}\mathbf{a}} = -\left(\frac{\partial^2 f_{\text{FK}}}{\partial \mathbf{s}^2}\right)^{-1} \frac{\partial^2 f_{\text{FK}}}{\partial \mathbf{a} \partial \mathbf{s}}. \tag{2}$$

As long as the constraint Jacobian is not rank deficient[2], the mapping from $\mathbf{a}$ to $\mathbf{s}$ is locally well-defined, and small changes to actuator values lead to changes in states bounded in magnitude. Forward singularities correspond to degeneracies in this mapping and to unbounded sensitivities. Analogously, we can compute the sensitivity of states with respect to end-effector parameters, $\frac{\mathrm{d}\mathbf{s}}{\mathrm{d}\mathbf{b}}$, using the backward kinematics objective.

Both uni-directional mappings are thus well-defined, hence bijective, around non-singular configurations. Provided singularities are avoided, these local mappings can be stitched together to ensure a non-ambiguous translation of any motor-space *trajectory* into the corresponding end-effector motion, and vice versa. [3] In particular, complex path-planning considering multiple kinematics solutions is unnecessary.

To measure proximity to either forward or backward singularities, a candidate objective sums up the inverse of the two smallest singular values of the constraint Jacobians

$$f_{\text{sing}}(\mathbf{a}, \mathbf{b}, \mathbf{s}) = \frac{1}{\sigma_{\min}\left(\frac{\partial \boldsymbol{\mathcal{C}}_{\text{FK}}}{\partial \mathbf{s}}\right)} + \frac{1}{\sigma_{\min}\left(\frac{\partial \boldsymbol{\mathcal{C}}_{\text{BK}}}{\partial \mathbf{s}}\right)}. \tag{3}$$

As we only consider states satisfying the constraints, values for $\mathbf{a}, \mathbf{b}$ can be read off the state $\mathbf{s}$, and we write $f_{\text{sing}}(\mathbf{s})$.

As noted in previous work, e.g. in [10], quantities with inconsistent units (distances vs. angles) appear in kinematic constraints and variables, rendering f_{sing} physically meaningless, unless the state $\mathbf{s}$ and constraints $\boldsymbol{\mathcal{C}}_{\text{FK}}$ and $\boldsymbol{\mathcal{C}}_{\text{BK}}$ are homogenized. To do so, we scale constraints and variables that measure distances by the inverse of a reference length, set to the diameter of the initial mechanism (i.e., the largest distance between any pair of points on the mechanism).[4]

IV. Evaluating the Performance of a Design

Given a design with fixed parameters, we must evaluate the linkage with respect to our singularity-free reachability target. To this end, we introduce a sparse sampling strategy, which selects relevant locations at which to evaluate (a) f_{BK}, to control the reachability of the target range, defining *IK samples*, and (b) f_{sing}, to enforce singularity-free motion, defining *adversarial samples*.

A. Evaluating Reachability

To evaluate if all desired locations can be reached by a given end-effector, we sparsely sample the boundary of the range in end-effector space, resulting in a set of target points $\mathbf{b}^t$. We then solve for the actuation parameters that minimize the distance to these samples while staying a safe distance from singular configurations and fulfilling our forward kinematics objective to

[1] In practice, these additional constraints are implemented by measuring the value $\mathbf{b}_k(\mathbf{s})$ on the current state, and evaluating its signed distance to the target value, taking into account modulo 2π values for angles.

[2] If $f_{\text{FK}} = 0$, the Hessian $\frac{\partial^2 f_{\text{FK}}}{\partial \mathbf{s}^2}$ rewrites $(\frac{\partial \boldsymbol{\mathcal{C}}_{\text{FK}}}{\partial \mathbf{s}})^T \frac{\partial \boldsymbol{\mathcal{C}}_{\text{FK}}}{\partial \mathbf{s}}$, so rank deficiency of the Hessian can be studied directly on the constraint Jacobian.

[3] Note that the *global* map from motor to end-effector values can be non-bijective even in singularity-free regions: there can be singularity-free paths connecting distinct kinematics solutions, see e.g. [26]. This is however not a problem for non-ambiguous control as long as the mapping between *trajectories* in these two spaces is bijective.

[4] The chosen reference length was validated by plotting the obtained profiles for f_{sing}, and proved sufficient for all our examples. A more detailed study might be needed for corner cases.

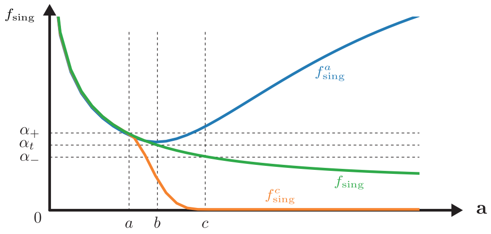


Fig. 3. Comparative plot of f_{sing}, f^c_{sing}, f^a_{sing} over a segment of motor space for the initial design of the finger assembly, featuring a singularity at the origin. For $f_{\text{sing}} > \alpha_+$ (left of configuration a), the three metrics coincide. For $f_{\text{sing}} < \alpha_-$ (right of configuration c), f^c_{sing} is zero. Configuration b corresponds to a local minima in f^a_{sing} to which adversarial samples will converge, attained when $f_{\text{sing}} = \alpha_t$. By picking $\alpha_t > \alpha_-$, we ensure it falls within the active domain of f^c_{sing}.

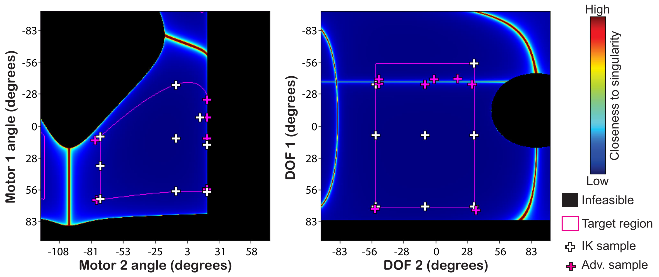


Fig. 4. Result of IK (white markers) and adversarial sampling (magenta markers) on the initial design of our finger mechanism example, visualized in motor (left) and end-effector (right) space. Note how one IK sample crossed a forward singularity with a too big step during minimization of Eq. 4. Without adversarial sampling, and if the same happened to the two neighbouring reachability samples, inverse kinematics would not suffice to detect the deficiency of this design.

first-order optimality, using a custom *Inverse Kinematics* (IK) formulation

$$\min_{\mathbf{a}} f_{\text{BK}}(\mathbf{b}^t, \mathbf{s}) + f^c_{\text{sing}}(\mathbf{s}) \text{ s.t. } \frac{\partial f_{\text{FK}}}{\partial \mathbf{s}}(\mathbf{a}, \mathbf{s}) = \mathbf{0}. \tag{4}$$

The *culled* singularity metric f^c_{sing} equals f_{sing} for values above a user-specified threshold α_+, smoothly transitioning to zero at a cut-off threshold α_- (see Fig. 3; Appendix A) to avoid perturbing IK results away from singularities.

To numerically solve this problem, we make use of the implicit function theorem (Eq. 2) to compute the analytical gradient of the objective with implicit dependence $\mathbf{s}(\mathbf{a})$, and finite differences for the singularity objective. For minimization, we use a standard quasi-Newton scheme [33], with the starting point set to the initial state of the mechanism. The result is a set of IK samples $\mathbf{a}^t_{\text{IK}}$. The residual distance to the targets $\mathbf{b}^t$ indicates the overall coverage of the target range by the current design.

Occasionally, a too large step in minimization may lead to the crossing of a forward singularity despite the barrier term f^c_{sing}. This is resolved by the use of adversarial samples to detect the crossed singularity, as explained next.

B. Detecting Singularities With Adversarial Sampling

Starting from every IK sample $\mathbf{a}^t_{\text{IK}}$ in end-effector space, we identify a potential singularity in a local neighborhood by solving

$$\min_{\mathbf{a}} f^a_{\text{sing}}(\mathbf{s}) + R_1(\mathbf{a}) + R_2(\mathbf{s}) \quad \text{s.t.} \quad \frac{\partial f_{\text{FK}}}{\partial \mathbf{s}}(\mathbf{a}, \mathbf{s}) = \mathbf{0}. \tag{5}$$

The *attractive* singularity metric f^a_{sing} is a warped version of f_{sing} that attracts adversarial samples towards singularities, but is repulsive at short distances (see Fig. 3; Appendix A). To keep the search space to a local neighborhood, we add a first regularizer $R_1 = \frac{1}{2}\|\mathbf{a} - \mathbf{a}^t_{\text{IK}}\|^2$ that keeps the adversarial sample close to $\mathbf{a}^t_{\text{IK}}$. A second regularizer R_2 keeps the corresponding end-effector point $\mathbf{b}(\mathbf{s})$ within the targeted end-effector space (see Appendix B). The relative importance of the two regularizers is controlled with weights set by the user. The rationale behind introducing R_1 is to prevent adversarial samples from travelling to separate regions of motor space whose projection in end-effector space are also in the target range, thus undetected by R_2. See for instance configuration (e) in Fig. 2.

This results in a coarse set of adversarial samples $\mathbf{a}^t_{\text{adv}}$. A non-zero value of f^c_{sing} at these locations indicates the existence of a singularity in the range. To reduce the likelihood of missing singularities, we optionally add samples $\mathbf{b}^t$ in the interior of the end-effector space. 3-4 samples in total per dimension have proven sufficient for effective design optimization as we demonstrate with several examples in our Results section.

Fig. 4 illustrates a result of IK + adversarial sampling, in motor space and end-effector space.

C. Evaluating Overall Performance

To evaluate the performance of a design, we then add together the reachability performance and the proximity to singular configurations within the target range

$$f_{\text{design}} = \frac{1}{\#t} \sum_t f_{\text{BK}}(\mathbf{b}^t, \mathbf{s}^t_{\text{IK}}) + f^c_{\text{sing}}(\mathbf{s}^t_{\text{adv}}), \tag{6}$$

where $\mathbf{s}^t_{\text{IK}}, \mathbf{s}^t_{\text{adv}}$ are the states corresponding to $\mathbf{a}^t_{\text{IK}}, \mathbf{a}^t_{\text{adv}}$.

If this objective takes on a value of zero, the design can reach the user-specified end-effector range without getting trapped in a singularity, uniquely mapping actuator parameters to end-effector positions along all trajectories within the range.

V. Energy Minimization

The core part of our design optimization is concerned with minimizing f_{design} over the space spanned by the user-selected design parameters $\mathbf{p}$. To encourage small deviation from the initial design $\mathbf{p}_0$, we add a standard regularizer of the form $\frac{1}{2}\|\mathbf{p} - \mathbf{p}_0\|^2$. We solve this problem with a stochastic optimization method, since several aspects of the problem prevent the use of classical gradient-based methods.

More specifically, derivative computations are made unstable by the sparsity of the sampling scheme: for similar values for the design parameters $\mathbf{p}$, the samples could converge to different local minima. Some exploratory profile plots of f_{design} along components of $\mathbf{p}$ revealed almost flat portions with sudden jumps at their boundary, corresponding to a topology change in motor or end-effector space. Furthermore, analytical derivatives are impractical to compute due to the chain of implicit dependencies between f_{design}, adversarial sampling, IK sampling and FK; and finite differences are expensive to evaluate due to the multiple sub-problems that need to be solved.

Based on these insights, we implemented a first algorithm, which we refer to as *partial gradient descent:* we evaluate the gradient of f_{design} with frozen sample locations $\mathbf{a}^t_{\text{IK}}, \mathbf{a}^t_{\text{adv}}$, thus removing the problematic partial derivatives. The samples are then recomputed after a (partial) gradient descent step. This algorithm can solve simple problem instances, but has no provable convergence and fails in more complicated cases.

Encouraged by the promising results of stochastic optimization for mechanism design in existing work [34]–[36], we chose to rely on *Particle Swarm Optimization* (PSO) [37], which requires no derivatives and is known for its robustness towards local minima. To leverage the partial information still available on the derivatives of f_{design}, and to improve on vanilla PSO, we use a modified scheme: for each particle and with 50% probability, we take either a regular PSO step, or a partial gradient descent step as described above. This allows us to tweak the probability distribution towards the exploration of locally interesting directions, while leaving room for random exploration.

On our examples, PSO optimization was run with 20 particles initialized randomly around the initial design. The learning rate was set to 1, both acceleration coefficients to 2, and the inertia coefficient was initialized to 0.9, decreasing linearly until reaching 0.4 at a hundred iterations. The algorithm is stopped after a satisfactory design is found (with $f_{\text{design}} \simeq 0$ and a sufficiently low regularizer value), and additional iterations do not lead to further improvements.

VI. Results

We demonstrate in this section the efficacy of our method on several examples. We refer the reader to the accompanying video for supplementary visual comparison of initial vs. optimized designs, together with sample motions.

A. Five-Bar Linkage

As a first example, we optimize a five-bar robot to be able to reach any point inside a rectangular box. The initial and optimized designs are shown in Fig. 5. During the optimization, we ask for the design to remain symmetric.

As can be seen in the plots in Fig. 2, the initial design suffers from a forward singularity crossing its range. In addition, only part of the range is within reach. Both problems are resolved in the optimized result, see Fig. 6.

B. Finger Assembly

In this example, we optimize a 2-DOF mechanism that drives the metacarpophalangeal (MCP) joint of a finger, with the goal of increasing the feasible range of motion. See Fig. 7 for the initial and optimized designs. As can be seen in Fig. 8, the

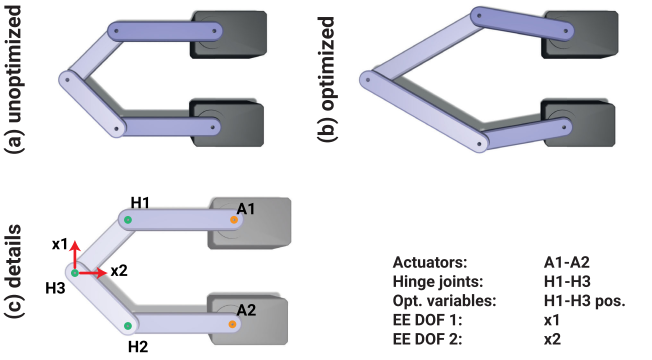


Fig. 5. Five-bar linkage.

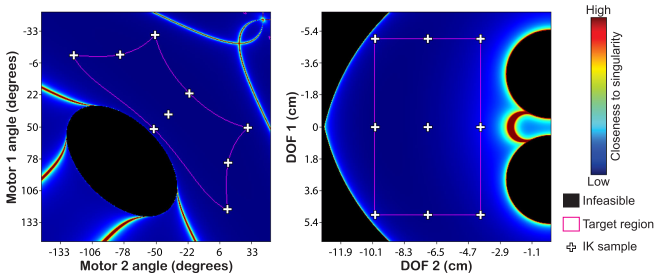


Fig. 6. Motor space (left) and end-effector space (right) plots for the optimized five-bar linkage example. Comparison with the equivalent plots in Fig. 2 reveals a correction of the initial defects (too small workspace, forward singularity). The clearer shape correspondence between the target region as seen from the perspective of the motor or end-effector illustrates a two-way well-defined mapping.

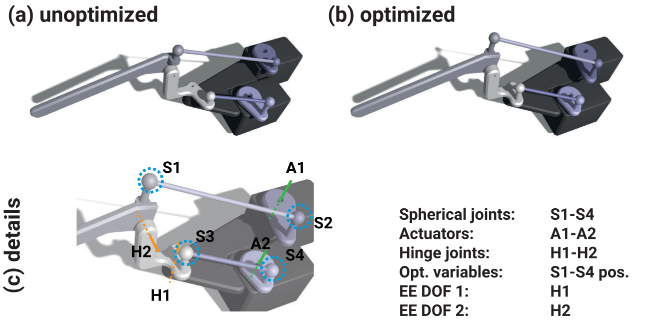


Fig. 7. Finger.

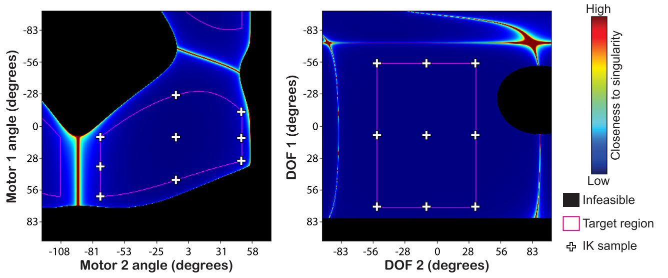
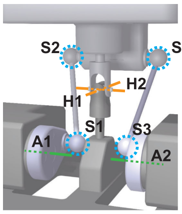


Fig. 8. Motor space (left) and end-effector space (right) plots for the optimized finger assembly example. Comparison with the equivalent plots in Fig. 4 show that the forward singularity limiting the initial motion has been pushed out of the target range.



Spherical joints:	S1-S4
Actuators:	A1-A2
Hinge joints:	H1-H2 (univ joint)
Opt. variables:	S1-S4 pos.
EE DOF 1:	H1
EE DOF 2:	H2

Fig. 9. Two-motor neck details.

optimized design pushed away the singularity limiting the range of motion of the initial design. A higher regularizer weight was used here than for other examples due to the spatial nature of the assembly, in order to keep the assembly compact. In the supplementary video, we show the initial and optimized design executing a motion profile. To control the initial design, we used our singularity-aware IK that comes as close as possible to the desired target motion while not passing through singularities.

C. Two-Motor Neck Assembly

We next considered a 2-DOF neck assembly, capable of pitch and roll motions. The initial design suffers from a forward singularity when tilting forward, as illustrated in Fig. 1. Getting too close to this causes catastrophic and unrecoverable failure. See also Fig. 9 for details. During the optimization, we asked the design to remain symmetric. To illustrate a good correspondence between our simulations and reality, we fabricated the initial and final designs and ran motion sequences, as shown in the supporting video. Again, we used a singularity-aware IK for the initial design.

D. Two-Motor Jansen Leg

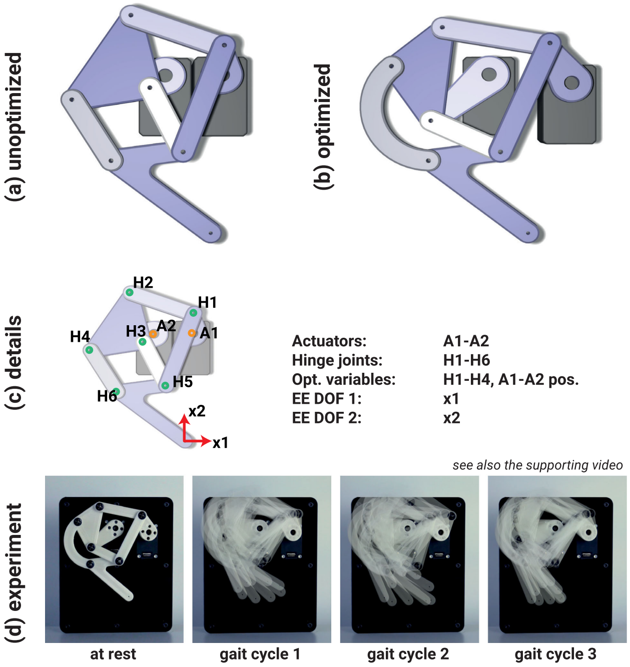
The Jansen linkage is a well-known example of a 1-DOF leg mechanism with a distinctive visual appeal, that performs a fixed walking cycle. With the addition of a second actuator, the mechanism should be able to perform arbitrary gait cycles—essential for a walking robot—however due to the complex kinematics it is non-trivial to make this design change.

In this example, we start from the 1-DOF Jansen leg, and naively place a second motor by moving the grounded joint in the 1-DOF version onto a crank. See Fig. 10 and also the supporting video. We define a rectangular target region for the end-effector. The initial design suffers from forward and backward singularities in the target region, which the optimization is able to remove.

To demonstrate arbitrary leg motions, we execute three different gait cycles. We also created a physical version of the optimized leg, to validate our modeling.

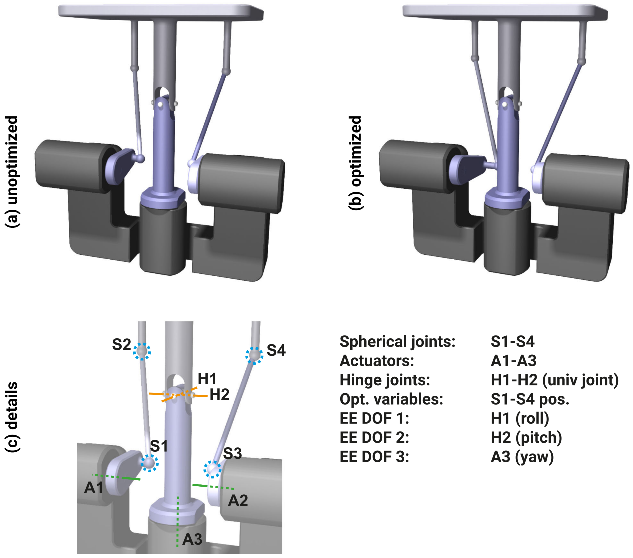
E. Three-Motor Neck Assembly

To evaluate the scalability and generalization capabilities of our algorithm, we augmented the neck mechanism presented above with an additional DOF, controlling the yaw angle (see Fig. 11). Our optimization pipeline worked as expected, and corrected the singularities and range limitations of the initial design. We can therefore expect that our method will generalize well to higher dimensions.



Actuators:	A1-A2
Hinge joints:	H1-H6
Opt. variables:	H1-H4, A1-A2 pos.
EE DOF 1:	x1
EE DOF 2:	x2

Fig. 10. Two-motor Jansen leg.



Spherical joints:	S1-S4
Actuators:	A1-A3
Hinge joints:	H1-H2 (univ joint)
Opt. variables:	S1-S4 pos.
EE DOF 1:	H1 (roll)
EE DOF 2:	H2 (pitch)
EE DOF 3:	A3 (yaw)

Fig. 11. Three-motor neck, details.

To examine the optimization results, we adapted our 2D visualization method to display a 2D section of the 3D space. Please see the supplemental video.[5] It can be seen that the optimization successfully increases the feasible range of the mechanism to match the specified target.

[5] In the visualization, some discontinuities between successive section views can be observed. This does not affect the optimization result, as this is an artefact of the visualization only, which arises from the overlapping projections of complex topologies.

TABLE I
PERFORMANCE STATISTICS

	$\lvert\mathbf{s}\rvert$	$\lvert\mathbf{p}\rvert$	#targets	#opt. iter.	Opt. time
Five-bar	35	3	9	8	15min
Finger	49	12	9	21	1h 05min
Neck 2 DOF	42	6	9	25	55min
Jansen leg	63	12	16	38	13h
Neck 3 DOF	56	6	27	52	18h 30min

[5] $\lvert\mathbf{s}\rvert$: number of FK variables; $\lvert\mathbf{p}\rvert$: number of design parameters, with all optimizer constraints taken into account; *#targets:* number of IK targets, and thus of IK and adversarial samples; *# opt. iter., opt. time:* number of optimization iterations and approximate optimization time

F. Performance

All optimizations were performed on a machine with an Intel Core i7-7700 processor (4 cores, 4.2 GHz) and 32 GB of RAM. Our pipeline was implemented in C++, using Eigen for linear algebra and a custom symbolic differentiation library to compute derivatives. Tab. I reports the key performance statistics for all the examples. Straightforward parallelism at the IK / adversarial sample level was used to accelerate the computations, but there are other unexplored opportunities for parallelization at the PSO particle level. We believe that the performance could also be significantly improved by an early detection of “bad” designs, for which sample computations tend to be numerically more challenging.

VII. CONCLUSION

We first summarize key contributions of our method.

- **Problem formulation.** Using an end-effector parameterization allows for an explicit formulation of the desired mechanism range. Furthermore, the requirement to avoid both forward kinematic and backward kinematic singularities allows for all trajectories within the range to be feasible on optimized designs.
- **Scalability.** Only a sparse set of configurations is used to summarize the assembly behavior in its operation range, as opposed to previous methods where target trajectories were densely sampled. In parameter space, the use of PSO supplants grid search approaches, impractical for large numbers of design variables.
- **Robustness.** The combination of IK and adversarial samples allows for dealing with both types of singularities, regardless of the partial reliability of IK. The use of a modified PSO scheme lets us consistently find relevant solutions to the optimization problem, despite the impracticality of gradient computation.
- **Generality.** Our method improves over previous work through the handling of multi-motor designs. Furthermore, it is agnostic to the type of joints in the mechanism or to the definition of the end-effector variables $\mathbf{b}$, as long as the associated constraint functions are provided. Our 3-DOF neck example is an indication that the method will also generalize well to mechanisms with higher DOF counts.

Our approach, however, has several limitations, providing interesting avenues for future work.

Although our adversarial sampling strategy proved successful for a reasonably low number of samples on all our examples, we cannot give a mathematical guarantee that all singularities are detected this way. Deriving bounds on the detection accuracy vs. the number of samples would thus be beneficial.

While our optimization method can be expected to generalize well to higher number of DOFs, an exciting direction for future work is the visualization of the performance of a mechanism for higher-DOF (> 3) results. In particular, the non-bijectivity of the mapping from motor space to end effector space can be expected to become increasingly prominent in higher dimensions.

In our demonstrations, generating the geometry of the optimized components so as to avoid collisions was trivial on planar examples, and relatively simple on spatial examples. Nonetheless, resolving collisions will likely become a bottleneck for complex spatial mechanisms. We believe our algorithm could be augmented seamlessly with a collision avoidance objective, analogous to the singularity proximity metric. The main challenge would be to make this objective fast to evaluate while smooth and non-zero even at a significant distance from the closest collision event.

Another interesting research direction would be a systematic analysis of the effects of errors in fabrication, motor control and joint backlash, as done in [38] on a parallel manipulator. As we have seen, avoiding singular configurations avoids the worst-case sensitivity to a change in the kinematic parameters; a similar argument could be made for changes in design parameters. However, the explicit optimization of an arbitrary assembly design to increase its robustness to errors in the absence of singular configurations has yet to be investigated.

Finally, continuous design optimization is only one half of the more general mechanism synthesis problem. Being able to automatically select and connect components so as to generate an initial linkage design, given a higher-level motion input, would be an exciting area of future research.

APPENDIX

A. Design of the Culled and Attractive Versions of f_{sing}

The culled singularity proximity metric f^c_{sing} is given by $f^c_{\text{sing}}(\mathbf{s}) = f_{\text{sing}}(\mathbf{s})\varphi(f_{\text{sing}}(\mathbf{s}))$, with φ a C^2 step function:

$$\varphi(x) = \begin{cases} 0 & \text{for } x < \alpha_- \\ 6X^5 - 15X^4 + 10X^3 & \text{for } \alpha_- \leqslant x < \alpha_+ \\ 1 & \text{for } x \geqslant \alpha_+ \end{cases} \tag{7}$$

where $X = \frac{x-\alpha_-}{\alpha_+-\alpha_-}$. The user-set threshold α_- is the highest accepted value of f_{sing}, fixed in our results between 1.2 and 4 times $f_{\text{sing}}(\mathbf{s}_0)$, depending on the example, and with $\mathbf{s}_0$ the initial state. α_+ is fixed to $\frac{10}{7}\alpha_-$, but its value has little incidence.

For adversarial sampling, the attractive version f^a_{sing} of f_{sing} is defined as $f^a_{\text{sing}}(\mathbf{s}) = \psi(f_{\text{sing}}(\mathbf{s}))$, with

$$\psi(x) = \begin{cases} ax^3 + bx^2 + cx + d & \text{for } x < \alpha_+ \\ x & \text{for } x \geqslant \alpha_+ \end{cases} \tag{8}$$

The coefficients $a = \frac{-1}{3(\alpha_+ - \alpha_t)^2}$, $b = \frac{\alpha_+}{(\alpha_+ - \alpha_t)^2}$, $c = 1 - \frac{\alpha_+^2}{(\alpha_+ - \alpha_t)^2}$, and $d = \frac{\alpha_+^3}{3(\alpha_+ - \alpha_t)^2}$ are computed so that ψ is C^2-continuous and has a local minimum at the target value α_t, defined by $\alpha_t = \frac{\alpha_- + \alpha_+}{2}$.

B Second Regularizer in Adversarial Sampling

We detail here the expression for the regularizer R_2 in adversarial sampling. Assuming a target range given as a cartesian product $[\mathbf{b}_0^{\min}, \mathbf{b}_0^{\max}] \times [\mathbf{b}_1^{\min}, \mathbf{b}_1^{\max}] \times \ldots$ (with $\mathbf{b}_i$ the components of $\mathbf{b}$), we define

$$R_2(\mathbf{s}) = \sum_i \beta_-\left(\mathbf{b}_i(\mathbf{s}), \mathbf{b}_i^{\min}\right) + \beta_+\left(\mathbf{b}_i(\mathbf{s}), \mathbf{b}_i^{\max}\right) \qquad (9)$$

where the inferior and superior smooth barrier functions β_-, β_+ are defined as

$$\begin{cases} \beta_-(x, x_{\min}) = M \exp\left(\log\left(\frac{m}{M}\right) \frac{x - x_{\min}}{\varepsilon}\right) \\ \beta_+(x, x_{\max}) = M \exp\left(\log\left(\frac{m}{M}\right) \frac{x_{\max} - x}{\varepsilon}\right) \end{cases} \qquad (10)$$

The parameters M, m, ε are such that β_-, β_+ take the value m at $x = x_{\min} + \varepsilon$ (resp. $x_{\max} - \varepsilon$) and M at $x = x_{\min}$ (resp. $x_{\max}$); in practice, we use $m = 0.1$, $M = 5$ and $\varepsilon = 0.03(\mathbf{b}_i^{\max} - \mathbf{b}_i^{\min})$.

References

[1] J.-P. Merlet, *Parallel Robots*, vol. 128. Dordrecht, Netherlands: Springer, 2006.

[2] L. Burmester, *Lehrbuch der Kinematik Geometrisch Dargestellt: Die Ebene Bewegung.* Leipzig: Arthur Felix, 1888, no. Bd. 1.

[3] F. Freudenstein, "Design of Four-link Mechanisms," Ph.D. Dissertation, Columbia University, USA, 1954.

[4] M. Arsenault and R. Boudreau, "The synthesis of three-degree-of-freedom planar parallel mechanisms with revolute joints (3-RRR) for an optimal singularity-free workspace," *J. Field Robot.*, vol. 21, pp. 259–274, 2004.

[5] A. Hay and J. Snyman, "Optimal synthesis for a continuous prescribed dexterity interval of a 3-dof parallel planar manipulator for different prescribed output workspaces," *Int. J. Numer. Methods Eng.*, vol. 68, pp. 1–12, Oct. 2006.

[6] S. Leguay-Durand and C. Reboulet, "Optimal design of a redundant spherical parallel manipulator," *Robotica*, vol. 15, no. 4, p. 399–405, 1997.

[7] K. Miller, "Optimal design and modeling of spatial parallel manipulators," *Int. J. Robot. Res.*, vol. 23, no. 2, pp. 127–140, 2004.

[8] G. Liu, Y. Chen, Z. Xie, and X. Geng, "Ga-sqp optimization for the dimensional synthesis of a delta mechanism based haptic device design," *Robot. Comput.-Integr. Manuf.*, vol. 51, pp. 73–84, 2018.

[9] E. Rodriguez, C. Riaño Jaimes, A. Alvares, and R. Bonnard, "Design and dimensional synthesis of a linear delta robot with single legs for additive manufacturing," *J. Braz. Soc. Mech. Sci. Eng.*, vol. 41, no. 11, pp. 1–23, 2019.

[10] O. Ma and J. Angeles, "Optimum architecture design of platform manipulators," in *Proc. 5th Int. Conf. Adv. Robot. 'Robots Unstructured Environ.*, vol. 2, 1991, pp. 1130–1135.

[11] K. H. Pittens and R. P. Podhorodeski, "A family of stewart platforms with optimal dexterity," *J. Robot. Syst.*, vol. 10, no. 4, pp. 463–479, 1993.

[12] B. Monsarrat and C. Gosselin, "Workspace analysis and optimal design of a 3-leg 6-dof parallel platform mechanism," *IEEE Trans. Robot. Automat.*, vol. 19, no. 6, pp. 954–966, Dec. 2003.

[13] C. Eryılmaz, V. E. Omurlu, and Y. Li, "Sqp optimization of 6dof 3x3 upu parallel robotic system for singularity free and maximized reachable workspace," *J. Robot.*, vol. 2019, Jan. 2019, Article ID 3928705.

[14] Y. Lou, D. Zhang, and Z. Li, "Optimal design of a parallel machine based on multiple criteria," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2005, pp. 3219–3224.

[15] Y. Lou, G. Liu, and Z. Li, "Randomized optimal design of parallel manipulators," *IEEE Trans. Automat. Sci. Eng.*, vol. 5, no. 2, pp. 223–233, Apr. 2008.

[16] L. Zhu, W. Xu, J. Snyder, Y. Liu, G. Wang, and B. Guo, "Motion-guided mechanical toy modeling," *ACM Trans. Graph.*, vol. 31, no. 6, pp. 1–10, Nov. 2012.

[17] S. Coros *et al.*, "Computational design of mechanical characters," *ACM Trans. Graph.*, vol. 32, no. 4, pp. 1–12, Jul. 2013.

[18] D. Ceylan, W. Li, N. J. Mitra, M. Agrawala, and M. Pauly, "Designing and fabricating mechanical automata from mocap sequences," *ACM Trans. Graph.*, vol. 32, no. 6, pp. 1–11, 2013.

[19] B. Thomaszewski, S. Coros, D. Gauge, V. Megaro, E. Grinspun, and M. Gross, "Computational design of linkage-based characters," *ACM Trans. Graph.*, vol. 33, pp. 1–9, Jul. 2014.

[20] V. Megaro, B. Thomaszewski, D. Gauge, E. Grinspun, S. Coros, and M. H. Gross, "Chacra: An interactive design system for rapid character crafting," in *Proc. Eurographics / ACM SIGGRAPH Symp. Comput. Animation*, Copenhagen, Denmark, 2014, pp. 123–130.

[21] M. Bächer, S. Coros, and B. Thomaszewski, "Linkedit: Interactive linkage editing using symbolic kinematics," *ACM Trans. Graph.*, vol. 34, pp. 99: 1–99:8, Jul. 2015.

[22] D. N. Nenchev and M. Uchiyama, "Singularity-consistent path planning and control of parallel robot motion through instantaneous-self-motion type singularities," in *Proc. IEEE Int. Conf. Robot. Automat.*, vol. 2, 1996, pp. 1864–1870.

[23] J. F. O'Brien and J. T. Wen, "Kinematic control of parallel robots in the presence of unstable singularities," in *Proc. IEEE Int. Conf. Robot. Automat. (Cat. No.01CH37164)*, vol. 1, 2001, pp. 354–359.

[24] S. Sen, B. Dasgupta, and A. K. Mallik, "Variational approach for singularity-free path-planning of parallel manipulators," *Mechanism Mach. Theory*, vol. 38, no. 11, pp. 1165–1183, 2003.

[25] S. Faraji and A. J. Ijspeert, "Singularity-tolerant inverse kinematics for bipedal robots: An efficient use of computational power to reduce energy consumption," *IEEE Robot. Automat. Lett.*, vol. 2, no. 2, pp. 1132–1139, Apr. 2017.

[26] C. Innocenti and V. Parenti-Castelli, "Singularity-free evolution from one configuration to another in serial and fully-parallel manipulators," *J. Mech. Des.*, vol. 120, no. 1, pp. 73–79, Mar. 1998.

[27] E. Macho, O. Altuzarra, C. Pinto, and A. Hernandez, *Transitions Between Multiple Solutions of the Direct Kinematic Problem*. Dordrecht, The Netherlands: Springer, 2008, pp. 301–310.

[28] D. Zlatanov, R. Fenton, and B. Benhabib, "Identification and classification of the singular configurations of mechanisms," *Mechanism Mach. Theory*, vol. 33, no. 6, pp. 743–760, 1998.

[29] T. R. Chase and J. A. Mirth, "Circuits and branches of single-degree-of-freedom planar linkages," *J. Mech. Des.*, vol. 115, no. 2, pp. 223–230, Jun. 1993.

[30] D. H. Myszka, A. P. Murray, and C. W. Wampler, "Computing the branches, singularity trace, and critical points of single degree-of-freedom, closed-loop linkages," *J. Mechanisms Robot.*, vol. 6, no. 1, Dec. 2013, Art no. 011006.

[31] C. Gosselin and J. Angeles, "Singularity analysis of closed-loop kinematic chains," *IEEE Trans. Robot. Automat.*, vol. 6, no. 3, pp. 281–290, Jun. 1990.

[32] C. Schumacher, E. Knoop, and M. Bächer, "A versatile inverse kinematics formulation for retargeting motions onto robots with kinematic loops," *IEEE Robot. Automat. Lett.*, vol. 6, no. 2, pp. 943–950, Apr. 2021.

[33] J. Nocedal and S. Wright, *Numerical Optimization*. New York, NY, USA: Springer, 2006.

[34] R. Boudreau and C. M. Gosselin, "The synthesis of planar parallel manipulators with a genetic algorithm," *J. Mech. Des.*, vol. 121, no. 4, pp. 533–537, Dec. 1999.

[35] D. Zhang, Z. Xu, C. M. Mechefske, and F. Xi, "Optimum design of parallel kinematic toolheads with genetic algorithms," *Robotica*, vol. 22, no. 1, pp. 77–84, 2004.

[36] Y. Lou, Y. Zhang, R. Huang, X. Chen, and Z. Li, "Optimization algorithms for kinematically optimal design of parallel manipulators," *IEEE Trans. Automat. Sci. Eng.*, vol. 11, no. 2, pp. 574–584, Apr. 2014.

[37] J. Kennedy and R. Eberhart, "Particle swarm optimization," *Proc. ICNN'95-Int. Conf. Neural Netw.*, vol. 4, 1995, pp. 1942–1948.

[38] J. Ryu and J. Cha, "Volumetric error analysis and architecture optimization for accuracy of hexaslide type parallel manipulators," *Mechanism Mach. Theory*, vol. 38, no. 3, pp. 227–240, 2003.