Optim Eng
DOI 10.1007/s11081-014-9254-x

Synergetic structure–control design via a hybrid gradient-evolutionary algorithm

Miguel G. Villarreal-Cervantes · Carlos Alberto Cruz-Villar · Jaime Alvarez-Gallegos

Received: 9 April 2012 / Accepted: 6 February 2014

Abstract This paper proposes a synergetic approach to design a planar parallel robot with its control system. In this proposal, the design problem is stated as a dynamic optimization problem with dynamic and static constraints on both the robot parameters and the control input to the robot. Control parameterization via PID controllers is used to rewrite the dynamic optimization problem as a nonlinear programming problem, which is solved by using a hybrid gradient-evolutionary optimization technique. The dynamic optimization problem presents singularity regions in the design space requiring the use of the proposed hybrid gradient-evolutionary optimization technique. The rationale behind the proposed hybrid algorithm lies in using a exploratory search mechanism for finding the initial guess to the fine search mechanism, which is used to search in a local region of a solution. We discuss both the results of the proposed optimization technique and the experimental results of the robot designed with the proposed approach. In addition, the result provided by the proposed synergetic design approach is compared with a sequential design approach, showing the advantages of the synergetic approach.

Keywords Hybrid optimization algorithm · Structure–control design · Parallel robot · Differential evolution

M. G. Villarreal-Cervantes
Mechatronics Group, Instituto Politécnico Nacional - CIDETEC, Mexico City, Mexico
e-mail: mvillarrealc@ipn.mx

C. A. Cruz-Villar (✉) · J. Alvarez-Gallegos
Mechatronics Section, Electrical Engineering Department, CINVESTAV, Mexico City, Mexico
e-mail: cacruz@cinvestav.mx

J. Alvarez-Gallegos
e-mail: jalvarez@cinvestav.mx

Published online: 19 March 2014

1 Introduction

The integration of mechanical systems (mechanisms, machine parts, precision mechanics) with electronic systems (microelectronics, power electronics technology, sensors and actuators) and information technology (systems theory, control and automation, software engineering, artificial intelligence, etc.) results in integrated systems called mechatronic systems. Design of mechatronic systems must involve an optimal balance of different areas of engineering such as mechanical engineering, electrical and electronics engineering. In this work such an optimal balance will be referred to as synergetic design approach. The quest of such synergy has led to various design methodologies, which have generated high-performance mechatronic systems.

Several design methods have been proposed in order to comply with a synergetic design. One approach to accomplish an integrated design process is to perform an iterative design (Messac et al. 1985; Pil and Asada 1996; Zhang et al. 1999). In an iterative design process, the mechanical structure is designed first, by taking into account mechanical performance criteria. Then, the control system is designed by taking into account the control objective. If design specifications are not met, the mechanical structure is redesigned and after that, the control system is also redesigned. This process is iteratively performed until the design specifications are met. However, designing in such a way does not arise a synergetic design, as the mechanical and control parameters are not specified to simultaneously consider all the performance criteria.

In order to avoid iterative designs, several researchers have proposed design methodologies that consider in a single design stage, both mechanical and control design specifications (Fu and Mills 2005). Ravichandran et al. (2006) present a methodology based on numerical optimization techniques for simultaneously optimizing design parameters of a two-link planar rigid manipulator and a nonlinear gain PD controller designed for performing multiple tasks. They modified both the mechanical structure by changing the mass of a counter-weight system in each link, and the control structure by changing its parameters. The performance functions are the gravity balance of the manipulator and the angular position error of the links. The $(\mu + \lambda)$-ES algorithm is used to solve the problem. Cruz-Villar et al. (2009) develop a concurrent redesign approach in order to minimize the required time to force an already existing under-actuated robot to travel between two specific points. The redesign procedure simultaneously finds the optimal mass and location of a counter-weight system and the optimal switching intervals of an open-loop bang-bang control law that fulfills the design objective. The projected gradient method is implemented to solve the problem. Yan and Yan (2009) state an integrated design approach to design a planar four-bar linkage driven by a servomotor. The mechanical structure parameters, such as the link lengths and counter-weights, as well as control parameters, such as the gains of the controller and the control points of the input-speed trajectory, are considered as the design variables to optimize the shaking force and moment, the speed of the trajectory tracking and the motor power dissipation. They used the Genetic Algorithm to solve the problem. An important issue in Yan and Yan (2009) is that the proposed integrated design approach is

compared with the iterative design approach and the proposed integrated design approach yields a better robot performance than the iterative one.

In this paper, the synergetic design of structure and control of a planar parallel robot with five rotational joints and two degrees of freedom (5R 2DoF) and its control system is set to an optimization framework, where the mechanical and control design criteria are included as two weighted terms in the performance index to be optimized. Therefore, the resulting optimization problem is a dynamic optimization one, since the dynamic model of the robot is represented as an equality constraint to be fulfilled. The main characteristic of the established dynamic optimization problem is that it presents singularities conditions. Those conditions are presented when the parallel manipulator gains one or more degrees of freedom and therefore loses its stiffness completely. In such conditions the dynamic model of the system can not be computed, such that gradient based techniques are not useful for finding the optimal solution.

To numerically solve nonlinear dynamic optimization problems, transcription methods such as sequential and simultaneous methods have been used to transform a dynamic optimization problem into a nonlinear programming problem. In the sequential method, only the control variables are discretized, while in the simultaneous method, both state variables and control variables are discretized (Betts 2010). The main rationale behind in this work is similar to the sequential approach in the sense that the control signals are parameterized. However, unlike previous studies that consider the system as a classical open loop optimal control problem, here the closed loop system behavior is considered, where control signals are parameterized in terms of the used controller (Alvarez-Gallegos et al. 2005). Thus, control parameterization in terms of orthonormal signals, such as B-splines or wavelets is disregarded. Moreover, the link parameters of the robot are also considered as independent variables in the optimization problem.

On the other hand, evolutionary algorithms (EAs) inherently and simultaneously explore a set of possible solutions, working with several initial starting conditions, called population (Salomon 1998; Arnold and Salomon 2007). Thus, EAs avoids the sensitivity of the algorithm to initial conditions. In addition, EAs are less susceptible than gradient search to problem dependent characteristics, such as highly nonlinear problems (problems with several solutions to the local level), discrete problems, etc.

In reference Villarreal-Cervantes et al. (2010) the design of a 5R 2DoF robot is addressed by using a constraint handling differential evolution algorithm, and simulation results are presented.

The aim of this paper is twofold: (i) the proposal of the hybrid gradient-EA to simultaneously design the 5R 2DoF robot and its control of reference Villarreal-Cervantes et al. (2010). (ii) The real validation of the obtained design with the synergetic structure–control approach, where the experimental results (laboratory testing) are performed and discussed. Moreover a comparison between synergetic and sequential approaches is presented for the design of the 5R 2DoF robot.

The rest of the paper is organized as follows, Sect. 2 presents the formulation of the dynamic optimization problem for designing the parallel robot. Section 3 presents both the hybrid evolutionary-gradient algorithm to solve the dynamic

optimization problem, and its performance to solve the problem of robot design. The comparison of the proposed synergetic structure–control design approach with a sequential design approach is discussed in Sect. 4. Section 5 details the experimental results of the robot, which was manufactured using the parameters as given by the optimal solution obtained with the proposed design approach. Finally, conclusions are drawn in Sect. 6.

2 Synergetic structure–control design as a dynamic optimization problem

In the following, we present the proposed approach to design a parallel robot, considering both the robot dynamic model and the robot controller, as well as design constraints and design objectives. From a mathematical point of view the design problem stands for a dynamic optimization problem.

2.1 Five link planar parallel robot

Let us consider the planar parallel robot shown in Fig. 1. In order to obtain the system dynamics, the reduced model method (Ghorbel et al. 2000) is employed. This method considers the parallel structure as a constrained system consisting of a free system and a closed-chain constraint. The free system is obtained by virtually cutting off the closed chain linkage at point "c", then, it is conformed by two serial robots. Hence, the links conforming the first serial robot are a_{s_1} and a_{s_3}, while a_{s_2} and a_{s4} are the links corresponding to the second serial robot. The dynamic model for the free system can be obtained by using the Lagrangian approach and is given

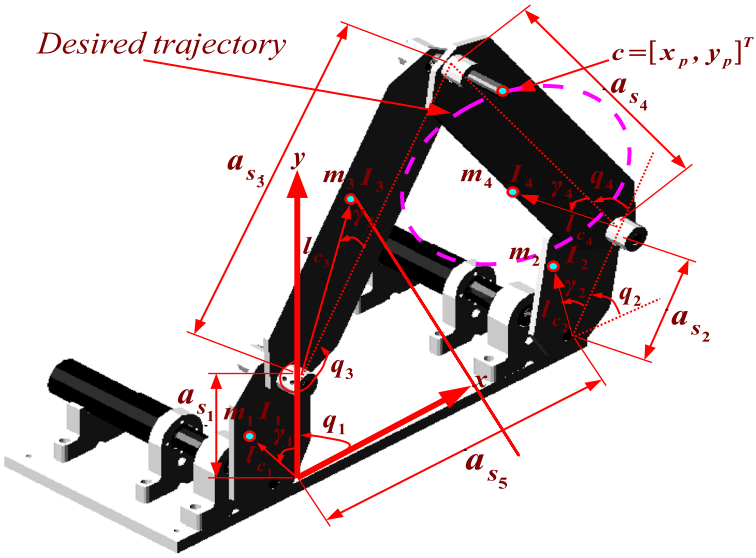

Fig. 1 Schematics of the planar five link parallel robot

by (1), where $\boldsymbol{q}' = [q_1', q_2', q_3', q_4']^T \in R^4$ is the independent generalized coordinate vector for the free system, q_1', q_2' are the angular position of the actuated links, q_3', q_4' are the angular positions of the unactuated links, $\widetilde{\boldsymbol{\tau}} = [\tau_1, \tau_2, 0, 0]^T \in R^4$ is the input torque vector and $\dot{\zeta}$ represents the time derivative of ζ. The inertia, Coriolis and gravity matrices are given by (2), (3) and (4) respectively, where $\bar{g} = 9.81\frac{\mathrm{m}}{\mathrm{s}^2}$ is the gravitational acceleration constant.

$$\boldsymbol{D}'(\boldsymbol{q}')\ddot{\boldsymbol{q}}' + \boldsymbol{C}'(\boldsymbol{q}', \dot{\boldsymbol{q}}')\dot{\boldsymbol{q}}' + \boldsymbol{g}'(\boldsymbol{q}') = \tilde{\boldsymbol{\tau}} \tag{1}$$

$$\boldsymbol{D}'(\boldsymbol{q}') = \begin{bmatrix} D_{11}' & 0 & D_{13}' & 0 \\ 0 & D_{22}' & 0 & D_{24}' \\ D_{31}' & 0 & D_{33}' & 0 \\ 0 & D_{42}' & 0 & D_{44}' \end{bmatrix} \tag{2}$$

$$\boldsymbol{C}'(\boldsymbol{q}', \dot{\boldsymbol{q}}') = \begin{bmatrix} h_1\dot{q}_3' & 0 & h_1(\dot{q}_1' + \dot{q}_3') & 0 \\ 0 & h_2\dot{q}_4' & 0 & h_2(\dot{q}_2' + \dot{q}_4') \\ -h_1\dot{q}_1' & 0 & 0 & 0 \\ 0 & -h_2\dot{q}_2' & 0 & 0 \end{bmatrix} \tag{3}$$

$$\boldsymbol{g}'(\boldsymbol{q}') = \bar{g}\begin{bmatrix} m_1 l_{c_1}\cos(q_1' + \gamma_1) + m_3 a_{s_1}\cos q_1' + m_3 l_{c_3}\cos(q_1' + q_3' + \gamma_3) \\ m_2 l_{c_2}\cos(q_2' + \gamma_2) + m_4 a_{s_2}\cos q_2' + m_4 l_{c_4}\cos(q_2' + q_4' + \gamma_4) \\ m_3 l_{c_3}\cos(q_1' + q_3' + \gamma_3) \\ m_4 l_{c_4}\cos(q_2' + q_4' + \gamma_4) \end{bmatrix} \tag{4}$$

The auxiliary terms in the aforementioned matrices, are given by

$$\begin{aligned} D_{11}' &= m_1 l_{c_1}^2 + I_1 + I_3 + m_3\left(a_{s_1}^2 + l_{c_3}^2 + 2a_{s_1}l_{c_3}\cos(q_3' + \gamma_3)\right) \\ D_{22}' &= m_2 l_{c_2}^2 + I_2 + I_4 + m_4\left(a_{s_2}^2 + l_{c_4}^2 + 2a_{s_2}l_{c_4}\cos(q_4' + \gamma_4)\right) \\ D_{31}' &= D_{13}' = m_3\left(l_{c_3}^2 + a_{s_1}l_{c3}\cos(q_3' + \gamma_3)\right) + I_3 \\ D_{42}' &= D_{24}' = m_4\left(l_{c_4}^2 + a_{s_2}l_{c4}\cos(q_4' + \gamma_4)\right) + I_4 \\ D_{33}' &= m_3 l_{c_3}^2 + I_3 \\ D_{44}' &= m_4 l_{c_4}^2 + I_4 \\ h_1 &= -m_3 a_{s_1} l_{c_3}\sin(q_3' + \gamma_3) \\ h_2 &= -m_4 a_{s_2} l_{c_4}\sin(q_4' + \gamma_4) \end{aligned} \tag{5}$$

Once the dynamic model of the free system is obtained, the closed chain constraint (6) must be solved, thus obtaining (7), where $q = [q_1, q_2]^T \in R^2$, is the

angular position vector of the constrained system and the auxiliary $\bar{Z}$ terms are given by (8). Notice in (7), that in order to obtain q_3', $q_4'(q_1, q_2)$ should be obtained first.

$$\boldsymbol{\phi}(\boldsymbol{q}') = \begin{bmatrix} a_{s_1}\cos(q_1') + a_{s_3}\cos\left(q_1'+q_3'\right) - a_{s_2}\cos(q_2') - a_{s_4}\cos\left(q_2'+q_4'\right) - a_{s_5} \\ a_{s_1}\sin(q_1') + a_{s_3}\sin\left(q_1'+q_3'\right) - a_{s_2}\sin(q_2') - a_{s_4}\sin\left(q_2'+q_4'\right) \end{bmatrix} = 0 \quad (6)$$

$$\begin{bmatrix} q_1' \\ q_2' \\ q_3' \\ q_4' \end{bmatrix} = \begin{bmatrix} q_1 \\ q_2 \\ \tan^{-1}\left(\frac{\bar{Z}_5 + a_{s_4}\sin\left(q_2+q_4'\right)}{\bar{Z}_4 + a_{s_4}\cos\left(q_2+q_4'\right)}\right) - q_1 \\ \tan^{-1}\left(\frac{\bar{Z}_2}{\bar{Z}_1}\right) + \tan^{-1}\left(\frac{\pm\sqrt{\bar{Z}_1^2+\bar{Z}_2^2-\bar{Z}_3^2}}{\bar{Z}_3}\right) - q_2 \end{bmatrix} \quad (7)$$

$$\begin{aligned} \bar{Z}_1 &= 2a_{s_4}\bar{Z}_4 \\ \bar{Z}_2 &= 2a_{s_4}\bar{Z}_5 \\ \bar{Z}_3 &= a_{s_3}^2 - a_{s_4}^2 - \bar{Z}_4^2 - \bar{Z}_5^2 \\ \bar{Z}_4 &= a_{s_2}\cos(q_2) - a_{s_1}\cos(q_1) + a_{s_5} \\ \bar{Z}_5 &= a_{s_2}\sin(q_2) - a_{s_1}\sin(q_1) \end{aligned} \quad (8)$$

Finally, the dynamic model of the closed chain robot is obtained as it is shown in (9),

$$\boldsymbol{\tau} = \boldsymbol{D}(\boldsymbol{q}')\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q}', \dot{\boldsymbol{q}}')\dot{\boldsymbol{q}} + \tilde{\boldsymbol{g}}(\boldsymbol{q}') \quad (9)$$

where

$$\begin{aligned} &\boldsymbol{D}(\boldsymbol{q}') = \boldsymbol{\rho}^{\boldsymbol{T}}(\boldsymbol{q}')\boldsymbol{D}'(\boldsymbol{q}')\boldsymbol{\rho}(\boldsymbol{q}') \in R^{2\times 2} \\ &\boldsymbol{C}(\boldsymbol{q}', \dot{\boldsymbol{q}}') = \boldsymbol{\rho}^{\boldsymbol{T}}(\boldsymbol{q}')\boldsymbol{C}'(\boldsymbol{q}', \dot{\boldsymbol{q}}')\boldsymbol{\rho}(\boldsymbol{q}') + \boldsymbol{\rho}^{\boldsymbol{T}}(\boldsymbol{q}')\boldsymbol{D}'(\boldsymbol{q}')\dot{\boldsymbol{\rho}}(\boldsymbol{q}', \dot{\boldsymbol{q}}') \in R^{2\times 2} \\ &\tilde{\boldsymbol{g}}(\boldsymbol{q}') = \boldsymbol{\rho}^{\boldsymbol{T}}(\boldsymbol{q}')\boldsymbol{g}'(\boldsymbol{q}') \in R^2 \end{aligned}$$

$$\boldsymbol{\rho}(\boldsymbol{q}') = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ -\frac{a_{s_1}\sin\left(q_1'-q_2'-q_4'\right)}{a_{s_3}\sin\left(q_1'-q_2'+q_3'-q_4'\right)} - 1 & -\frac{a_{s_2}\sin(q_4')}{a_{s_3}\sin\left(q_1'-q_2'+q_3'-q_4'\right)} \\ \frac{a_{s_1}\sin(q_3')}{a_{s_4}\sin\left(q_1'-q_2'+q_3'-q_4'\right)} & -\frac{a_{s_2}\sin\left(q_1'-q_2'+q_3'\right)}{a_{s_4}\sin\left(q_1'-q_2'+q_3'-q_4'\right)} - 1 \end{bmatrix}$$

$$
\begin{aligned}
\dot{\boldsymbol{\rho}}(\boldsymbol{q}') &= -\boldsymbol{\psi}_{\boldsymbol{q}'}^{-1}(\boldsymbol{q}')\dot{\boldsymbol{\psi}}_{\boldsymbol{q}'}(\boldsymbol{q}')\boldsymbol{\rho}(\boldsymbol{q}') \in R^{4\times 2} \\
\boldsymbol{\psi}_{q'}(\boldsymbol{q}') &\in R^{4\times 4} \\
\boldsymbol{\psi}_{\boldsymbol{q}'}(\boldsymbol{q}')(1,1) &= -a_{s_1}\sin\left(q_1'\right) - a_{s_3}\sin\left(q_1' + q_3'\right) \\
\psi_{\boldsymbol{q}'}(\boldsymbol{q}')(1,2) &= a_{s_2}\sin\left(q_2'\right) + a_{s_4}\sin\left(q_2' + q_4'\right) \\
\psi_{\boldsymbol{q}'}(\boldsymbol{q}')(1,3) &= -a_{s_3}\sin\left(q_1' + q_3'\right) \\
\psi_{\boldsymbol{q}'}(\boldsymbol{q}')(1,4) &= a_{s_4}\sin\left(q_2' + q_4'\right) \\
\psi_{\boldsymbol{q}'}(\boldsymbol{q}')(2,1) &= a_{s_1}\cos\left(q_1'\right) + a_{s_3}\cos\left(q_1' + q_3'\right) \\
\psi_{\boldsymbol{q}'}(\boldsymbol{q}')(2,2) &= -a_{s_2}\cos\left(q_1'\right) - a_{s_4}\cos\left(q_2' + q_4'\right) \\
\psi_{\boldsymbol{q}'}(\boldsymbol{q}')(2,3) &= a_{s_3}\cos\left(q_1' + q_3'\right) \\
\psi_{\boldsymbol{q}'}(\boldsymbol{q}')(2,4) &= -a_{s_4}\cos\left(q_2' + q_4'\right) \\
\psi_{\boldsymbol{q}'}(\boldsymbol{q}')(3,1) &= \psi_{\boldsymbol{q}'}(\boldsymbol{q}')(4,2) = 1 \\
\psi_{\boldsymbol{q}'}(\boldsymbol{q}')(3,2) &= \psi_{\boldsymbol{q}'}(\boldsymbol{q}')(3,3) = \psi_{\boldsymbol{q}'}(\boldsymbol{q}')(3,4) = \psi_{\boldsymbol{q}'}(\boldsymbol{q}')(4,1) = \psi_{\boldsymbol{q}'}(\boldsymbol{q}')(4,3) \\
&= \psi_{\boldsymbol{q}'}(\boldsymbol{q}')(4,4) = 0
\end{aligned}
$$

In (9), $\boldsymbol{\tau} = [\tau_1, \tau_2]^T \in R^2$ represents the applied generalized force vector. It is important to mention that the dynamic model (9), presents several kinematic singularity configurations (Tsai 1999). The motion of the parallel robot towards such configurations should be avoided, as the robot might for instance, lose degrees of freedom. In this paper the region where singularity configurations occurr is referred to as singularity region or discontinuous region. Therefore the parallel robot model is defined on a compact set Ω in the independent generalized coordinates $\boldsymbol{q}$, where closed-chain constraints (6) are satisfied and singularities in the dynamic model are avoided. Moreover, all control laws for open chain robots could also be applied with the only restriction that the guaranteed (Lyapunov) stability conclusion will be local (Ghorbel et al. 2000).

2.2 Closed loop system

One of the objectives of the synergetic design approach is to get the best performance from simple controllers. Therefore, despite of the development of many other control strategies, PID controllers are employed to control the parallel robot. Note that assuming the input to the robot as the output of the PID controller, the control input is being parameterized in terms of the steady gains of the PID controller.

The closed loop system of the dynamic model (9) and its PID controller is given in (10), where $\boldsymbol{x} = \left[q_1, q_2, \dot{q}_1, \dot{q}_2, \int_0^{t_f} e_1 dt, \int_0^{t_f} e_2 dt\right]^T = [x_1, \ldots, x_6]^T \in R^6$ and $\bar{x} \in R^6$ are the current and desired state variable vectors, respectively, $\boldsymbol{e} = [e_1, e_2]^T = [\bar{x}_1 - x_1, \bar{x}_2 - x_2]^T$ and $\dot{\boldsymbol{e}} = [\dot{e}_1, \dot{e}_2]^T = [\bar{x}_3 - x_3, \bar{x}_4 - x_4]^T$ are the angular position and velocity error vectors of the actuated angles, respectively, t is the time, t_f is the

final time and $\bar{\boldsymbol{u}} = [\bar{u}_1, \bar{u}_2]^T$ is the input vector as defined in (11), with k_{p_i}, k_{i_i} and k_{d_i} the proportional, integral and derivative constant gains respectively.

$$\dot{\boldsymbol{x}} = \frac{d\boldsymbol{x}}{dt} = \begin{bmatrix} x_3 \\ x_4 \\ \boldsymbol{D}^{-1}(\boldsymbol{q}')\left(-\boldsymbol{C}(\boldsymbol{q}', \dot{\boldsymbol{q}}')[x_3, x_4]^T - \boldsymbol{g}(\boldsymbol{q}') + \bar{\boldsymbol{u}}\right) \\ e_1 \\ e_2 \end{bmatrix} \tag{10}$$

$$\bar{u}_i = \tau_i = k_{p_i} e_i + k_{i_i} \int_0^{t_f} e_i dt + k_{d_i} \dot{e}_i \quad \text{for} \quad \text{i} = 1, 2. \tag{11}$$

2.3 Design parameters

From (4) and (5), it is clear that the robot closed loop dynamics depends on link inertial parameters $m_i, l_{c_i}, \gamma_i, I_i$, which in turn depend on the link shape. Typically, after m_i, l_{c_i}, γ_i, and I_i are determined, a shape design takes place in order to find the link geometry (Tu and Rastegar 1993). Nevertheless, in order to obtain the geometric parameters of the link in a single design stage, we assume the link geometry shown in Fig. 2. Thus, both inertial and kinematic link parameters depend on geometry parameters $(a_{s_i}, b_{s_i}, c_{s_i}, d_{s_i}, e_{s_i}, f_{s_i}, g_{s_i}, h_{s_i}, i_{s_i}, j_{s_i}, k_{s_i})$. Therefore, the geometric parameters of all links, excepting the fixed base link (only its length is taken into account), are grouped into the vector of structural design parameters.

In order to find the mathematical expression of inertial and kinematic parameters for each link in terms of its geometric parameters, it is necessary to regard the following:

- The reference coordinate system of the i-th link is placed at the center of the C_i hole as it is shown in Fig. 2.
- The i-th link only can move in the xy plane.
- There is a shaft in the B_i hole at links 1, 2 and 3, and there are two shafts that couple the motors with the C_i hole of the links 1 and 2 respectively (see Fig. 3).
- Shafts are made of steel, and links are made of aluminium. The density of the aluminium and the steel are $\rho_{al} = 2,710\ \frac{\text{kg}}{\text{m}^3}$ and $\rho_{st} = 7,840\ \frac{\text{kg}}{\text{m}^3}$, respectively.
- The i-th mass is concentrated at the mass center of the i-th link.
- The links are divided into simple geometries in order to simplify the computing of their inertial parameters. As it is shown in Fig. 2, the i-th link is divided in eleven geometric shapes. A_i, D_i, and E_i have rectangular prism shape, B_i, C_i, J_i, and K_i have cylindrical shape and F_i, G_i, H_i, and I_i have triangular prism shape.

The relationship between inertial parameters and geometric parameters can be found by following classical references, such as (Hibbeler 2003).

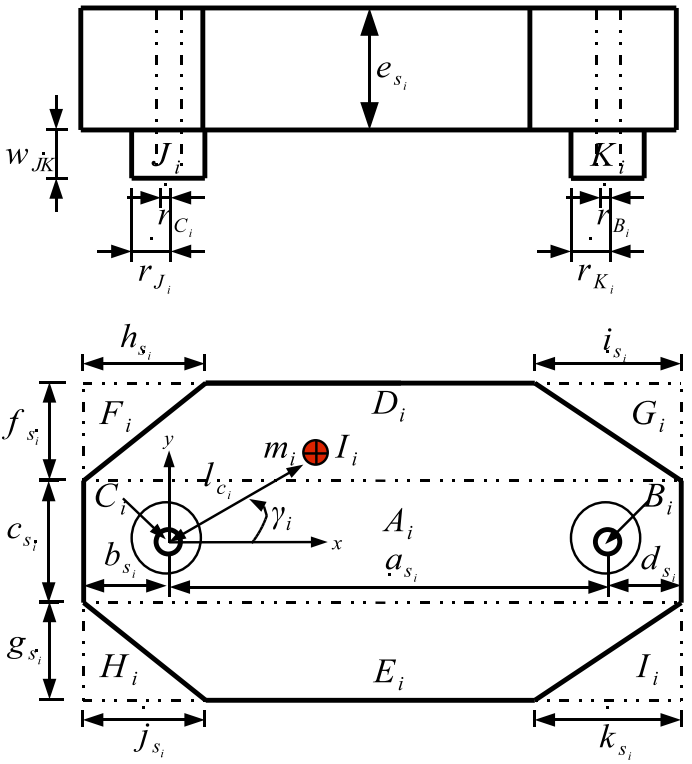

Fig. 2 Schematic representation of the top view (*above*) and the front view (*below*) of a link

The vector of control design parameters consists of the proportional, integral and derivative gains of the controller for each motor. Thus the full vector of design parameters $\boldsymbol{p}$ is given by (12), where $\boldsymbol{p_s}$ and $\boldsymbol{p_c}$ are given by (13) and (14), respectively.

$$\boldsymbol{p} = [\boldsymbol{p_s}, \boldsymbol{p_c}]^T \in R^{51} \tag{12}$$

$$\begin{aligned} \boldsymbol{p_s} = [&a_{s_1} \ldots a_{s_5}, b_{s_1} \ldots b_{s_4}, c_{s_1} \ldots c_{s_4}, d_{s_1} \ldots d_{s_4}, e_{s_1} \ldots e_{s_4}, f_{s_1} \ldots f_{s_4}, \\ &g_{s_1} \ldots g_{s_4}, h_{s_1} \ldots h_{s_4}, i_{s_1} \ldots i_{s_4}, j_{s_1} \ldots j_{s_4}, k_{s_1} \ldots k_{s_4}]^T \in R^{45} \end{aligned} \tag{13}$$

$$\boldsymbol{p_c} = [k_{p_1}, k_{p_2}, k_{i_1}, k_{i_2}, k_{d_1}, k_{d_2}]^T \in R^6 \tag{14}$$

2.4 Design objectives

The main purpose for which a robot is built, is to perform a task, which is commonly defined as trajectory tracking in Cartesian space. Therefore, one of the

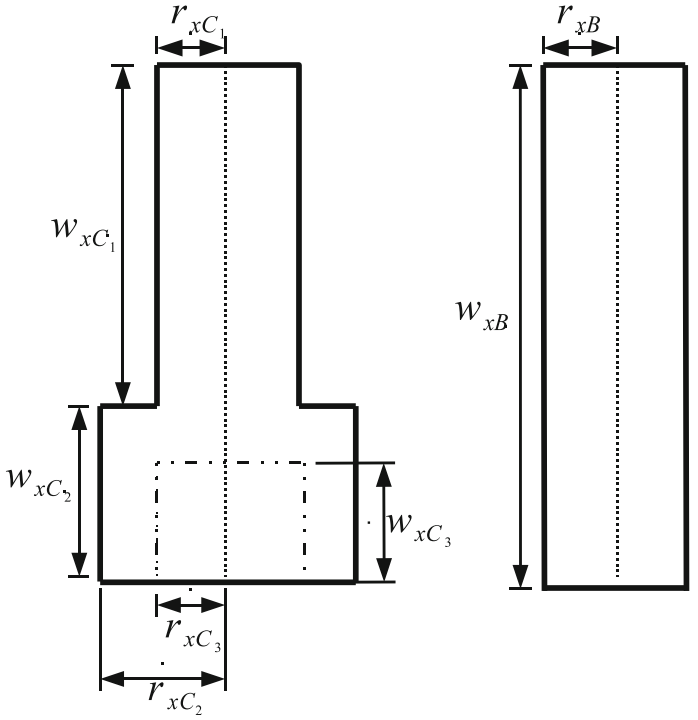

Fig. 3 Schematic representation of the B hole shaft (*right* figure) and the shaft that couples the motor with the C_i hole (*left* figure)

design objectives considered in this work is to minimize the error between the desired trajectory and the actual trajectory followed by the robot. A circle in the Cartesian space with a radius of 0.1 m and center in (0.15 m, 0.30 m) is proposed as the desired trajectory. As the desired trajectory is time-varying, we are interested in minimizing the tracking error over the entire period of time the robot is performing the task. Therefore, for each actuated joint a performance index, $\bar{J}_i$, is defined as (15).

$$\bar{J}_i : \bar{J}_i(\boldsymbol{x}, \bar{\boldsymbol{x}}, \boldsymbol{p}) = \int_0^{t_f} e_i^2(t)dt \quad \text{for} \quad i = 1, 2 \tag{15}$$

On the other hand, the performance of the robot must be the same for all of the robot kinematic configurations. However, it is well known that the robot under study may have different kinematic singularities (Liu et al. 2006), which must be avoided in order not to lose its kinematic characteristics. For that purpose, several robot design objectives have been proposed (Asada 1983; Yoshikawa 1985; Ma and Angeles 1990). Hence, other design objective to be considered, is the maximization of the manipulability over the entire period of time the robot is performing the task, which

is given in (16), where $\mathbf{J}$ is the robot Jacobian matrix given by (17), $\det(\mathbf{A})$ and $tr(\mathbf{A})$ represent the determinant and the trace of matrix $(\mathbf{A})$, respectively.

$$\bar{J}_3 = -\frac{2}{t_f}\int_0^{t_f} \frac{\sqrt{\det\left(\mathbf{J}^{-1}\mathbf{J}^{-T}\right)}}{tr\left(\mathbf{J}^{-1}\mathbf{J}^{-T}\right)}dt \tag{16}$$

$$\mathbf{J} = \begin{bmatrix} -\frac{a_{s_1}\left(\sin\left(q'_2+q'_4\right)\right)\sin q'_3}{\sin\left(q'_1-q'_2+q'_3-q'_4\right)} & \frac{a_{s_2}\left(\sin\left(q'_1+q'_3\right)\right)\sin q'_4}{\sin\left(q'_1-q'_2+q'_3-q'_4\right)} \\ \frac{a_{s_1}\left(\cos\left(q'_2+q'_4\right)\right)\sin q'_3}{\sin\left(q'_1-q'_2+q'_3-q'_4\right)} & -\frac{a_{s_2}\left(\cos\left(q'_1+q'_3\right)\right)\sin q'_4}{\sin\left(q'_1-q'_2+q'_3-q'_4\right)} \end{bmatrix} \tag{17}$$

2.5 Design constraints

Two different kinds of constraints are recognized in the design problem addressed in this work. Those that are explicit constraints on design parameters and those that are explicit constraints on time variables, as for instance the input torque to the robot.

Constraints on design parameters The five-link planar robot has two degrees of freedom, so to ensure that the actuated links behave as cranks, a complete rotation of these links must be ensured, this is achieved if the five-bar Grashof criteria (18) are fulfilled (Ting 1986).

$$\begin{aligned} a_{s_5} + a_{s_1} + a_{s_2} - a_{s_3} - a_{s_4} &< 0 \\ a_{s_1} + a_{s_2} - a_{s_3} &< 0 \\ a_{s_1} + a_{s_2} - a_{s_4} &< 0 \\ a_{s_4} - a_{s_5} &< 0 \\ a_{s_3} - a_{s_5} &< 0 \end{aligned} \tag{18}$$

There are upper and lower limits in each of the structural parameters of the system, which must be met in order to have a feasible design; those limits are given by (19)–(20), where upper and lower limits are depicted in Table 1.

$$a_{s_{low}} \leq a_{s_l} \leq a_{s_{upp}} \quad \text{for } l = 1,2,3,4,5 \tag{19}$$

$$v_{s_{l_{low}}} \leq v_{s_l} \leq v_{s_{l_{upp}}} \quad \text{for} \quad v = b,c,\ldots,k \quad \text{and} \quad l = 1,2,3,4 \tag{20}$$

Constraints on time variables The torque that motors can provide is bounded by an upper limit, such constraints are given by (21), where $u_{i_{upp}}$ represents the torque upper limit for motor i.

$$\begin{aligned} |\bar{u}_1(t)| - u_{1_{upp}} &< 0 \\ |\bar{u}_2(t)| - u_{2_{upp}} &< 0 \end{aligned} \tag{21}$$

2.6 Dynamic optimization problem statement

The integrated structure–control design of the parallel robot addressed in this work, is stated as a dynamic optimization problem that reads as follows. Find the design

Table 1 Upper and lower limits for links structural parameters

Str. Parameter	Upper limit (m)	Lower limit (m)
a_{s_l} for $l = 1, \ldots, 5$	0.5	0.35
b_{s_l} for $l = 1, 2$	0.3	0.01905
b_{s_l} for $l = 3, 4$	0.3	0.015
c_{s_l} for $l = 1, 2$	0.1	0.0381
c_{s_l} for $l = 3, 4$	0.1	0.03175
d_{s_l} for $l = 1, \ldots, 4$	0.3	0.015
e_{s_l} for $l = 1, \ldots, 4$	0.03	0.00635
f_{s_l} for $l = 1, \ldots, 4$	0.1	0
g_{s_l} for $l = 1, \ldots, 4$	0.1	0
h_{s_l} for $l = 1, \ldots, 4$	$b_{s_l} + a_{s_l} + d_{s_l}$	0
i_{s_l} for $l = 1, \ldots, 4$	$b_{s_l} + a_{s_l} + d_{s_l} - h_{s_l}$	0
j_{s_l} for $l = 1, \ldots, 4$	$b_{s_l} + a_{s_l} + d_{s_l}$	0
k_{s_l} for $l = 1, \ldots, 4$	$b_{s_l} + a_{s_l} + d_{s_l} - j_{s_l}$	0

parameters (12), that minimize the design objectives (15) and (16), subject to the differential equation modeling the closed loop robot dynamics (10), as well as, inequality constraints on the structure parameters (18), (19)–(20) and on the actuator torque (21).

Although the above mentioned problem is a multi-objective one, the weighted sum approach is employed, thus it can be considered as a single objective one. To this end, we propose the functional (22), where weights w_i of appropriate units, are employed only to normalize each of the objectives $\bar{J}_i$, thus ensuring both, each objective $\bar{J}_i$ has the same importance and the functional $\bar{J}$ is unitless.

$$\bar{J} : \bar{J}(\boldsymbol{x}, \bar{\boldsymbol{x}}, \boldsymbol{p}) = \sum_{i=1}^{3} w_i \bar{J}_i \tag{22}$$

2.7 Singularity regions

Singularity configurations of the parallel robot are generated by particular Cartesian positions of the end-effector, for which parallel robots lose their inherent infinite rigidity and the end-effector will have uncontrollable degrees of freedom (Merlet 2001). Singularity configurations of the parallel robot make that singularity regions appear in the unfeasible and the feasible regions of the design space. In the singularity regions neither the dynamic model, nor the Jacobian matrix can be computed. Hence, the problems of using gradient based techniques arise when the proposed initial condition is inside the singularity region, or when the search direction goes to a singularity region since the gradient can not be computed in order to get the next search direction.

Let $\boldsymbol{p} \in \Omega \in R^{51}$ and defining the subspace $\Omega' = \{\boldsymbol{p} \in \Omega : \det(\mathbf{J}^{-1}) = \mathbf{0}, \sin(q'_1 - q'_2 + q'_3 - q'_4) = 0\} \subset \Omega$, the singularity regions are determined when $\boldsymbol{p} \in \Omega'$.

3 Hybrid optimization algorithm

Typically, a dynamic optimization problem is solved by using the variational approach (Bryson 1999) or by applying a discretization (Biegler 2010) transforming the original continuous time problem into a discrete problem, such as the sequential or direct approach. In this paper, the control input is parametrized using the PID control law, hence, the problem stated in the previous section can be considered as a NLP problem with dynamic equality constraints, which can be handled by explicitly solving the dynamic model of the robot. This fact implies that, if an EA is used to solve the problem, then, for each individual a simulation of the dynamic model of the robot should be performed. However, if a gradient based technique is used to solve the problem, then for each step, besides solving the dynamic model of the robot, the system state sensitivities with respect to the design parameters should also be obtained. However, in singularity regions neither the dynamic model of the robot nor the Jacobian matrix can be computed. Hence, problems would arise when initial conditions—for a gradient based technique—are within a singularity region, since neither the gradient nor the sensitivity can be computed in order to get the next search direction. Therefore, in this work a hybrid optimization algorithm is used to solve the problem at hand. The hybrid algorithm consists on using an exploratory search mechanism to search in the entire space of solution and finding a warm solution. After that, the quality of the solution is improved by a fine search mechanism. The exploratory search mechanism consists on using an EA (differential evolution (DE) algorithm (Price et al. 2005)) and the fine search mechanism consists on using a gradient based algorithm (SQP algorithm (Betts 2010)). The hybrid optimization algorithm is detailed below.

Key parameters The key parameters of the hybrid optimization algorithm are: *NP*—the population size that is the set of individuals, G_{Max}—the generation where the DE algorithm is changed by the SQP algorithm, *TolFun*—the minimum allowed difference between the performance index in the iteration i and the performance index in the iteration $i-1$ to stop the SQP algorithm, *IterMax*—the maximum allowed iteration to stop the SQP algorithm, *G*—the generation or iteration in the algorithm, *CR* and *F* for the DE algorithim—the crossover constant that controls the influence of the parent in the generation of the offspring (higher values mean less influence of the parent) and the weight applied to the influence of two of the three individuals selected at random in order to generate the offspring (scaling factor), respectively. The key parameters procedure is presented in row 2 of Fig. 5.

Initialization Let w_G^i be the individual of the i-th population in the G-th generation. The initialization consists on randomly selecting the individuals for the first generation and evaluating the dynamic performance function and the static and dynamic constraints of them. This is shown in rows 5 and 6 of Fig. 5.

Evaluation of the performance function and dynamic constraints The dynamic optimization problem requires the solution of the nonlinear differential equation (10) to provide the dynamic behavior of the closed loop system and to compute the performance function. Therefore, the following procedure is carried out: (i) divide the time interval $[t_0, t_f]$ into n time stages, each of length dt, i.e. $t = t_0 = 0$, $t_1 =$

$t_0 + dt, \ldots, t_f = t_n$. (ii) Choosing an initial state vector $\mathbf{x}(\mathbf{t_0})$, the nonlinear differential equation is solved from $t = 0$ to t_f to generate the state trajectory $\mathbf{x}(\mathbf{t_k}) \ \forall \ k = 1, \ldots, n$. A Runge Kutta method is used in this paper to solve the nonlinear differential equation. (iii) At each time stage, the performance function (15) from t_0 to t_k is computed and the dynamic constraints are evaluated. If there exist a singularity configuration in the parallel robot for the time interval $[t_{k-1}, t_k]$, a flag would be activated. Such a flag indicates that the algorithm must stop the evaluation of the performance index of the k-th individual and continue with the $(k+1)$-th individual. A counter "FNE", counts up the times that the flag is activated. Another counter "FE", counts up the times that the performance function is evaluated. (iv) To count the times that the dynamic and static constraints are unfeasible and keep it in a variable.

It is important to remark that the use of the flag reduces the time that the algorithm spends to get an optimal solution. Hence, once the flag is activated, the solution of the nonlinear differential equation (dynamic behavior of the robot) for the individual in a singularity configuration is not computed for the next time stages. The details are presented in Fig. 4 and this must be included in rows 6 and 18 in Fig. 5.

3.1 Exploratory search mechanism

Mutation and crossover Once the individuals of the first population are initialized and evaluated, the population is mutated and crossed to produce a population of NP trial vector named as $u^i_{G+1} \forall i = 1, \ldots, NP$. Uniform crossover is employed in the algorithm and the details are presented in rows 11–17 in Fig. 5.

```
1   Begin
2     Choose dt, t_f and x(t_0)
3     Flag=0
4     n = t_f / dt
5     For k = 0 to n Do
6       k_1 = f(x(t_k), t_k)
7       k_2 = f(x(t_k) + dt k_1/2, t_k + dt/2)
8       k_3 = f(x(t_k) + dt k_2/2, t_k + dt/2)
9       k_4 = f(x(t_k) + dt k_3, t_k + dt)
10      x(t_{k+1}) = x(t_k) + dt/6 (k_1 + 2k_2 + 2k_3 + k_4)
11      t_{k+1} = t_k + dt
12      Integration= (J̄(w^i_G, t_k) + J̄(w^i_G, t_{k+1}))dt
13      Evaluate g(x^i_G, t_k) h(x^i_G, t_k)
14      Count the unfeasible dynamic constraint in g(x^i_G, t_k) and h(x^i_G, t_k)
15      If there exist a singularity configuration then
16        Flag=1
17        FNE=FNE+1
18        Break the For loop
19      End If
20    End For
21    If Flag == 0
22      Sum the unfeasible static constraint to the sum of the unfeasible dynamic constraint.
23      FE=FE+1
24    End If
25  End
```

Fig. 4 Evaluation mechanism of the performance function and constraints for dynamic systems added to the algorithm

```
1    Begin
2      Key parameters, FNE=0, FE=0
3      Begin
4        G = 0
5        Create a random initial internal population w^i_G  ∀i, i = 1, ..., NP
6        Evaluate J̄(w^i_G, t), g(w^i_G, t), h(w^i_G, t) ∀i, i = 1, ..., NP, ∀t, t = 1, ..., t_f
7        Do
8         For i = 1 to NP Do
9           Select randomly r1 ≠ r2 ≠ r3
10          j_rand =randint(1, D)
11          For j = 1 to D Do
12            If (rand_j[0, 1) < CR or j = j_rand) Then
13              u^i_{j,G+1} = x^{r1}_{j,G} + F(x^{r2}_{j,G} − x^{r3}_{j,G})
14            Else
15              u^i_{j,G+1} = x^i_{j,G}
16            End If
17          End For
18          Evaluate J̄(u^i_{G+1}, t), g(u^i_{G+1}, t), h(u^i_{G+1}, t) ∀t, t = 1, ..., t_f
19          If (u^i_{G+1} is better than w^i_G (based on selection criteria)) Then
20            w^i_{G+1} = u^i_{G+1}
21          Else
22            w^i_{G+1} = x^i_G
23          End If
24        End For
25        G = G + 1
26       While (G <= G_Max)
27       Sort J̄(x^i_G, t) ∀i, i = 1, ..., NP in ascending order
28       SQP algorithm with x^1_G as the initial condition
29   End
```

Fig. 5 Hybrid algorithm. randint(min, max) is a function that returns an integer number between min and max. rand[0,1) is a function that returns a real number between 0 and 1. Both are based on a uniform probability distribution

Selection with a constraint-handling mechanism In this part of the algorithm, the trial vector $\mathbf{u^i_{G+1}}$ is compared with the target vector w^i_G and the better solution passes to the next generation. A constraint-handling approach similar (but not equal) to (Deb et al. 2002; Deb and Kain 2003) is included in the selection process, where two solutions (trial and target vector) are picked from the population and the better solution is chosen considering the following rules:

- Between two feasible individuals, the one with a higher fitness is chosen.
- A feasible individual is chosen over an infeasible one or over an individual within a singularity region (flag = activated).
- Between two infeasible individuals, the one with the lowest amount of constraint violation is chosen.
- A infeasible individual is chosen over an individual within a singularity region.
- Between two individuals within a singularity region (flags = activated), the individual is chosen by a random selection.

The selection with a constraint-handling mechanism is shown in rows 19–23 in Fig. 5.

Once the selection process is done, the mutation, cross and selection are repeated, but now with the new population generated in the previous generation.

3.2 Fine search mechanism

The mechanism to finely search the optimum solution consists on sorting the G_{Max}-th population and choosing the best individual of that population. When the population presents infeasible individuals the sort is done by using the following criteria (Mezura and Coello-Coello 2005):

- If the population has at least a feasible individual, the best individual would be the feasible individual with the highest fitness.
- If all individuals of the population are infeasible, the best individual would be the infeasible individual with the lowest amount of constraint violation. Individuals with singularity regions presents the highest amount of constraint violation.

Once the best individual of the G_{Max}-th population is selected, this individual will be chosen as the initial condition for the SQP algorithm. The fine search mechanism is presented in rows 27 and 28 of Fig. 5.

The SQP algorithm allows to closely mimic Newton's method for constrained optimization just as it is done for unconstrained optimization. At each major iteration, an approximation is made of the Hessian of the Lagrangian function using a quasi-Newton updating method. This is then used to generate a QP subproblem whose solution is used to form a search direction for a line search procedure. The general algorithm is stated as follows: given the problem description (23):

$$\min_{\boldsymbol{p}} \tilde{J}(\boldsymbol{p}) \tag{23}$$

subject to:

$$\begin{aligned} g_j(\boldsymbol{p}) \leq 0, \quad & j = 1, \ldots, n_g \\ h_k(\boldsymbol{p}) = 0, \quad & k = 1, \ldots, n_h \end{aligned}$$

The main idea is the formulation of a quadratic programming (QP) subproblem based on a quadratic approximation of the Lagrangian function (24).

$$L(\boldsymbol{p}, \boldsymbol{\lambda}) = \tilde{J}(\boldsymbol{p}) + \sum_{j=1}^{n_g} \lambda_j \cdot g_j(\boldsymbol{p}) + \sum_{k=1}^{n_h} \lambda_k \cdot h_k(\boldsymbol{p}) \tag{24}$$

Therefore, the QP subproblem (25) is obtained by linearizing the nonlinear constraints, where the matrix $\boldsymbol{H}_i = \nabla^2 L$ is a positive definite approximation of the Hessian matrix of the Lagrangian function (24). $\boldsymbol{H}_i$ is updated by the Broyden–Fletcher–Goldfarb–Shanno (BFGS) method. The gradient calculation is obtained by using sensitivity equations. Hence, if $\boldsymbol{d}_i$ solves the subproblem given in (25) and $\boldsymbol{d}_i = 0$, then the parameter vector $\boldsymbol{p}$ is an optimal solution to the original problem. Otherwise, we set $\boldsymbol{p}^{i+1} = \boldsymbol{p}^i + \alpha_i \boldsymbol{d}_i$ and with this new vector the process is done again. The step length parameter α_i is determined by a linear search procedure so that a sufficient decrease in the performance index is obtained.

$$\underset{\boldsymbol{p}\in n_p}{Min}\frac{1}{2}\boldsymbol{d}^T H_i \boldsymbol{d} + \nabla\tilde{J}(\boldsymbol{p}^i)^T\boldsymbol{d} \tag{25}$$

subject to:

$$\nabla g_j(\boldsymbol{p}^i)^T\boldsymbol{d} + g_j(\boldsymbol{p}^i) \le 0, \quad j = 1,\ldots,n_g$$
$$\nabla h_k(\boldsymbol{p}^i)^T\boldsymbol{d} + h_k(\boldsymbol{p}^i) \le 0, \quad k = 1,\ldots,n_h$$

The QP subproblem requires the gradient calculation, therefore the sensitivity equations must be solved. The number of sensitivity equations is the number of state variables of the system times the number of the design variables of the design vector $\boldsymbol{p}$. For the particular case of the problem at hand, the number of sensitivity equations is 306. Additionally, it is necessary to solve fifty one gradient equations of the objective function and fifty one gradient equations for each constraint. In the case of dynamic constraints, the gradient equation must be computed n times for each dynamic constraint.

The gradient calculation of the objective function is obtained by solving the ordinary differential equations of the sensitivity equations, both stated by (26) and (27), respectively, where L_i is the i-th term of the objective function, $\boldsymbol{f}$ is the dynamic model of the parallel robot, $\boldsymbol{x}$ is the state vector, $p(j)$ is the j-th element of the vector of the design variables $\boldsymbol{p}$ and t is the time variable.

$$\frac{\partial\tilde{J}}{\partial p(j)} = \sum_{i=1}^{n_J}\int_0^{t_f}\left(\frac{\partial L_i}{\partial\boldsymbol{x}}\left[\frac{\partial\boldsymbol{x}}{\partial p(j)}(t)\right] + \frac{\partial L_i}{\partial p(j)}\right)dt \tag{26}$$

$$\frac{d}{dt}\left[\frac{\partial\boldsymbol{x}}{\partial p(j)}\right] = \frac{\partial\boldsymbol{f}}{\partial\boldsymbol{x}}\left[\frac{\partial\boldsymbol{x}}{\partial p(j)}\right] + \frac{\partial\boldsymbol{f}}{\partial p(j)} \tag{27}$$

3.3 Optimization design results

The initial parameters of the hybrid algorithm are set as follows: (1) The population size NP consists of 100 individuals. (2) The scaling factor F and the crossover constant CR are randomly generated at each generation, taking values from the intervals $0.3 \le F \le 0.9$ and $0.8 \le CR \le 1$ at each optimization process. (3) Different maximum number of generations are proposed ($G_{Max} = 100, 200, 300, \ldots,$ 1,000, 1,500, 1,600, 1,800, 3,000, 7,200), the parameters $TolFun = 1\text{E}{-}10$ and $IterMax = 50$ are chosen. In order to solve the nonlinear differential equations of the system dynamics (10) by the Runge Kutta method, the initial state vector $x(0) = [1.7453,\ 0,\ 0,\ 0,\ 0,\ 0]^T$, the final time $t_f = 5$ s and the integration step $dt = 5$ ms are selected.

Results of the hybrid algorithm for different values of G_{Max} (generation number where the hybrid algorithm switches from the expanded search mechanism to the fine search mechanism) are shown in Table 2, where FE represents the number of times that the performance function of an individual (solution) is evaluated. FNE

represents the number of times that the performance function of an individual is not evaluated because the individual is within a singularity region. The table is divided into three main columns. The results of the exploratory search mechanism are shown in the first main column. The second main column presents the performance function evaluated at the best individual $\bar{J}$ in the generation G_{Max}, it is also indicated if the individual is fesible (F) or unfeasible (U). The best individual in the G_{Max} generation is the initial condition for the fine search mechanism. The results for the fine search mechanism are shown in the third main column for each established G_{Max}.

As a flag is established to reduce the convergence time of the hybrid algorithm, the performance functions for several individuals are not computed. This can be observed in the FNE column. Once all individuals in the population are feasible (what hapens after 479 generations, for $G_{Max} \geq 500$), the exploratory search mechanism finds less individuals within a singularity region (FE $>$ FNE), as it is shown in the first main column of Table 2. It indicates that the mutation and crossover of feasible individuals provide better individuals than the mutation and crossover of infeasible individuals. Hence it is observed that the selection of the G_{Max} variable is an important factor to find the best solution with less computational effort for the problem at hand. Infeasible individuals in the fine search mechanism provide divergence to the hybrid algorithm (see the third main column when $G_{Max} \leq 400$) because in the process of finding the optimal solution, the search direction goes to a singularity region. Feasible individuals in the fine search mechanism provide convergence to the algorithm in spite of finding singularity regions in the search of the solution (see FNE when $G_{Max} \geq 500$ in the fine search mechanism of the third main column). Several local optimum solution could be found if the G_{Max} variable is not carefully chosen (different performance indexes $\bar{J}$ result with different G_{Max} variable). Hence, this behavior shows a high sensitivity of the fine search mechanism to the starting condition because it must be carefully chosen in order to reach a solution. A large value of the G_{Max} variable provides a large computational time and it does not always provide a better performance index, as it is observed for $G_{Max} \in [1,600, 1,800, 3,000, 7,200]$. In addition, when the starting condition of the fine search mechanism is not within a singularity region but it is unfeasible (see the third column when $G_{Max} \leq 400$), the algorithm diverges due to the convergence to a singularity region. In this particular problem, in order to get the optimum solution, the G_{Max} variable may be chosen as $G_{Max} \geq 1,600$ according to the empirical results. Nevertheless, the best optimum solution without requiring a large computational time is found when $G_{Max} = 1,600$.

The value of the mean performance index ($\bar{J}_{mean}$) evaluated at the best individual of each generation in the exploratory search mechanism is shown in Fig. 6a, for $G_{Max} = 1{,}600$. In addition in Fig. 6b, the mean of number of unfulfilled constraints (VC_{mean}) at each generation is displayed also for $G_{Max} = 1,600$. In generations previous to the left side of the dashed vertical line, the population has at least one unfeasible individual. It is observed on this side (Fig. 6b), that the constraint-handling mechanism selects the individuals with less unfulfilled constraints,

Table 2 Results of the hybrid algorithm for different G_{Max} parameter

Exploratory search mechanism				P. index	Fine search mechanism				
Comp. time [h]	G_{Max}	FE	FNE	$\bar{J}_i$	Comp. time [h]	Iter. number	$\bar{J}^*$	FE	FNE
0.4694	100	3,240	6,860	−0.4876 (U)	Div.	–	–	–	–
1.1028	200	8,199	11,901	−0.4715 (U)	Div.	–	–	–	–
1.8460	300	14,090	16,010	−0.8711 (U)	Div.	–	–	–	–
2.6175	400	2,050	19,600	−0.7778 (U)	Div.	–	–	–	–
3.4058	500	26,934	23,166	−0.8665 (F)	0.3690	43	−0.9000	2,670	2
4.3645	600	35,834	24,266	−0.8897 (F)	0.4973	49	−0.9129	2,678	3
5.3792	700	45,745	24,355	−0.8995 (F)	0.1017	12	−0.9146	2,657	1
6.3991	800	55,668	24,432	−0.9072 (F)	0.3073	44	−0.9177	2,662	0
7.4160	900	65,620	24,480	−0.9100 (F)	0.2489	50	−0.9179	2,675	0
8.4314	1,000	75,560	24,540	−0.9121 (F)	0.2532	50	−0.9178	2,684	0
13.5192	1,500	125,498	24,602	−0.9169 (F)	0.2616	50	−0.9179	2,703	4
14.5372	1,600	135,494	2,4606	−0.9172 (F)	0.2517	50	−0.9281	2,684	0
16.5728	1,800	155,487	2,4621	−0.9176 (F)	0.2503	50	−0.9281	2,663	0
28.7524	3,000	275,431	24,669	−0.9181 (F)	0.2614	50	−0.9281	2,670	0
71.2626	7,200	695,280	24,820	−0.9181 (F)	0.3199	50	−0.9281	3,037	25

regardless the value of the performance function that the individuals generate. On the other hand, in the subsequent generation at the right side of the dashed vertical line, all individuals of the population are feasible, there, the performance function is minimized because the constraint-handling mechanism selects the individuals with the best performance function considering individuals without constraint violations. The dashed vertical line is on the generation number 479. Hence, in the first 478 generations, the algorithm does not find feasible solutions for all individuals of the population.

The main advantage of the hybrid algorithm is that it always works with several initial design vector (called population) generated at random. In this way, the sensitivity of the algorithm to the provided initial point is not a problem and the exploration of the entire solution space is done. In addition, the fine search mechanism exhaustively explores the local search space of the local solution found by the exploratory search mechanism, such that the found solution is improved and it provides a better final solution, as it is observed in the third main column of Table 2. The hybrid algorithm spends 14.7889 h to find the best optimum solution for $G_{Max} = 1,600$.

As expected, a hybrid gradient-evolutionary optimization algorithm provides a better performance than a gradient or an evolutionary algorithm. In addition, the hybrid gradient-evolutionary optimization algorithm could be very useful in highly nonlinear dynamic optimization problems where singularity regions are presented within the search space.

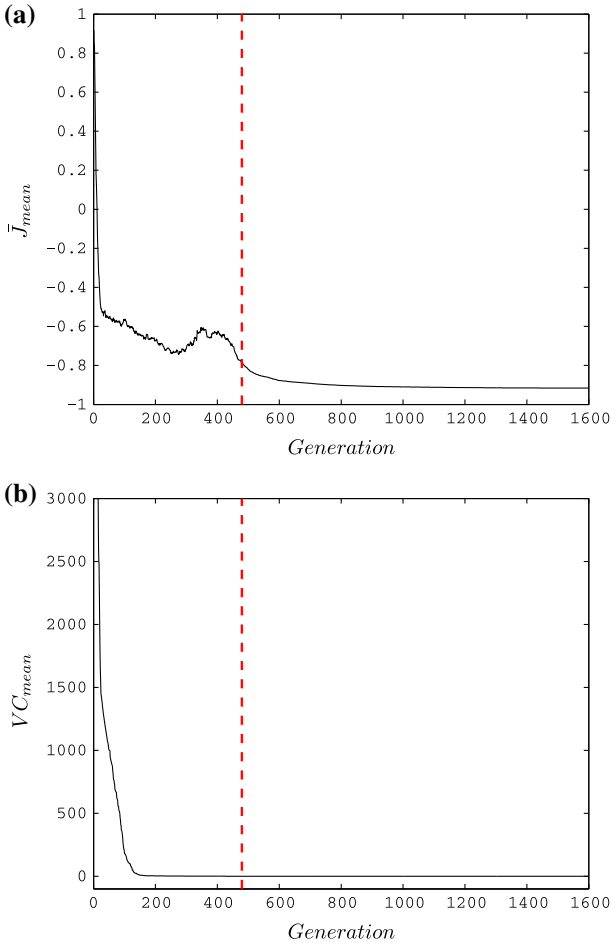

Fig. 6 Performance of the fist part of the algorithm (exploratory search algorithm): **a** behavior of the performance index, **b** behavior of the violated constraints (VC)

4 Comparative results of synergetic structure–control design approach and sequential design approach

In this section the synergetic structure–control design approach is compared with a sequential design approach. The idea here is to design the same planar parallel robot with its control system, and to formulate a sequential design approach using the same design variable vector, the same performance functions and constraints as stated in previous sections.

The proposed sequential design approach firstly designs the kinematic parameters of the parallel robot considering a static design performance (called in this paper as kinematic design). Then, the dynamic performance of the robot is considered in

other design step (called in this paper as dynamic design) by including both the control and the inertial parameters.

The kinematic design is formulated as an optimization problem (static optimization problem) where the kinematic design variable vector $\mathbf{p}_{\mathbf{s}_{\mathbf{seqI}}} = [a_{s_1}, \ldots, a_{s_5}]^T \in R^5 \subset \mathbf{p}_{\mathbf{s}}$ (13), which involves the link lengths, is obtained by optimizing the *kinematic design objective* "the manipulability" (16) subject to the five-bar Grashof criteria (18) and the limits in the kinematic design variable vector (19). Once the kinematic design variable vector is obtained, other optimization problem must be solved in order to find the control design parameters $\mathbf{p}_{\mathbf{c}} \in R^6$ (14) and the shape of the links (which modify the inertial parameters) $\mathbf{p}_{\mathbf{sII}} = [b_{s_1} \ldots b_{s_4}, c_{s_1} \ldots c_{s_4}, d_{s_1} \ldots d_{s_4}, e_{s_1} \ldots e_{s_4}, f_{s_1} \ldots f_{s_4}, g_{s_1} \ldots g_{s_4}, h_{s_1} \ldots h_{s_4}, i_{s_1} \ldots i_{s_4}, j_{s_1} \ldots j_{s_4}, k_{s_1} \ldots k_{s_4}]^T \in R^{40} \subset \mathbf{p}_{\mathbf{s}}$ (13). Therefore, the second optimization problem in the sequential design approach is stated as to find the control and shape design parameter vector $\mathbf{p}_{\mathbf{sc}_{\mathbf{seq}}} = [\mathbf{p}_{\mathbf{c}}, \mathbf{p}_{\mathbf{sII}}]^T \in R^{46}$ that minimize the *dynamic design objective* "the tracking error" (15), subject to the differential equation modeling the closed loop robot dynamics (10) and the limits in both, the shape of the links (20) and the actuator torque (21). Hence, the dynamic characteristic of the parallel robot is included into the optimization problem, resulting in a dynamic optimization one. Optimal parameters with the sequential design approach are obtained by a similar hybrid gradient-differential evolution algorithm proposed in this paper.

The optimal values of the structure and control parameters in the synergetic design approach and the sequential one are shown in Table 3. The inertial parameters of the parallel robot links with the optimal design parameters given by the two approaches, are shown in Table 4. Rows marked with "***Syn***" are the optimal design parameters $\boldsymbol{p}_{12}^*$ (Table 3) or inertial parameters (Table 4) of the synergetic design approach. Rows marked with "*Seq*" are the optimal design parameters $\mathbf{p}_{\mathbf{sc}_{\mathbf{seq}}}{}^*$ (Table 3) or inertial parameters (Table 4) of the sequential design approach. Some

Table 3 Structural design parameters and gains of the PID controller obtained by synergetic and sequential design approach

i Design approach	1 **Syn**/*Seq*	2 **Syn**/*Seq*	3 **Syn**/*Seq*	4 **Syn**/*Seq*	5 **Syn**/*Seq*
$a_{s_i}^*$ (m)	0.1361/0.1361	0.1343/0.1361	0.3393/0.3414	0.3619/0.3569	0.3619/0.3569
$b_{s_i}^*$ (m)	0.0360/0.0409	0.0202/$b_{s_{2_{\min}}}$	$b_{s_{3_{\min}}}$	$b_{s_{4_{\min}}}$	
$c_{s_i}^*$ (m)	$c_{s_{1_{\min}}}$	$c_{s_{2_{\min}}}$	$c_{s_{3_{\min}}}$	$c_{s_{4_{\min}}}$	
$d_{s_i}^*$ (m)	$d_{s_{1_{\min}}}$	$d_{s_{2_{\min}}}$	$d_{s_{3_{\min}}}$	$d_{s_{4_{\min}}}$	
$e_{s_i}^*$ (m)	$e_{s_{1_{\min}}}$	$e_{s_{2_{\min}}}$	$e_{s_{3_{\min}}}$	$e_{s_{4_{\min}}}$	
$f_{s_i}^*, g_{s_i}^*, h_{s_i}^*, i_{s_1}^*, j_{s_i}^*, k_{s_i}^*$ (m)			These parameters are not considered because $f_{s_i} \to 0$ and $g_{s_i} \to 0$		
$k_{p_1}^*, k_{i_1}^*, k_{d_1}^*$		37.3806/33.5587	154.6058/127.2387	0.9067/0.8255	
$k_{p_2}^*, k_{i_2}^*, k_{d_2}^*$		19.1429/17.7888	146.4211/90.8849	0.4931/0.4522	

"***Syn***" synergetic design approach. "*Seq*" sequential design approach

Table 4 Inertial parameters of the parallel robot links with optimal structural design variables obtained by synergetic and sequential design approach

Inertial parameter Design approach	Link 1 $i = 1$ **Syn**/*Seq*	Link 2 $i = 2$ **Syn**/*Seq*	Link 3 $i = 3$ **Syn**/*Seq*	Link 4 $i = 4$ **Syn**/*Seq*
m_i^*(kg)	0.3609/0.3640	0.3493/0.3497	0.3149/0.3160	0.3971/0.3944
$l_{c_i}^*$(m)	0.0340/0.0334	0.0353/0.0360	0.1500/0.1510	0.1860/0.1835
γ_i^*(rad)	0	0	0	0
I_i^*(kg m^2 $\times$ 10^{-4})	10.91/11.07	10.18/10.44	54.56/55.33	87.50/84.81

"***Syn***" synergetic design approach. "*Seq*" sequential design approach

Table 5 Performance of the final design by using the synergetic and sequential design approach

Design approach	$\bar{J}_1 + \bar{J}_2$	$\bar{J}_3$	$\bar{J}$
Synergetic structure–control design	0.001160	−0.929283	−0.928123
Sequential structure–control design	0.001210	−0.929337	−0.928127

$\bar{J}_1 + \bar{J}_2$ tracking error index (15). $\bar{J}_3$ manipulability index (16). $\bar{J}$ weighted sum

design parameters in Table 3 tend to their lower limits both the synergetic and sequential design approach. It is observed that the design parameters $f_{s_i}^*$ and $g_{s_i}^*$ tend to zero. So, the design parameters $f_{s_i}^*$, $g_{s_i}^*$, $h_{s_i}^*$, $i_{s_1}^*$, $j_{s_i}^*$, $k_{s_i}^*$ for $i = 1$ to 4 do not improve the performance index because these design parameters can not modify the inertial parameters of the links (see Fig. 2). The design parameters could yield a mass center angle of the i-th link in the closed interval $[0, \pm\pi]$. Nevertheless as $f_{s_i}^*$ and $g_{s_i}^*$ tend to zero and $b_{s_i}^* << a_{s_i}^* + d_{s_i}^*$, the optimum mass center angles of all links are zero radians (see Table 4). As a conclusion, the performance index $\bar{J}$ is improved if the mass center angles of the links are equal to zero radians.

The performance of the final designs by using both the synergetic and sequential design approach is shown in Table 5. It is observed that the sequential design approach obtains a better manipulability index (see row $\bar{J}_3$) than the synergetic structure–control design, but the tracking error index (see row $\bar{J}_1 + \bar{J}_2$) is worst. This indicates a tradeoffs between the kinematic design and the dynamic one, i.e. improving the manipulability of the robot means the increase of the tracking error. Hence, the synergetic design approach finds the chosen balance between both performance indexes such that a superior design is obtained.

The Cartesian distance between the center of the desired trajectory and the generated trajectory with both approaches is shown in Fig. 7. This distance is called modulus of the trajectory. It is observed in thick line, the radius of the desired trajectory (desired circle); in continuous line, the radius of the generated trajectory with the robot designed with the synergetic approach and in dotted line, the radius of the generated trajectory with the robot designed with the sequential approach. The L1 norm of the modulus of the trajectory for the robot designed with the sequential

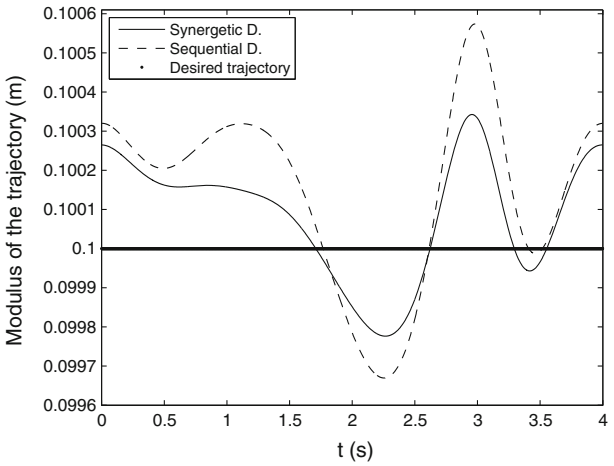

Fig. 7 The Cartesian distance between the center of the trajectory and the generated trajectory, called modulus of the trajectory

design approach is 0.1006 m and for the case of the robot designed with the synergetic design approach is 0.1003 m. These values are shown in Fig. 7 at $t = 3$ s. This indicates that the maximum deviations in the tracking error are 0.0006 and 0.0003 m, for both approaches, respectively. Then, the robot designed with the synergetic approach decrease the maximum deviation of the trajectory error around 50 %, when compared with the robot designed with the sequential design approach.

The improvement of the performance function (see row $\bar{J}$ in Table 5) of the synergetic structure–control design with respect to the sequential one is 0.0004 %. This improvement decreases the maximum deviation of the trajectory error at 50 % in the synergetic approach as it is commented above. Then the synergetic design approach will benefit the final product. Hence, the synergetic design framework must be considered in the development of products that require a superior performance.

5 Experimental results with the synergetically designed parallel robot

The real optimal structure manufactured with the parameters $\boldsymbol{p}_{12}^{*}$ of the synergetic structure–control design approach is shown in Fig. 8.

After the robot has been designed, its experimental performance should be evaluated. This Section presents the experimental results provided by the parallel robot designed with the proposed approach.

Regarding the implementation, it is well known that imperfections of actuators and sensors deteriorate the control performance. A critical imperfection is a nonlinearity which varies from component to component and increases with wear and tear. Actuators and sensors without or with less imperfections are costly to manufacture and their maintenance usually requires specialized personnel. So, in

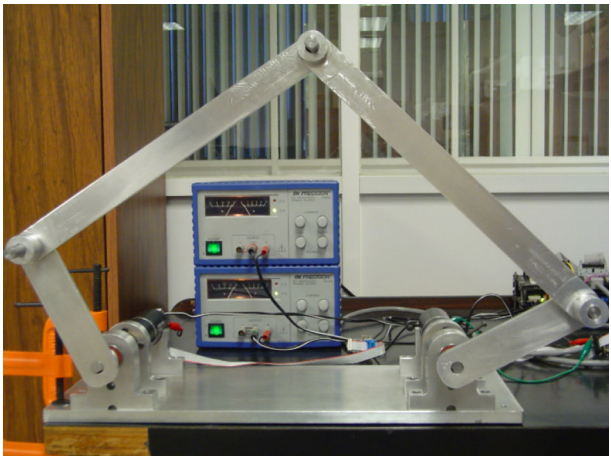

Fig. 8 Photo of the 5R 2DoF parallel robot with the optimum parameters p_{12}^{*} found by using the proposed synergetic structure–control design approach

order to eliminate or at least diminish these dynamics (nonlinearities), the inverse model (Tao and Kokotovic 1996) of these behaviors is used. In this paper the linear inverse model of the dead zone of the input torque is used. The schematic representation of the closed-loop system with the dead-zone and the inverse model for the dead-zone (D-Z) is shown in Fig. 9. The inverse model for the dead-zone characteristic is described in (28), where u_d is the input, u_{DZ} is the output, b_r is the right crossing, b_l is the left crossing and m_r, m_l determine the slopes. Via experiments, those parameters were selected as $b_r = b_l = 0.01$, $m_r = m_l = 0.4$.

$$u_{DZ}(t) = \begin{cases} \frac{u_d(t)+m_r b_r}{m_r} & \text{if} \quad u_d(t) > 0 \\ 0 & \text{if} \quad u_d(t) = 0 \\ \frac{u_d(t)+m_l b_l}{m_l} & \text{if} \quad u_d(t) < 0 \end{cases} \tag{28}$$

The system performance with the optimum structure–control design parameters p_{12}^{*} and with the inverse model of dead-zone (D-Z) is shown in Fig. 10 for simulation and experimental results. In Fig. 11 the Cartesian position errors for the

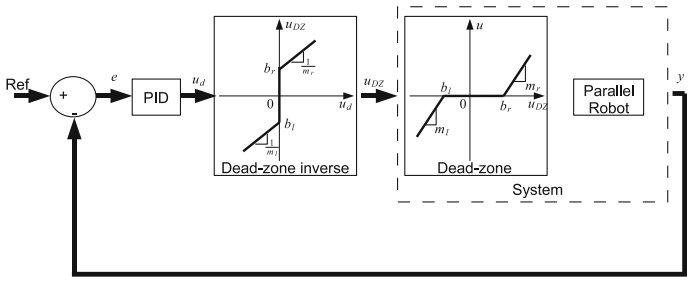

Fig. 9 Schematic representation of the closed-loop system with the nonlinearities

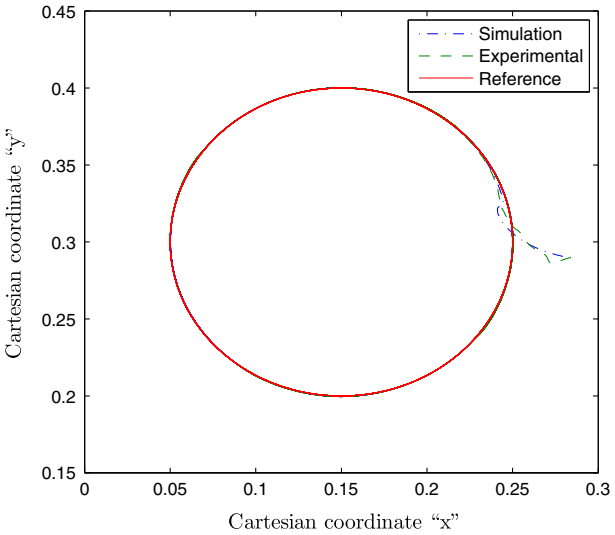

Fig. 10 Simulation and experimental results of the trajectory tracking of the 5R 2 DoF parallel robot

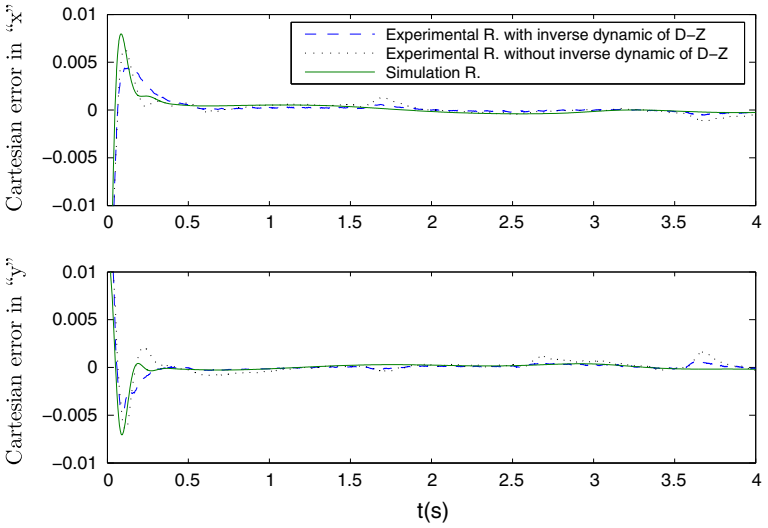

Fig. 11 Cartesian position error of the trajectory tracking

system with and without inverse model of D-Z are shown. It is observed that the Cartesian position errors are reduced when the inverse model of D-Z is included. After 0.5 s the maximum Cartesian position errors are around $\pm 0.5 \times 10^{-3}$ m for the simulation result and $\pm 0.6 \times 10^{-3}$, $\pm 1.3 \times 10^{-3}$ m for experimental results with and without inverse model of D-Z, respectively. Hence, the inverse model helps to

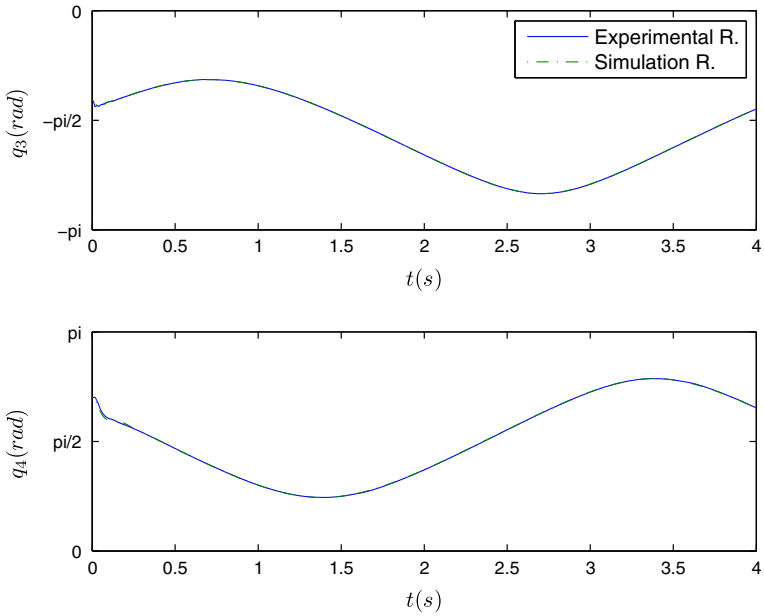

Fig. 12 Behavior of the actuated angles for the trajectory tracking of the 5R 2 DoF parallel robot via simulation and experimental results

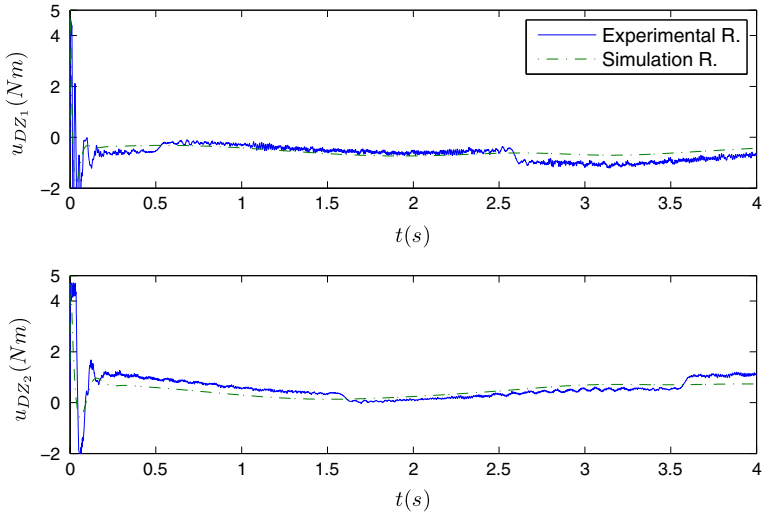

Fig. 13 Control signal behavior for the trajectory tracking

improve the position errors and to have experimental results as close as simulation results.

On the other hand, there are not singular configurations (the Jacobian determinant is not zero) in the trajectory tracking since the unactuated angles are in the following

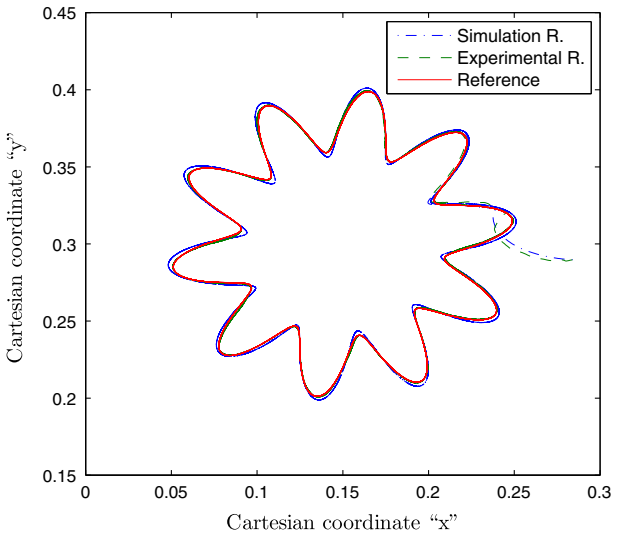

Fig. 14 Simulation and experimental results of a different trajectory tracking of the 5R 2 DoF parallel robot

interval: $-2.62 \leq q_3 \leq -0.98$ and $0.76 \leq q_4 \leq 2.47$ rad as it is observed in Fig. 14. So, the synergetic design objectives which involve the position errors and the manipulability measure are satisfied. In addition, in Fig. 13, both control signals hit their limits (5 Nm) as it is established in the optimization problem. Hence, all constraints are fulfilled.

The design parameters found with the proposed synergetic structure–control design approach do not only behave well with a circle as the desired trajectory. The design parameters could also work well with other trajectories as it is observed in Fig. 14. Nevertheless, the obtained parameters will only be optimum with the circle and hence there exist a small error in the tracking of the new proposed trajectory. That means, that if it is desired that the robot end effector follows other trajectories, the found design parameters p_{12}^* will not be optimum. So, it is necessary to include the new trajectory into the optimization problem in order to obtain the optimum design parameters for this new trajectories.

6 Conclusions

In this paper, a synergetic structure–control design of the planar 5R 2DoF parallel robot and its control system is formulated as a dynamic optimization problem, where the design space is highly dimensional. The proposed design methodology has two main advantages: (1) it is stated as a *nonlinear dynamic optimization problem* where both the *kinematic behavior* and the *dynamic behavior* are considered in a single design stage. (2) Dimensional synthesis, dynamic parameter design and control design are simultaneously considered by taking into account the

synergetic compromises among different performance functions. A superior design is done by considering this integration.

On the other hand, the main drawback of the proposed NLDOP is the presence of singularity regions. In such configurations, nonlinear programming techniques can not be used to solve it. In addition, this NLDOP presents a high sensitivity to initial conditions when it is solved by NLPTs. Therefore, the main advantages of using the proposed hybrid gradient-evolutionary optimization can be stated as follows: (1) it provides a better performance of the robot. (2) It is very useful in nonlinear dynamic optimization problems where singularity regions appear in the search space as well as several local solutions. (3) The convergence time is reduced.

After using the hybrid gradient-evolutionary optimization algorithm, the optimal design parameters are found. These parameters optimize the performance function and satisfy all constraints in simulation and experimental results. Considering the link shape as the structural design parameters, the mass center angles γ_i of the links can be modified without considering an added counter-weight system.

Acknowledgments The first author is grateful to Prof. Carlos A. Coello for useful discussions and courses taught. The authors acknowledges support from the National Council for Science and Technology of Mexico (CONACyT) through Grants 182298 and 084060 and the support from "Secretaría de Investigación y Posgrado" (SIP) of the Instituto Politécnico Nacional (IPN) through Grant No. 20131053.

References

Alvarez-Gallegos J, Cruz-Villar CA, Portilla-Flores E (2005) Evolutionary dynamic optimization of a continuously variable transmission for mechanical efficiency maximization, vol 3789. Lecture Notes in Artificial Intelligence Springer, Berlin, pp 1093–1102

Asada H (1983) A geometrical representation of manipulator dynamics and its application to arm design. ASME J Dyn Syst Meas Control 105:131–135

Arnold DV, Salomon R (2007) Evolutionary gradient search revisited. IEEE Trans Evol Comput 11:480–495

Betts JT (2010) Practical methods for optimal control and estimation using nonlinear programming, 2nd edn. SIAM, Philadelphia

Biegler LT (2010) Nonlinear programming: concepts, algorithms, and applications to chemical processes. SIAM, Philadelphia

Bryson-Jr AE (1999) Dynamic optimization. Addison-Wesley, Menlo Park

Cruz-Villar CA, Alvarez-Gallegos J, Villarreal-Cervantes MG (2009) Concurrent redesign of an underactuated robot manipulator. Mechatronics 19:178–183

Deb K, Kain S (2003) Multi-speed gearbox design using multi-objective evolutionary algorithms. ASME J Mech Des 125:609–619

Deb K, Pratap A, Agarwal S, Meyarivan T (2002) A fast and elitist multiobjective genetic algorithm NSGAII. IEEE Trans Evol Comput 6(2):182–197

Fu K, Mills JK (2005) A convex approach solving simultaneous mechanical structure and control system design problems with multiple closed-loop performance specifications. ASME J Dyn Syst Meas Control 127:57–68

Ghorbel FH, Chételat O, Gunawardana R, Longchamp R (2000) Modeling and set point control of closed-chain mechanisms: theory and experiment. IEEE Trans Control Syst Technol 8:801–815

Goldberg DE (1989) Genetic algorithms in search, optimization and machine learning. Addison-Wesley, Mass

Hibbeler RC (2003) Engineering mechanics : statics & dynamics. Prentice-Hall, New Jersey

Liu XJ, Wang J, Pritschow G (2006) Kinematics, singularity and workspace of planar 5r symmetrical parallel mechanisms. Mechanism and machine theory 41:145–169

Ma O, Angeles J (1990) The concept of dynamic isotropy and its applications to inverse kinematics and trajectory planning. In: Proceedings of the IEEE international conference on robotics and automation, pp 481–486

Merlet JP (2001) Parallel robots. Kluwer Academic Publishers, Dordrecht

Messac A, Turner J, Soosaar K (1985) An integrated control and minimum mass structural optimization algorithm for large space structures. In: JPL proceedings of the workshop on identification and control of flexible space structures, vol 2, pp 231–266

Mezura E, Coello-Coello CA (2005) A simple multimembered evolution strategy to solve constrained optimization problems. IEEE Trans Evol Comput 9:1–17

Pil AC, Asada H (1996) Integrated structure/control design of mechatronics systems using a recursive experimental optimization method. IEEE/ASME Trans Mechatron 1:191–203

Price KV, Storn RM, Lampinen JA (2005) Differential evolution: a practical approach to global optimization. Springer, New York

Ravichandran T, Wang D, Heppler G (2006) Simultaneous plant-controller design optimization of a two-link planar manipulator. Mechatronics 16:233–242

Salomon R (1998) Evolutionary algorithms and gradient search: similarities and differences. IEEE Trans Evol Comput 2:45–55

Tao G, Kokotovic PV (1996) Adaptive control of systems with actuator and sensor nonlinearities. Wiley, New York

Ting KL (1986) Five-bar grashof criteria. J Mech Transm Autom Des 108:533–537

Tsai LW (1999) Robot analysis: the mechanics of serial and parallel manipulators. Wiley, New York

Tu Q, Rastegar J (1993) Determination of allowable manipulator link shapes; and task, installation, and obstacle spaces using the monte carlo method. J Mech Des 115:457–461

Villarreal-Cervantes MG, Cruz-Villar CA, Alvarez-Gallegos J, Portilla-Flores EA (2010) Differential evolution techniques for the structure–control design of a five-bar parallel robot. Eng Optim 42:535–565

Yan HS, Yan GJ (2009) Integrated control and mechanism design for the variable input-speed servo four-bar linkages. Mechatronics 19:274–285

Yoshikawa T (1985) Dynamic manipulability of robot manipulators. J Robot Syst 2:113–124

Zhang WJ, Li Q, Guo LS (1999) Integrated design of mechanical structure and control algorithm for a programmable four-bar linkage. IEEE/ASME Trans Mechatron 4:354–362