Concurrent design optimization of the Delta manipulator

Salvador Botello-Aceves, S. Ivvan Valdez, Hector M. Becerra
Center for Research in Mathematics (CIMAT) A.C. Department of Computer Science
Jalisco S/N, Col. Valenciana, Guanajuato, Guanajuato, México
Email: {salvador.botello, ivvan, hector.becerra}@cimat.mx

***Abstract*—The problem of concurrent design of a mechanism can be defined as finding optimal structural parameters and control parameters for a given objective function during the same optimization process. In this paper, a general concurrent optimization methodology for kinematically complex mechanisms is tested using a Delta manipulator. This methodology intends to optimize any structure and control design, using any specified kinematic or dynamic models. Thus, general optimization methods not dependent on mathematical characteristics of the objective function are used. The main contribution of this work is to define, develop and test a general methodology that can generate optimal designs based on workspace and task requirements, such that they guarantee an adequate performance under a set of operating and joint constraints. We test three families of evolutionary algorithms: a genetic algorithm, an evolution strategy and an estimation of distribution algorithm, for two objective functions. The reported results give directions about the most adequate method to tackle the concurrent design problem.**

***Index Terms*—manipulators; evolutionary algorithms; design optimization;**

I. Introduction

Optimal concurrent mechanism design can be defined as *finding optimal structural parameters and control gains for a given objective function during the same optimization process, which are dependent on the kinematic or dynamic model of the mechanism.* The design parameters can be the number of links, links length, position and weight of masses, etc.

We differentiate three types of mechanical optimizations: static, kinematic and dynamic, according to the mathematical model describing the design problem. These models can be differential equations of zero, first and second order, respectively. A common static optimization is a workspace optimization, i.e., the mechanism must reach a given set of points or any point in a given workspace. Thus, we can describe the optimization problem as a maximization of volume [1], [2] or ratio of volume between the reached workspace and the user-desired workspace [3], [4], [5], penalizing collisions and dexterity constrains. The model to deal with this optimization problem consists in geometric relationships (rigid body transformations). For the kinematic and dynamic optimization problems the mechanism, commonly, has to track a time dependent trajectory, which can be given by a time dependent function or a set of points which define a function via interpolation methods. The aim is to minimize the error between a desired and a tracked trajectory, or to minimize the energy consumed by every joint through the whole trajectory, subject to collisions and dexterity constrains [6].

To perform trajectory tracking, it is necessary to use a control technique which feeds back the actuators according to the error. A set of differential equations describes the system, and the controllaw is a function of the error and a set of parameters named control gains. A classical control technique in robotics is the PID (Proportional-Integral-Derivative) controller and its variants as P (only proportional) or PD (Proportional-Derivative) controllers [7]. This control technique uses a factor (control gain) for each of the control parts (proportional, integral and derivative). Hence, the control gains define how well the trajectory is tracked and how much energy is needed for this purpose. The tuning of the control gains is not an easy task in order to achieve an optimum performance. Usually, a concurrent optimal design problem is defined to minimize the integral of the tracking error or the energy used to control the mechanism [8], or the weighted sum of both functions [9] [10].

In this work, we focus on two problems: a) the maximization of the reachable volume in the workspace, and b) the concurrent design problem of links length and control gains. Several algorithmic methodologies have been proposed for automatic search of design parameters, simulating the performance under a given task. For instance, in [11], the optimum design of the control parameters of a serial manipulator is approached, kinematics and dynamics are considered in a task-based problem. The task to perform is defined by a set of target points in the workspace which must be reached by the end-effector. In order to approximate the solution to this problem a memetic algorithm which uses a tunneling algorithm for local search is used. In [1] the maximization of the effective workspace, subject to a dexterity constraint, for a Delta manipulator is approached. The same problem is later replicated and expanded using evolutionary algorithms and different case studies in [12], [13], [2].

In this article, a concurrent optimization methodology for the Delta robot is presented. As far as we know, the concurrent optimization applied to the kinematic model of a Delta robot has never been implemented. The very same methodology can be applied to other kinematically complex mechanism. These mechanisms are such that, due to a large number of degrees of freedom, redundancy or multiple kinematic chains, the kinematics is, typically, solved numerically rather than in a closed form. In consequence of the mechanism complexity, the mathematical properties of an objective function such as

978-1-5090-5105-2/16/$31.00 ©2016 IEEE

gradient, smoothness and continuity, are not available or require a lot of computational or analytical work. The addressed problems might have several local minima and gradient-based methods are not suited for solving this kind of problems. Thus, we propose to use gradient-free methods such like genetic algorithms and other evolutionary optimizers, which provide a flexible way of treating with the concurrent design optimization problem. In this context, genetic algorithms (GAs) are the most widely applied for the concurrent design problem [14], [15], [16], nevertheless, the performance of other gradient-free optimizers has not been deeply studied. One of the contributions of this work is to present a set of optimizers from three families: The omnioptimizer [17], a well performed genetic algorithm which basically uses the same crossover and mutation operations than the NSGA-II [18], the BUMDA [19] from the estimation of distribution algorithm family, a kind of algorithm which replaces the crossover and mutation operations by the estimation and sampling of a probability distribution, and the CMA-ES, an evolution strategy which uses a normal distribution for mutation favoring the most promising directions. We compare the performance of these optimizers on the two design problem aforementioned, in order to elucidate if any of them performs the best for such problems.

We present two different objective function models for the optimal design of mechanisms: the first considers the maximization of a regular workspace, and the second one combines the concurrent optimization of both structure geometry and control parameters for a task-based problem. In particular, as aforementioned, these methods are tested on a simulated Delta parallel manipulator [20], but they are applicable for other mechanisms, e.g. a redundant serial manipulator of several degrees of freedom.

The organization of this article is as follows: in Section II we discuss and introduce the different models for optimum mechanism design. In Section III we briefly review the three state-of-the-art evolutionary algorithms used for approximating the solution to the optimum design problems. In Section IV we introduce the general work-flow applied to optimum mechanism design. In Section V we introduce the two case studies of actuated mechanism along with the results of applying the different optimization methods. Finally, in Section VI we present the final discussion and concluding remarks.

II. Objective Functions for Optimal Design

In order to define the optimization problem, we consider two cases: **1) Maximization of a regular workspace subject to a constant norm of the links lengths. 2) Minimization of the trajectory tracking error for concurrent optimal design.**

Nevertheless, the same optimization algorithms and kinematic simulators can be used for approaching any of the optimization problems, the most adequate objective function model can be selected according to the application and user needs.

Let us define the general terms used in the optimization models. Consider a mechanism with m joints and a actuators for all the kinematic chains, as well as d degrees of freedom (*DOF*) of the end-effector. The mechanism is defined by a set of design parameters $\alpha \in \mathbb{R}^p$, for instance, the lengths of the links, the relative position of each link, the relative arrangement between each axis, the size and shape of the end effector, etc. Let us define a set of Cartesian and orientation coordinates of points in a target workspace W as $X \in \mathbb{R}^{d\times n}$, where n is the number of points. Using the target data we can compute a set of values for the actuated joint variables $\theta \in \mathbb{R}^a$, and passive joint variables $\phi \in \mathbb{R}^{m-a}$, as in Equations (1) and (2).

$$\begin{aligned} \theta_i &= \theta_i(X_k, \alpha), \quad i = 1, \ldots, a, && (1)\\ \phi_j &= \phi_j(X_k, \alpha), \quad j = 1, \ldots, m-a. && (2) \end{aligned}$$

Let us define $x \in \mathbb{R}^d$ as the end-effector position and orientation. Thus, the end-effector velocities are given by:

$$\dot{x} = J\dot{\theta} \tag{3}$$

where $J = J(x, \theta, \alpha) \in \mathbb{R}^{d\times a}$ is the Jacobian matrix, which relates the angular velocities $\dot{\theta}$ with the Cartesian velocities $\dot{x}$ of the end-effector.

A. Objective 1: Optimal mechanism design for the maximization of a regular workspace

This problem consist in finding the maximum regular workspace for a manipulator whose sum of the links lengths is normalized to the unity [2]. The problem is subject to a manipulability constraint measured by the inverse of the condition number of the Jacobian matrix $\kappa(J)$. This is defined as $\kappa(J) = \sigma_{\min}(J)/\sigma_{\max}(J)$ [21], where $\sigma_{\min}(J)$ and $\sigma_{\max}(J)$ are the minimum and maximum singular values of the Jacobian matrix, respectively. Therefore, $\kappa \in [0, 1]$. In order to decouple translational and rotational manipulability of the end-effector, we can rewrite Equation (3) as follows:

$$\dot{x} = \begin{bmatrix} \dot{x}_t \\ \dot{x}_r \end{bmatrix} = \begin{bmatrix} J_t \\ J_r \end{bmatrix} \dot{\theta} \tag{4}$$

where $\dot{x}_t$ and $\dot{x}_r$ are translational and rotational velocities of the end-effector, respectively. Thus, we can compute separately manipulability measures for position $\kappa(J_t)$ and orientation $\kappa(J_r)$. The optimization problem is defined as follows:

$$\max_{\alpha} \mathcal{F}(\alpha) = \Phi(\alpha) \tag{5}$$

subject to

$$\begin{gathered} \kappa(J_t(X, \theta, \alpha)) \geq \gamma_1; \qquad (6)\\ \kappa(J_r(X, \theta, \alpha)) \geq \gamma_2;\\ \theta_i^{\min} \leq \theta_i \leq \theta_i^{\max}\\ \phi_j^{\min} \leq \phi_j \leq \phi_j^{\max}\\ \sum_{k=1}^{q} l_k(\alpha) = \tau; \end{gathered}$$

where $\Phi(\alpha)$ is the length of the side of a cube, which is equivalent to maximize its volume, $i = 1, \ldots, a$, $j = 1, \ldots, m - a$, γ_1 and γ_2 are position and orientation manipulability bounds, respectively. τ is a given normalizing constant, set to $\tau = 1$ for this work, this constraint removes the dimension effects by normalizing the design parameters α of the manipulator.

For both objective functions we assume that the joint limits $[\theta_i^{min}, \theta_i^{max}, \phi_j^{min}, \phi_j^{max}]$ are set in such a way that auto-collisions are avoided, i.e., the working ranges of the joint angles are such that mechanical interference between links is not possible.

B. Objective 2: Concurrent optimal design of kinematic control

In this problem, we consider an actuated mechanism under a control action. Given a trajectory (a function of positions in time), we aim to minimize the error between the desired trajectory and the current position of the mechanism. The error is a time-dependent function. Hence, the optimization problem considers the integral of the absolute value of the control signal, as follows:

$$\min_{\alpha} \mathcal{F}(\alpha) = k_P(\alpha) \int_{t_0}^{t_n} ||e(\alpha, t)|| \partial t + k_D(\alpha) \int_{t_0}^{t_n} ||\dot{e}(\alpha, t)|| \partial t \quad (7)$$

subject to

$$\theta_i^{\min} \leq \theta_i \leq \theta_i^{\max}$$
$$\phi_j^{\min} \leq \phi_j \leq \phi_j^{\max}$$

where $i = 1, \ldots, a$, $j = 1, \ldots, m - a$, $k_P(\alpha)$ and $k_D(\alpha)$ are proportional and derivative control gains, respectively, which are included as optimization variables in α. Additionally, $e(\alpha, t) = X^{Cal}(\alpha, t) - X^{Des}(t) \in \mathbb{R}^d$ and $\dot{e}(\alpha, t) = \dot{X}^{Cal}(\alpha, t) - \dot{X}^{Des}(t) \in \mathbb{R}^d$ are translation and velocity errors, respectively. $X^{Cal}(\alpha, t)$ and $\dot{X}^{Cal}(\alpha, t)$ are the translation and velocity coordinates (including orientation in both cases) of the manipulator at an instant t. Likewise, $X^{Des}(t)$ and $\dot{X}^{Des}(t)$ are the desired translation and velocity coordinates imposed to the manipulator by the desired trajectory. The kinematic control that allows the manipulator to track a desired trajectory is introduced as follows:

$$\dot{\theta}_{t+1} = J_t^{\dagger}(\dot{X}^{Des}(t) - k_P e_t - k_D \dot{e}_t) \quad (8)$$

where $J_t^{\dagger} = J_t^{\top}(J_t J_t^{\top})^{-1} \in \mathbb{R}^{a \times d}$ is the right pseudo-inverse of J_t. Notice that if $a = d$ then $J_t^{\dagger} = J_t^{-1}$. The values of the joints variables in the next instant of time are obtained using the Euler method, so that $\theta_{t+1} = \theta_t + \Delta t \ \dot{\theta}_{t+1}$. The translation variables (including orientation) in the next instant of time can be computed by using the Cartesian velocity, as follows: $X_{t+1}^{Cal} = X_t^{Cal} + \Delta t \ \dot{X}_{t+1}^{Cal}$, where $\dot{X}_{t+1}^{Cal} = J_t \dot{\theta}_{t+1}$.

III. Evolutionary Algorithms

Evolutionary Algorithms (EAs) have been used for approximating optimum mechanism designs, such as synthesis of actuated mechanisms and optimal control [2]. The EAs used in this work can be classified in three families: Estimation of Distribution Algorithms (EDAs), Evolution Strategies (ESs), and Genetic Algorithms (GAs). Nevertheless, all of them can be classified as evolutionary algorithms, each of them as a particular way of working. The Omnioptimizer re-combinates the most promising individuals, selected using binary tournament, and exploration is maintained via a mutation operator. The evolution strategy, CMA-ES, uses a reproduction operator which favors promising directions. The BUMDA estimates a probability distribution by using the best individuals, the better an individual is, the higher the weight of such individual in the estimation formula. Thus, the resulting probability functions favor to sample the best regions already known. The Omnioptimizer shares the characteristic of sampling the regions where the best individuals are, while the CMA-ES samples in directions where the best individuals are generated. A brief introduction to each of the algorithms is given in the rest of the section.

1) The Omni-optimizer [22]: is a general optimization GA, which is used, in this case, to solve a single objective problem. Nevertheless, it is an optimizer that can be applied to a wide range of problems from monoobjective to multiobjective with and without constraints in discrete and continuous domains, in our problem we used it as a classic GA, with simulated binary crossover (SBX) [18] and polynomial mutation. These are the same operators than those used in the NSGA-II [18] which probably, is the most widely used GA in the last ten years.

2) The Boltzmann Univariate Marginal Distribution Algorithm (BUMDA) [19]: is an EDA that uses a univariate Gaussian model to approximate the Boltzmann distribution, whose energy function is related with the objective function. That means that the better an individual is the most probable is to be sample the region where it is. The mean and variance parameters of the Gaussian model are derived from the analytical minimization of the Kullback-Leibler divergence.

3) The Covariance Matrix Adaptation Evolutionary Strategy (CMA-ES) [23]: uses a multivariate Gaussian model. Parent solutions are used to determine the size, position and orientation of a Gaussian distribution used to sample the children candidate solutions. One of the most important characteristics of the CMA-ES is that the orientation of the multivariate Gaussian model directs the search to promising regions, in a kind of descent path.

IV. General work-flow for optimization of a mechanism

Let us describe the proposed general methodology for optimization of a kinematically complex mechanism. This methodology can be used for static, kinematic or dynamic task, in the last cases considering a concurrent design problem: the optimization of geometry and control parameters simultaneously. Figure 1 shows a graphic representation of the methodology, which summarizes the process in three main steps: 1) Firstly, the mechanism to be optimized must be selected by defining a set of design parameters, e.g., masses, fixed lengths, inertial moments, desired workspace size and shape, etc. as well as

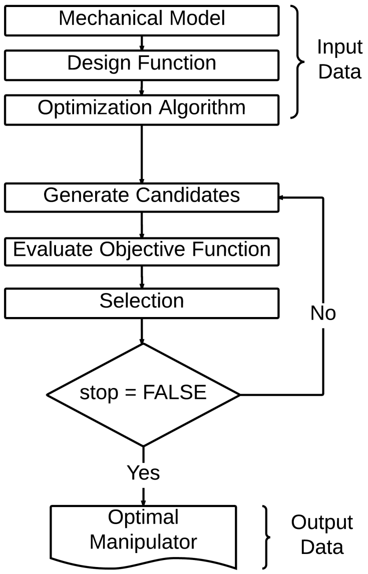


Fig. 1. Graphic representation of the concurrent optimization method for kinematically complex mechanisms.

a kinematic or dynamic model. 2) Secondly, the objective function must be selected, thus a control method, trajectory or workspace must be also selected. 3) Lastly, an optimization algorithm must be selected, currently from the three options described in the previous section, and the parameters of such algorithm must be introduced. These three steps are flexible to the user choice and provide a whole methodology whose output is a set of design parameters (geometric and/or control parameters) that approximates the optimal solution.

V. Case study: Delta parallel manipulator

The case study is a Delta parallel manipulator, which is considered a kinematically complex mechanism according to our definition. The Delta manipulator, as shown in Figure 2(c), is a 3-DOF purely translational parallel robot, it is well-known for its high speed and accuracy [20].

The kinematic parameters are depicted in Figure 2, where a denotes the length of arms $\overline{A_iB_i}$, b the length of the parallelogram $\overline{B_iC_i}$, $R = \overline{OA_i}$ and $r = \overline{PC_i}$, with O and P being centers of the base and the moving platform, respectively. By obviating passive joint variables from the kinematic equation, we derive loop-closure equations relating the actuated joint variable $\Phi_1 = [\Phi_{1,1}\Phi_{1,2}\Phi_{1,3}]$ with the reference point P as $(x-(d+a\cos\Phi_{1,i})\cos\phi_i)^2+(y-(d+a\cos\Phi_{1,i})\sin\phi_i)^2+(z+a\sin\Phi_{1,i})^2-b^2=0$, for $i=1,\ldots,3$, where $d = R - r$ and ϕ_i denotes the angle between the i^{th} sub-chain and x axis of the reference framework, in this case $\phi_i = (2\pi/3)\ (i-1)$ for $i = 1,\ldots,3$.

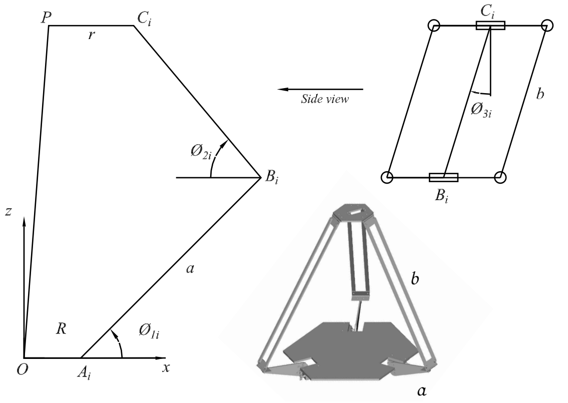


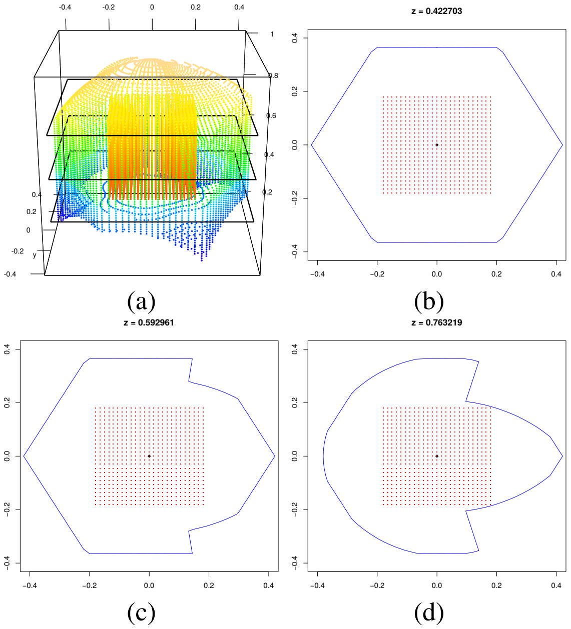
Fig. 2. Architecture and the $i-th$ subchain of the Delta robot.

Each optimization algorithm is executed 10 times for each design problem to present a statistical comparison. The stopping criteria are the following: the CMA-ES stops if the best objective function value is less than 10^{-9}, or the difference between two consecutive generations is less than 10^{-20}. The Omnioptimizer stops when it reaches 40 generations. The BUMDA stops if the maximum variance is less than 10^{-12}. All algorithms stops at a maximum of $10,000 \cdot p$ evaluations, for p optimization variables.

A. Normalized optimal mechanism design.

We define a regular-cubic workspace W. The length of an edge of the cubic workspace is denoted by $2l$, it is used to compute the objective function in (5).

The center of the resultant maximal effective regular workspace is unknown for the evaluation of each candidate solution, but it is known that the manipulator is symmetrical with respect to the axis z. In consequence, the coordinates of the center of the workspace has the form $(0,0,z_c)$. In our work z_c is found by intensive search, discretizing the axis in partitions of size 10^{-5}. The set of design parameters are denoted as $\alpha = [a\ b\ d]^T$. Joint limits and manipulability constraints are taken as in [2], and the inverse kinematics for the Delta robot is computed according to the approach in [24]. The angles are constrained to the following limits $-40 \leq \Phi_{3,i} \leq 40$, $-45 \leq \Phi_{1,i} + \Phi_{2,i} \leq 180$, and $-30 \leq \Phi_{1,i} \leq 100$. Figure 3 presents the optimal approximation of the whole workspace for the Delta robot, as well as the enclosed regular workspace. Fig. 3 (a) shows the minimum and maximum robot workspace with the optimal design. Figures 3(b), (c) and (d) show a cross section of the workspace at heights of 0.4137, 0.5940 and 0.7742 units, respectively. Table I shows the best solution found by each algorithm for the lengths $\{a, b, c\}$ and the corresponding objective function as well as the mean and standard deviation of ten independent executions. As can be notice, the best solution is found by the CMA-ES, nevertheless, the lengths are similar for all the algorithms. In addition, we report the mean and standards deviation of the values, we can observe that in average the CMA-ES also performs the best, and has the smallest standard



(a) (b)

(c) (d)

Fig. 3. Workspace visualization, (a) 3-D total and regular Workspace, (b) cross-section at 0.4137 , (c) cross-section at 0.5940 , (d) cross-section at 0.7742.

TABLE I
BEST SOLUTION FOR EQUATION (5), LINKS LENGTH $\{a, b, d\}$ AND FITNESS $\mathcal{F}(\alpha)$. $\overline{\mathcal{F}}(\alpha)$ AVERAGE AND STANDARD DEVIATION FROM 10 RUNS.

EA	a	b	d	$\mathcal{F}(\alpha)$	$\overline{\mathcal{F}}(\alpha)$
CMA-ES	**4.01e-1**	**5.67e-1**	**3.23e-2**	**1.80e-1**	**1.70e-1± 1.29e-2**
Omini	3.64e-1	5.61e-1	2.57e-2	1.75e-1	1.22e-1±5.91e-2
BUMDA	3.81e-1	5.77e-1	3.67e-2	1.68e-1	1.52e-1±1.92e-2

deviation, that means that it reaches almost the same solution most of the times.

B. Optimal concurrent mechanism design for kinematic control.

For the sake of validating the proposal, we use a well mathematically defined trajectory, nevertheless our proposal considers a set of points in four dimensions (x, y, z and time) in benefit of generalizing the methodology. We use a tridimensional spiral function sampled in 100 equidistant points in time to generate the input points for the algorithm. Additionally, the desired orientation is computed, for each point, as the corresponding direction cosines of the point, i.e, the desired trajectory for the configuration of the end-effector is defined by $X_i(t) = \langle n \sin(2.0\pi \cdot m), n \cos(\pi \cdot m), l^4 + 0.2 \rangle$, where $m = \frac{2.0 \cdot t - t_f}{t_f}$, $n = 0.3$, $l = 0.5$, $0.0 \leq t \leq t_f$, $t_f = 30.0$ s and $i = 1, \ldots, 100$. A very large value proportional to the remaining simulation time is assigned to unfeasible mechanisms which can not be simulated due to singularities.

Table II shows results obtained by the EAs for the Delta robot. Notice that for both controls, P and PD, the best structure parameters are almost the same, meaning that those parameters might be at least one of the local optima. Nevertheless, the proportional gain is quite large for the P than for the PD controller, meaning that the P controller is using more energy in order to achieve a reasonable performance, which is clear in the large value of the objective functions in comparison to the PD controller. In addition, notice that all algorithms reach basically the same objective function value, but the CMA-ES is more stable, and delivers the same result for the PD case, and a very similar for the P control. Figure 4 shows the logarithmic absolute End-Effector Position Error (*EEPE*) over time for the P and PD control, respectively. In both cases, this error remains low, which means, that the reference trajectory is tracked with good precision.

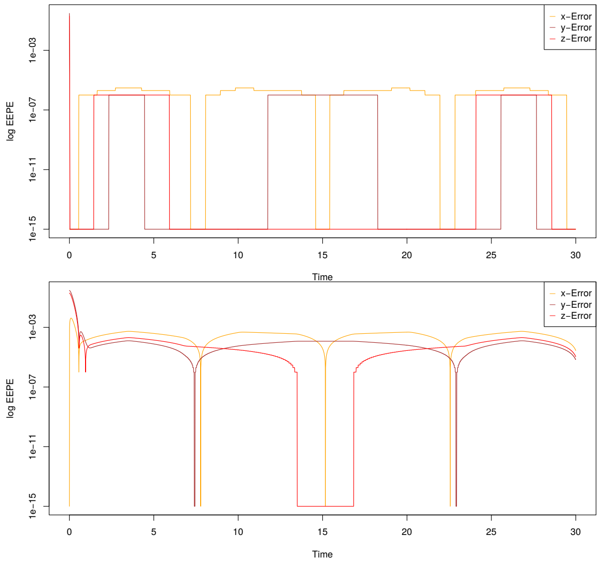


Fig. 4. Control Path Tracking for the Delta robot, (Top) Logarithmic absolute *EEPE* over time for P control, (Bottom) Logarithmic absolute *EEPE* over time for PD control

TABLE II
BEST SOLUTION FOR EQ. (7), LINKS LENGTH $\{a, b, d\}$, CONTROL GAINS $\{k_P, k_D\}$ AND FITNESS $\mathcal{F}(\alpha)$. $\overline{\mathcal{F}}(\alpha)$ AVERAGE AND STANDARD DEVIATION FROM 10 RUNS.

EA	a	b	d	K_P	K_D	$\mathcal{F}(\alpha)$	$\overline{\mathcal{F}}(\alpha)$
			Proportional Control				
CMA-ES	4.35e-1	4.90e-1	7.38e-2	9.99e+1	-	3.71e-1	3.71e-1±0.0
Omini	4.40e-1	4.87e-1	7.56e-2	9.99e+1	-	3.71e-1	4.58e-1±1.24e-1
BUMDA	4.35e-1	4.92e-1	7.64e-2	1.00e+2	-	3.71e-1	4.05e-1±7.34e-2
			Proportional-Derivative Control				
CMA-ES	4.36e-1	4.89e-1	7.35e-2	9.99e-1	8.28e-1	8.74e-3	8.79e-3±2.49e-5
Omini	4.37e-1	4.88e-1	7.66e-2	9.99e-1	8.28e-1	8.76e-3	8.82e-3±3.32e-5
BUMDA	4.35e-1	4.92e-1	7.36e-2	9.99e-1	8.28e-1	8.75e-3	8.82e-3±4.34e-5

TABLE III
HYPOTHESIS TESTS ABOUT THE MEAN OF THE OBJECTIVE FUNCTION.

Problem	CMA-ES vs OMNI	CMA-ES vs BUMDA	OMNI vs BUMDA
Normalized design	**CMA-ES (0.0)**	**CMA-ES(6.25e-4)**	Neither
P control	**CMA-ES (0.0)**	**CMA-ES (0.0)**	Neither
PD control	**CMA-ES (1.85e-4)**	**CMA-ES (3.38e-2)**	Neither

According to the results on Tables I and II, the optimization methods deliver similar values of the design variables as well as objective function values in the same order of magnitude, all of them fulfill the task and constraints. Nevertheless the methods perform similar, considering the objective function values as well as their standard deviation, we can see that CMA-ES reports the best objective function value and the smallest standard deviation. In order to objectively compare the optimization methods we use hypothesis tests to compare pairs of them via the Boostrap methodology. The null hypothesis is that both algorithms deliver the same mean of the objective function value, while the alternative is that one of them performs better (delivers a smaller mean of the fitness for minimization, or a greater value for maximization). All possible combinations are tested and the results are reported in Table III. If one of the algorithms can be considered the best, we report its name and the p-value. As can be observed the **CMA-ES** clearly is better than the other two. Additionally, even though the BUMDA seems to perform better than the Omnioptimizer according to numerical results, the hypothesis test says that neither of them can be consider better than the other.

VI. Conclusions

In this work we have reviewed two mathematical models for optimal mechanism design. We have shown that the problems can be address under a unified framework, although they are different in the goal, optimization variables and complexity. To use the same methodology for the different problems and mechanisms, the optimization algorithm must be independent of the type of mechanism and numerical simulation performed. Hence, we propose to use a set of evolutionary algorithms, which are of the family of black-box optimization.

It is interesting that, in the case of concurrent design (control and geometry), the cases optimized for the P and PD controls performs (visually) equally well, compensating with similar precision the error in the trajectory. EAs can not guarantee converge to the optimum, but it is worth to notice that the different algorithms deliver similar values of the decision variables, in consequence, we can argue that is very possible that the solutions reported are, at least, one of the best local optima.

According to the encouraging results, future work contemplate to unify different algorithms, case studies, and optimization problems in a single tool for optimal design under different kinematics and dynamics requirements. Possibly the approximations found by the evolutionary optimizers could be improved by local search. Hence, in addition, future work contemplate to combine global and local search.

References

[1] Y. Lou, G. Liu, and Z. Li, "Randomized optimal design of parallel manipulators," *Automation Science and Engineering, IEEE Transactions on*, vol. 5, no. 2, pp. 223–233, 2008.

[2] Y. Lou, Y. Zhang, R. Huang, X. Chen, and Z. Li, "Optimization algorithms for kinematically optimal design of parallel manipulators," *Automation Science and Engineering, IEEE Transactions on*, vol. 11, no. 2, pp. 574–584, 2014.

[3] C. Riaño, C. Peña, and A. Pardo, "Approach in the optimal development of parallel robot for educational applications," in *Proceedings of the WSEAS international conference on Recent Advances in Intelligent Control, Modelling and Simulation (ICMS)*, vol. 145, 2014.

[4] V. Kumar, S. Sen, S. S. Roy, C. Har, and S. Shome, "Design optimization of serial link redundant manipulator: An approach using global performance metric," *Procedia Technology*, vol. 14, pp. 43–50, 2014.

[5] S. Patel and T. Sobh, "Goal directed design of serial robotic manipulators," in *American Society for Engineering Education (ASEE Zone 1), 2014 Zone 1 Conference of the*. IEEE, 2014, pp. 1–6.

[6] T. Ravichandran, G. Heppler, and D. Wang, "Task-based optimal manipulator/controller design using evolutionary algorithms."

[7] G. Reynoso-Meza, J. Sanchis, X. Blasco, and M. Martínez, "Algoritmos evolutivos y su empleo en el ajuste de controladores del tipo pid: Estado actual y perspectivas," *Revista Iberoamericana de Automática e Informática Industrial RIAI*, vol. 10, no. 3, pp. 251–268, 2013.

[8] M. Pellicciari, G. Berselli, F. Leali, and A. Vergnano, "A method for reducing the energy consumption of pick-and-place industrial robots," *Mechatronics*, vol. 23, no. 3, pp. 326–334, 2013.

[9] H. V. H. Ayala and L. dos Santos Coelho, "Tuning of pid controller based on a multiobjective genetic algorithm applied to a robotic manipulator," *Expert Systems with Applications*, vol. 39, no. 10, pp. 8968–8974, 2012.

[10] T. Ravichandran, D. Wang, and G. Heppler, "Simultaneous plant-controller design optimization of a two-link planar manipulator," *Mechatronics*, vol. 16, no. 3, pp. 233–242, 2006.

[11] R. dos Santos, V. Steffen, and S. Saramago, "Optimal task placement of a serial robot manipulator for manipulability and mechanical power optimization," *Intelligent Information Management*, vol. 2, no. 9, pp. 512–525, 2010.

[12] X. Liu and J. Wang, "A new methodology for optimal kinematic design of parallel mechanisms," *Mechanism and Machine Theory*, vol. 42, no. 9, pp. 1210–1224, 2007.

[13] A. Omran, M. Bayoumi, A. Kassem, and G. El-Bayoumi, "Optimal forward kinematics modeling of stewart manipulator using genetic algorithms," *Jordan Journal of Mechanical and Industrial Engineering*, vol. 3, no. 4, pp. 280–293, 2009.

[14] R. Boudreau and C. Gosselin, "The synthesis of planar parallel manipulators with a genetic algorithm," *Journal of mechanical design*, vol. 121, no. 4, pp. 533–537, 1999.

[15] S. Khatami and F. Sassani, "Isotropic design optimization of robotic manipulators using a genetic algorithm method," in *Intelligent Control, 2002. Proceedings of the 2002 IEEE International Symposium on*. IEEE, 2002, pp. 562–567.

[16] P. K. Jamwal, S. Xie, and K. C. Aw, "Kinematic design optimization of a parallel ankle rehabilitation robot using modified genetic algorithm," *Robotics and Autonomous Systems*, vol. 57, no. 10, pp. 1018–1027, 2009.

[17] K. Deb and S. Tiwari, "Omni-optimizer: A generic evolutionary algorithm for single and multi-objective optimization," *European Journal of Operational Research*, vol. 185, no. 3, pp. 1062–1087, 2008.

[18] K. Deb, *Multi-objective optimization using evolutionary algorithms*. John Wiley & Sons, 2001, vol. 16.

[19] S. Valdez, A. Hernández, and S. Botello, "A boltzmann based estimation of distribution algorithm," *Information Sciences*, vol. 236, pp. 126–137, 2013.

[20] R. Clavel, "Delta, a fast robot with parallel geometry," in *Proc. 18th Int. Symp. on Industrial Robots, Lausanne, 1988*, 1988, pp. 91–100.

[21] C. Klein and B. Blaho, "Dexterity measures for the design and control of kinematically redundant manipulators," *The International Journal of Robotics Research*, vol. 6, no. 2, pp. 72–83, 1987.

[22] A. Klanac and J. Jelovica, "A concept of omni-optimization for ship structural design," *Advancements in Marine Structures, Proceedings of MARSTRUCT*, pp. 473–481, 2007.

[23] N. Hansen, A. Niederberger, L. Guzzella, and P. Koumoutsakos, "A method for handling uncertainty in evolutionary optimization with an application to feedback control of combustion," *Evolutionary Computation, IEEE Transactions on*, vol. 13, no. 1, pp. 180–197, 2009.

[24] M. López, E. Castillo, G. García, and A. Bashir, "Delta robot: inverse, direct, and intermediate jacobians," *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, vol. 220, no. 1, pp. 103–109, 2006.