Parametric Reconfiguration Improvement in Non-Iterative Concurrent Mechatronic Design Using an Evolutionary-Based Approach

Edgar Alfredo Portilla-Flores[a,*], Efren Mezura-Montes[b], Jaime Alvarez-Gallegos[c], Carlos Artemio Coello-Coello[d], Carlos Alberto Cruz-Villar[c], Miguel Gabriel Villarreal-Cervantes[e]

[a]*Mechatronic Section, Postgraduate Department, CIDETEC-IPN, Av. Juan de Dios Bátiz s/n Esq. Miguel Othon de Mendizabal, Unidad Profesional Adolfo López Mateos, C.P. 07700, México D.F., México*
[b]*Laboratorio Nacional de Informática Avanzada (LANIA A.C.), Rébsamen 80, Centro C.P. 91000, Xalapa, Veracruz, México*
[c]*Mechatronic Section, Electrical Engineering Department, Cinvestav-IPN, Av. Instituto Politécnico Nacional 2508, Col. San Pedro Zacatenco, C.P. 07360, México D.F., México*
[d]*Computer Science Department (Evolutionary Computation Group), Cinvestav-IPN, Av.Instituto Politécnico Nacional 2508, Col. San Pedro Zacatenco, C.P. 07360, México D.F., México*
[e]*Advanced Technology Department, UPIITA-IPN, Av. Instituto Politénico Nacional 2508, Col. San Pedro Zacatenco, C.P. 07340, México D.F., México*

Abstract

Parametric reconfiguration plays a key role in non-iterative concurrent design of mechatronic systems. This is because it allows the designer to select, among different competitive solutions, the most suitable without sacrificing sub-optimal characteristics. This paper presents a method based on an evolutionary algorithm to improve the parametric reconfiguration feature in the optimal design of a continuously variable transmission and a five-bar parallel robot. The approach considers a solution-diversity mechanism coupled with a memory of those sub-optimal solutions found during the process. Further-

*Corresponding author. Tel.:+52 5557296000.
Email address: `aportilla@ipn.mx` (E. A. Portilla-Flores), `emezura@lania.mx` (E. Mezura-Montes), `jalvarez@cinvestav.mx` (J. Alvarez-Gallegos), `ccoello@cs.cinvestav.mx` (C. A. Coello-Coello), `cacruz@cinvestav.mx` (C. A. Cruz-Villar), `gvillarr@cinvestav.mx` (M. G. Villarreal-Cervantes) (Miguel Gabriel Villarreal-Cervantes)

more, a constraint-handling mechanism is added to bias the search to the feasible region of the search space. Differential Evolution is utilized as the search algorithm. The results obtained in a set of five experiments performed per each mechatronic system show the effectiveness of the proposed approach.

Keywords: Concurrent Design, Mechatronic Systems, Evolutionary Algorithms, Multi-objective Optimization

1. Introduction

Nowadays, machines have changed from purely mechanical systems to complex mechatronic systems which integrate mechanical and electrical components, electronic devices, control systems and software tools. This is the reason why it is necessary to use new design methodologies that consider integral aspects of a system as a whole.

In order to fulfill these requirements, a multidisciplinary analysis approach must be used in the design process of mechatronic systems. This approach deals with the mechanical behavior and the dynamic performance of the system. In this way, a concurrent design concept must be used in order to jointly consider the mechanical and control performances as well. Therefore, a real challenge in the design of mechatronic systems is to find a set of parameters which achieves the best performance, or at least improves it. These parameters must provide the mechanical design and the control of the whole system.

In the concurrent design framework, several approaches have been proposed (Pil and Asada, 1996),(Zhang et al., 1999),(Li et al., 2001). However, these concurrent approaches are based on an iterative process, i.e., the parameters obtained to build the mechanical structure are obtained in a first step and the parameters of the controller are found in a second step. If the resulting control structure is very difficult to implement, then the first step must be repeated all over again.

On the other hand, an alternative approach to formulate the mechatronic system design problem is to consider it as a dynamic optimization problem (Alvarez-Gallegos et al., 2005a,b). In order to do this, the parametric optimal design of the mechatronic system needs to be stated as a constrained multi-objective dynamic optimization problem (MDOP). In this approach, both the kinematic and the dynamic models of the mechanical structure and the dynamic model of the controller are considered at the same time, together with system performance criteria. This approach aims to obtain the set of optimal mechanical and controller parameters in a single step.

Based on the fact that the design problem can be modeled as a MDOP, the final solution will consist of a set of trade-off solutions. Therefore, depending of the degree of conflict between the kinematic and the dynamic models, this final set may be biased to some type of solutions from where the structure and control parameters could complicate the parametric reconfiguration of the system. In this work, parametric reconfiguration is defined as the diversity of sub-optimal solutions in the objective space i.e. in the values of the two objectives to be optimized. This diversity is reflected in different sub-optimal designs from where the designer may choose one according to suitable objective values. An adequate parametric reconfiguration allows the modification of the mechanical structure (e.g., the mass, length, mass center length, etc.) and the parameters of the control system (e.g., the gains of the controller) without losing sub-optimal conditions of both criteria. In this way, a wide set of final solutions is highly desirable.

A MDOP can be solved by its conversion into a Nonlinear Dynamic Optimization Problem (NLDOP). There are several mathematical programming methods to solve multi-objective optimization problems (Miettinen, 1999). However, for complex real-world problems such as those from the noniterative approach used in this work, they present different shortcomings e.g.,(1) these methods have the possibility of getting trapped at local minimum in the neighborhood of the starting search point, upon the degree of non-linearity and initial conditions, (2) they require a transformation of the original problem (Alvarez-Gallegos et al., 2005a), (3) they are sensitive to the initial conditions and they involve the computation of the gradient and the Hessian of the objective function and constraints, which implies that continuity of the second order must be ensured (Portilla-Flores et al., 2007) and, (4) they usually provide a single solution per run. In order to avoid this problem, various points are needed to initialize the solution search; nevertheless, a considerable sensitivity to its initial search point is observed on the algorithm's convergence (Cruz-Villar et al., 2009). These disadvantages indicate that a mathematical programming method is not convenient to promote the parametric reconfiguration of the system.

In the recent past, non-traditional optimization techniques based on stochastic methods such as evolutionary algorithm (EAs), genetic algorithms (GAs) or particle swarm optimization (PSO) have been developed to overcome these drawbacks. Among the advantages of these approaches are: (1) these are population-based methods, therefore they are not sensitive to their starting point, and a global minimum solution can be reached (although not for every problem), (2) they do not require additional information in order to start the search, i.e. gradients, Hessian matrices, initial search points, etc.; (3) with these methods, complex problems can be solved, meaning that the optimization problem can include discontinuous physical models; i.e. they do not require the objective functions and constraints to be continuous and/or differentiable (4) finally, these methods are independent of the problem characteristics; that is, these methods can be used and/or adapted to a large set of problems, because they do not require special mathematical formulation (problem transformation) in order to obtain a set of solutions.

Therefore, the use of heuristic-based approaches working with a set of solutions such as Evolutionary Algorithms (EAs) has become very useful (Eiben and Smith, 2003). In (Mezura-Montes et al., 2008b), it was found that the performance of an Evolutionary Algorithm was clearly superior with respect to that provided by a Mathematical Programming approach in non-iterative concurrent design for a pinion-rack CVT mechanism. However, it was observed that getting the proper distribution of trade-off solutions is a difficult task in this type of designs.

These disadvantages lead, as mentioned before, to a deficient reconfigurability property of the mechatronic system design. This is the motivation of the current research, which aims to provide a competitive albeit simple method to generate feasible sub-optimal designs by promoting diversity in the final set of solutions obtained. The goal is to provide the designer with a wider set of sub-optimal solutions which facilitate the reconfiguration of the whole mechatronic system. The mechanism must have a low computational cost because the evaluation of a single solution of this type of systems requires a significant processing time.

The rest of the paper is organized as follows: Section 2 contains the formal definition of the non-iterative concurrent design. After that, Section 3 summarizes different approaches reported in the specialized literature on concurrent design. Our proposed evolutionary-based approach to improve the reconfigurability feature in non-iterative concurrent design is detailed in Section 4. Section 5 presents the two mechatronic system design problems to be solved, while the experimental design and the results obtained in each problem, besides their corresponding discussions, are shown in Section 6. The paper finishes with some conclusions and possible paths for future work in Section 7.

2. Statement of the Problem

As it has been previously mentioned, the mechatronic design problem should be established in a non-iterative concurrent way. Therefore, the system to be designed can be mathematically expressed as a MDOP as follows:

$$\min \Phi(x,p,t) = [\Phi_1, \Phi_2, ..., \Phi_n]^T \tag{1}$$

$$\Phi_i = \int_{t_0}^{t_f} L_i(x,p,t)dt \qquad i = 1,2,...,n$$

under p and subject to:

$$\dot{x} = f(x,p,t) \tag{2}$$

$$g(x,p,t) \leq 0 \tag{3}$$

$$h(x,p,t) = 0 \tag{4}$$

$$x(0) = x_0$$

In the problem stated by equations (1) to (4): p is a vector of the design variables which belong to the mechanical and control structure, x is the vector of the state variables and t is the time variable. Specifically, given a set of initial values x_0 for the state variables, called the initial conditions, the dynamic model defined by $f(\cdot)$ must be solved in order to obtain the state vector x at time t. This dynamic model is represented mathematically by a set of nonlinear differential equations. Additionally, some performance criteria $\Phi_i(\cdot)$ must be selected for the mechatronic system. On the other hand, practical engineering problems are ever constrained by a set of conditions which belong to mechanical and control conditions. As a result, constraints $g(\cdot)$ and $h(\cdot)$ generally are nonlinear real-valued functions of the vector that contains the design variables p, the vector of the state x and the time variable t. Therefore, the parameter vector p, which is a solution of the previous problem, will be an optimal set of structure and controller parameters which minimizes the performance criteria selected for the mechatronic system (1) and it will be subject to the constraints imposed by the dynamic model of the system (2) and the design. If criteria are not in conflict among them and all of them are equally important, then only one possible solution can be found. However, if there are indeed conflicts among the performance criteria, a set of trade-off solutions will be possible and desirable. The full set of trade-off solutions may be very large and even impossible to find. Nevertheless, obtaining a good set of well-distributed trade-off solutions is highly desirable, so that an engineer can consider a wider range of options before reaching a decision. This is precisely one of the main aims of this paper, in which a simple approach is introduced in order to improve the distribution of trade-off solutions obtained by a multi-objective evolutionary algorithm.

3. Previous Related Work

Usually, the design of mechanical elements involves kinematic and static behaviors while the design of the control system uses mainly the dynamic behavior. Therefore, from a dynamic point of view, this approach cannot produce an optimal system behavior (Norton, 1996; Van Brussel et al., 2001). Several works on mechatronic systems design propose a concurrent design methodology which simultaneously considers the mechanical and control performances. In (Pil and Asada, 1996), a concurrent design concept was proposed using an integrated structure/control design method, based on an iterative algorithm for robotic system development. In this design method, the mechanical structure was modified iteratively and the control parameters were adjusted according to the mechanical structure update. Another concurrent design methodology was proposed in (Zhang et al., 1999) where the main objective was to improve the motion tracking performance for an existing four-bar closed loop linkage. An appropriate mechanical design produces a simple dynamic model. With the simplified dynamic model, a simple controller design was obtained. In (Li et al., 2001) a concurrent method for mechatronic systems design was proposed. A general model was required to mathematically describe the mechatronic system. The design method allowed to obtain a simple mechanical structure and its dynamic model. The dynamic model favored an easier controller design which improved the dynamic performance. The above methods proposed a concurrent design concept based on an iterative process.

On the other hand, some works have presented non-iterative concurrent design approaches. In (Ravichandran et al., 2006), 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 was shown. In that work, a simultaneous plant-controller design optimization problem and the description of solution techniques based on an evolutionary algorithm for solving the optimization problem were considered. In (Portilla-Flores et al., 2007), a concurrent design methodology to formulate the mechatronic design problem was proposed. The methodology states the mechatronic design problem as a dynamic optimization problem. A concurrent design of a pinion-rack Continuously Variable Transmission was carried out by using both mathematical programming and evolutionary algorithms.

From this literature review it is clear that:

- Most of the existing work is still focused on iterative concurrent design.
- From the still scarce set of non-iterative concurrent design approaches, none of them has promoted the parametric reconfiguration, which is very important within this methodology.

4. Our Proposed Approach

In this section, the proposed approach to promote a better reconfigurability property in the obtained solutions of a non-iterative concurrent design method is explained. The search algorithm utilized was Differential Evolution and was adapted to solve constrained multi-objective problems. This adaptation and the reconfigurability promotion mechanism are presented below.

4.1. Evolutionary Algorithm

Based on the fact that: (1) the computational cost of the type of designs solved by the non-iterative concurrent design is usually high, (2) an easy-to-implement method is highly desirable and (3) a competitive performance for Differential Evolution (DE) (Price et al., 2005) had been observed in previous works (Alvarez-Gallegos et al., 2005b), this approach was chosen as our search engine.

DE is a simple, but powerful direct-search algorithm which was designed to solve global numerical optimization problems. DE does not require the objective function of the problem and/or the constraints to be linear, differentiable or continuous i.e. it works as a black-box without requiring specific features of the problem being solved. DE simulates natural evolution combined with a mechanism to generate multiple search directions based on the distribution of solutions, called vectors (a design solution in this paper), in the current population of size NP. Each vector $\vec{s}_{i,g}, i = 1, \ldots, NP$ containing n decision variables, in the current population at generation g,

$\vec{s}_{i,g} = [s_{1,i,g}, s_{2,i,g}, \ldots, s_{n,i,g}]^T$, called at the moment of reproduction as the target vector, is able to generate one offspring, called trial vector $\vec{v}_{i,g}$. This trial vector is generated as follows: First of all, a search direction is defined by calculating a difference vector by subtracting two randomly chosen vectors (r_1 and r_2) from the current population $\vec{s}_{r_1,g}$ and $\vec{s}_{r_2,g}$. This difference vector is also scaled by using a user-defined parameter called *scale factor* $F \geq 0$ (Price et al., 2005). This scaled difference vector is then added to a third vector (r_0) $\vec{s}_{r_0,g}$, called base vector. As a result, a new vector is obtained, known as the mutant vector. After that, this mutant vector is recombined, based on a user-defined parameter, called crossover probability $0 \leq CR \leq 1$, with the target vector (also called parent vector) by using discrete recombination, usually uniform (i.e., binomial crossover), to generate a trial (child) vector. The CR value determines how similar the trial vector will be with respect to the mutant vector.

Despite the fact that different DE variants have been proposed (Price et al., 2005), DE/rand/1/bin, remains as the most popular in the specialized literature, and it is the utilized version in this work. The first term in the variant's name means Differential Evolution, the second term indicates how the base vector is chosen (at random in this case), the number in the third term indicates how many vector differences (i.e., vector pairs) will contribute in the differential mutation (one pair in this case). Finally, the fourth term shows the type of crossover utilized (binomial, in this case). The detailed pseudocode of DE/rand/1/bin to solve unconstrained single-objective optimization problems is presented in Figure 1 and a graphical example is explained in Figure 2.

Based on the type of problems solved by the methodology explained in Section 2, DE was adapted to deal with two objective functions and several constraints. Therefore, the selection criterion was modified and an external archive was utilized to store the optimal solutions found during the search.

There are different approaches based on DE to solve multi-objective optimization problems (Mezura-Montes et al., 2008a) and some of them have been applied to mechatronic design problems (Alvarez-Gallegos et al., 2005b; Saravanan and Ramabalan, 2008). However, they usually solve unconstrained multi-objective problems or they use additional mechanisms which modify the simplicity of DE by considerably increasing its processing time. In contrast, here, we propose a DE-based approach that retains the simplicity of the original algorithm and which adds computationally inexpensive mechanisms to solve constrained multi-objective optimization problems. The modifica-

```
Begin
    g=0
    Create a random initial population $\vec{s}_{i,g}$ $\forall i$, $i = 1, \ldots, NP$
    Evaluate $\Phi_1(\vec{s}_{i,g})$ $\forall i$, $i = 1, \ldots, NP$
    For g=1 to MAX_GEN Do
        For i=1 to NP Do
            Select randomly $r_0 \neq r_1 \neq r_2 \neq i$
            $j_{rand}$ = randint$[1, n]$
            For j=1 to $n$ Do
                If (rand$_j[0, 1] < CR$ or $j = j_{rand}$) Then
                    $v_{j,i,g+1} = s_{j,r_0,g} + F(s_{j,r_1,g} - s_{j,r_2,g})$
                Else
                    $v_{j,i,g+1} = s_{j,i,g}$
                End If
            End For
            If ($\Phi_1(\vec{v}_{i,g+1}) \leq \Phi_1(\vec{s}_{i,g})$) Then
                $\vec{s}_{i,g+1} = \vec{v}_{i,g+1}$
            Else
                $\vec{s}_{i,g+1} = \vec{s}_{i,g}$
            End If
        End For
        $g = g + 1$
    End For
End
```

Figure 1: “DE/rand/1/bin” pseudocode for unconstrained single-objective numerical optimization. rand$_j[0, 1]$ is a function that returns a real number between 0 and 1. randint[min,max] is a function that returns an integer number between min and max. NP, MAX_GEN, CR and F are user-defined parameters. n is the number of decision variables of the problem. The set of NP solutions to the non-iterative concurrent mechatronic design is represented by $\vec{s}_{i,g}$, $\forall i$, $i = 1, \ldots, NP$

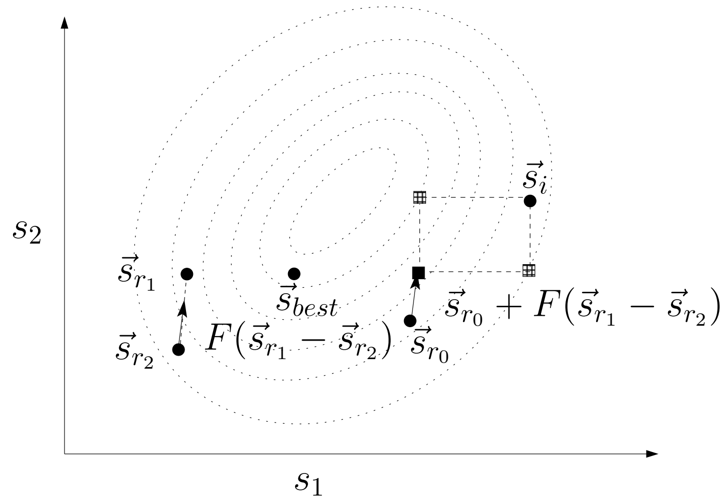

Figure 2: DE/rand/1/bin graphical example. $\vec{s}_i$ is the target vector, $\vec{s}_{r_0}$ is the base vector chosen at random, $\vec{s}_{r_1}$ and $\vec{s}_{r_2}$ (chosen at random as well) generate the difference vector to define a search direction. The black square represents one of the possible locations of the trial vector generated after performing recombination. The other two squares represent the two other possible locations for the trial vector after recombination.

tions introduced are the following:

Instead of using one objective function value as the only criterion to select the fitter solution between the target and trial vectors, *Pareto Dominance* was utilized as a criterion to select between them. The aim is to keep the non-dominated solutions from the current population, because they represent a better trade-off among the objectives (Coello Coello et al., 2007). A vector of objectives $\Phi = [\Phi_1, \ldots, \Phi_k]$ is said to Pareto dominate $\Phi' = [\Phi'_1, \ldots, \Phi'_k]$ (denoted by $\Phi \preceq \Phi'$) if and only if Φ is partially less than Φ', i.e. $\forall i \in \{1, \ldots, k\}$, $\Phi_i \leq \Phi'_i \wedge \exists i \in \{1, \ldots, k\} : \Phi_i < \Phi'_i$. The set of all the Pareto non-dominated solutions (i.e., those that are not dominated by any other solution) is called the *Pareto optimal set*. The objective function values corresponding to the solutions contained in the Pareto optimal set constitute the so-called *Pareto front* of the problem. In our case, the mechatronic design problems have two objectives, one related to the mechanical design and another related to the controller of the mechanism. Therefore, $k = 2$ in this paper.

The following expression formally defines the set of solutions to be obtained as a result in a multi-objective problem:

If we denote the feasible region of the search space as $\mathcal{F}$, the multi-objective evolutionary algorithm will look for the Pareto optimal set ($\mathcal{P}^*$)

defined as:

$$\mathcal{P}^* := \{\vec{v} \in \mathcal{F} \mid \neg\exists\ \vec{s} \in \mathcal{F}\ \ \Phi(\vec{s}) \preceq \Phi(\vec{v})\}. \tag{5}$$

As in real-world problems $\mathcal{P}^*$ is unknown, a sub-optimal Pareto set including sub-optimal trade-off solutions for the non-iterative concurrent design is the solution sought.

Since DE was conceived as an unconstrained optimization technique, a constraint-handling technique had to be added to the proposed approach. A review of the specialized literature showed that the technique proposed by Deb (Deb, 2000) has provided very competitive results when combined with DE (Kukkonen and Lampinen, 2006; Zielinski and Laur, 2008; Mezura-Montes et al., 2006; Huang et al., 2006). Furthermore, these comparison criteria do not add extra parameters to be fine-tuned by the user, as traditional penalty functions. The comparison criteria from (Deb, 2000) has to be modified, in order to incorporate Pareto Dominance. The new comparison criteria are the following (Oyama et al., 2007):

- Between 2 feasible design solutions, the one which dominates the other is preferred.
- If one design solution is feasible and the other one is infeasible, the feasible design solution is preferred.
- If both design solutions are infeasible, the one with the lowest sum of constraint violation is preferred.

As a result, the selection criterion detailed in rows $16-20$ in Figure 1 is replaced with the criteria shown in Figure 3.

```
If (Φ(v_{i,g+1}) is better thanΦ(s_{i,g})  based on the three selection criteria) Then
    s_{i,g+1} = v_{i,g+1}
Else
    s_{i,g+1} = s_{i,g}
End If
```

Figure 3: Modified selection mechanism added to the DE algorithm. This process replaces rows $16-20$ in Figure 1.

Because of the importance of adding elitism to a multi-objective evolutionary algorithm (Coello Coello et al., 2007) and more specifically, to DE

(Mezura-Montes et al., 2008a), the proposed approach adopts an external archive which stores the set of non-dominated vectors found during the optimization process. This archive is updated at each generation in such a way that all non-dominated solutions from the population will be included in the archive. After that, a non-dominance checking is performed with respect to all the solutions (the newcomers and also the solutions in the archive). The solutions that are non-dominated with respect to every other solution will remain in the archive. When the search ends, the set of non-dominated solutions in the archive will be reported as the final set of solutions obtained by the approach.

4.2. Reconfigurability promotion by a crowding mechanism

Based on the need to improve the reconfigurability in the solutions obtained in the optimization process, a more diverse set of solutions is required in the Pareto optimal set. Different techniques have been proposed to promote and preserve diversity in the objective space of a multi-objective problem solved by an evolutionary algorithm, such as niches (Fonseca and Fleming, 1996), ϵ-dominance (Laumanns et al., 2002), crowding distance (Deb et al., 2002), among others.

From these options, the crowding distance was chosen based on the fact that it does not add extra parameters which require specific knowledge of the problem and that require to be fine-tuned (e.g., niching requires a niche radius and ϵ-dominance requires an ϵ value, both of which are problem dependent). Furthermore, its performance has been found to be highly competitive when used within multi-objective evolutionary algorithms (Deb et al., 2002).

The crowding distance operates on the space defined by the vectors of functions, usually named function space (2-objective space in this paper), and it estimates the perimeter of the cuboid for each solution, which is formed by using the nearest neighbors as the vertices. See Figure 5.

As it can be noted, the crowding distance gives an idea of how crowded are the closest neighbors of a given vector of functions in the objective function space. Therefore, a higher value is preferred, and it can be utilized as a criterion to select those solutions which objective function values are more different from the rest of them, i.e., the parametric reconfiguration among non-dominated solutions can be improved.

The implementation of the crowding distance is as follows (Deb et al., 2002): The set of non-dominated solutions, which are stored in the external archive in our case, is sorted with respect to each objective function value, i.e., k sorted lists will be obtained ($k = 2$ in our case). For each sorted list, the non-dominated solutions located at the beginning and at the end of the list are assigned an ∞ crowding distance value, i.e., they are good candidate solutions because they can extend the length of the Pareto front and the parametric reconfiguration might be improved. After that, for the remaining solutions, the normalized difference between the two adjacent values is calculated. The details are presented in Figure 4.

Begin
    For ($j = 1$ to *Sol*) **Do** // *Sol* is the size of the non-dominated set
        Crw_Dist(j) = 0
    End For
    For ($i = 1$ to k) **Do** // k=2 objective functions in our case
        Sort in descent order the non-dominated set
        with respect to objective function Φ_i
        Crw_Dist(1) = ∞
        Crw_Dist(*Sol*) = ∞
        For ($j = 2$ to *Sol*-1) **Do**
            Crw_Dist(j)=Crw_Dist(j)+$\frac{|\Phi_i^{j-1} - \Phi_i^{j+1}|}{\Phi_i^{max} - \Phi_i^{min}}$
        End For
    End For
End

Figure 4: Calculation of the crowding distance Crw_Dist(j) for solution j in the set of non-dominated solutions. *Sol* is the number of non-dominated solutions in the external archive, k is the number of objective functions of the problem, Φ_i^{min} and Φ_i^{max} are the minimum and maximum values, respectively, for objective function Φ_i.

The crowding distance value added to the DE/rand/1/bin algorithm considers the following:

- The external archive must contain at least three non-dominated solutions. This is the minimum number of solutions required in the reproduction step within DE.

- A parameter called *Normal Selection* (NS) is introduced in order to indicate the percentage of generations where the selection of vectors (design solutions) to apply the DE operators is made in a traditional way for the three randomly chosen vectors from the current population to generate one trial vector (as explained in Section 4.1 and detailed in

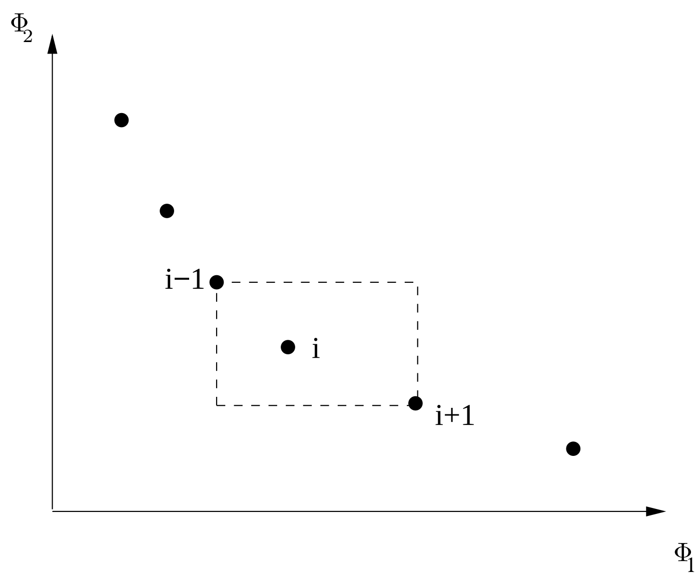

Figure 5: The crowding distance value in an two-objective example. Vectors with a larger value of this value are preferred.

> Figure 1). The remaining $(1-NS)$ percentage of generations, selections are made only from the external archive i.e., the set of non-dominated solutions, using the crowding distance as a criterion. The aim is to promote diversity among the solutions obtained, such that a better defined Pareto front can be obtained.

This selection mechanism based on crowding distance works as follows: Instead of selecting vectors r_0, r_1 and r_2 from the current population (row 7 in Figure 1), they are selected from the external archive based on the crowding distance value of each vector within it.

This type of selection will be activated according to the NS parameter value $(0 \leq NS \leq 1)$. That is, while the generation number g is smaller than the product of the normal selection value NS and the maximum number of generations MAX_GEN, the three vectors are selected from the current DE population at random (row 7 in Figure 1), i.e., the search focuses on finding feasible nondominated solutions (promising designs). Otherwise, the individuals are selected from the external archive by using the crowding distance value as a criterion, i.e., the search, based on the nondominated solutions previously found, looks to improve the shape of the Pareto front (the parametric reconfiguration is improved). Larger values of the crowding distance are preferred.

This combination of selection mechanisms within DE, and controlled by the NS parameter, promotes two behaviors:

- The first part of the search aims to explore the search space by allowing randomly chosen vectors to generate a diverse set of search directions in order to find non-dominated vectors to fill in the external archive.
- The second part of the process keeps DE from focusing in those previously found regions of the Pareto front by choosing, based on the crowding distance factor, those non-dominated vectors in scarcely explored regions of the Pareto front.

The mechanism which substitutes that in row 7 from Figure 1 in order to promote reconfigurability in optimal designs is shown in Figure 6.

```
If (g ≤ (NS × MAX_GEN)) Then
    Select r0, r1 and r2 randomly
Else
    If (Sol < 3) Then
        Select r0, r1 and r2 randomly
    Else
        Select r0, r1 and r2 from the external archive
        based on crowding distance factor
    End If
End If
```

Figure 6: Reconfigurability promotion mechanism added to the DE/rand/1/bin algorithm. g is the current generation number, NS is the normal selection parameter value, MAX_GEN is the maximum number of generations to be performed by the algorithm and Sol is the number of vectors in the external archive. r_0 is the base vector and r_1 and r_2 are used to calculate the difference vector. This mechanism replaces row 7 in Figure 1
.

5. Mechatronic Design Problems

Two real-world mechatronic design problems with different features were used to test the proposed approach explained in Section 4. A detailed description of each one is presented next.

5.1. Continuously Variable Transmission System

5.1.1. Mechatronic System

In (De Silva et al., 1994), a transmission mechanism is presented. The mechanism belongs to the class of continuously variable transmission (CVT). A CVT is a mechanism whose transmission ratio can be continuously changed in an established range, producing a smooth behavior at its output. This pinion-rack CVT is built-in with conventional mechanical elements as a gear pinion, one circular cam, two pairs of racks and two sliders. An advantage of this mechanism is the relative simple mechanical design of its elements such as the racks, gear pinion and sliders. However, a special analysis of the circular cam is necessary, that is because the circular cam is a mechanical device which is fundamental in the performance of the whole system.

The pinion-rack CVT changes its transmission ratio when the distance between the input and output rotation axes is changed. This distance is called “offset” and will be denoted by “e”. Inside the CVT an offset mechanism is integrated. This mechanism is built with a lead screw attached by a nut to the vertical transport cam. Figure 7 shows the built CVT prototype. A detailed explanation about the kinematic and dynamic model of the pinion-rack CVT can be obtained in (Alvarez-Gallegos et al., 2005a).

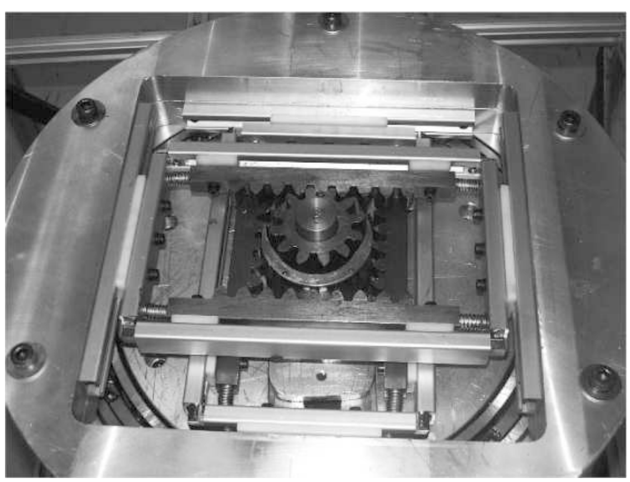

Figure 7: CVT Prototype

5.1.2. Optimization problem.

As it has been previously discussed, in order to obtain the mechanical CVT parameter optimal values, a multi-objective dynamic optimization problem described by equations (6) to (14) is proposed. The dynamic model of the pinion-rack CVT with the state variables $x_1 = \dot{\theta}$, $x_2 = i$, $x_3 = e$, $x_4 = \dot{e}$ and the control signal $u(t)$ is given by equation (8). Also, the vector of design variables is stated as $p = [p_1, p_2, p_3, p_4, p_5, p_6]^T = [N, m, h, e_{max}, K_P, K_I]^T$. A detailed explanation to obtain the performance criteria and objective functions, constraints functions and design variables is available in (Alvarez-Gallegos et al., 2005b). The optimization problem is the following:

$$\min_{p \in R^6} \Phi(x, p, t) = [\Phi_1, \Phi_2]^T \tag{6}$$

where

$$\Phi_1 = \int_0^{10} \left[\frac{1}{p_1} \left(\frac{p_1 p_2 + x_3 \cos\theta_R}{\frac{p_1 p_2}{2} + x_3 \cos\theta_R} \right) \right] dt$$

$$\Phi_2 = \int_0^{10} u^2 dt$$

subject to

$$\begin{aligned}
\dot{x}_1 &= \frac{AT_m + \left[J_1 A \frac{2x_3}{p_1 p_2} \sin\theta_R \right] x_1^2 - T_L - \left[b_2 + b_1 A^2 + J_1 A \frac{2x_4}{p_1 p_2} \cos\theta_R \right] x_1}{J_2 + J_1 A^2} \\
\dot{x}_2 &= \frac{u(t) - (\frac{nK_b}{d}) x_4 - Rx_2}{L} \\
\dot{x}_3 &= x_4 \\
\dot{x}_4 &= \frac{(\frac{nK_f}{d}) x_2 - (b_l + \frac{b_c}{r_p d}) x_4 - \frac{T_m}{r_p} \tan\phi \cos\theta_R}{M + \frac{J_{eq}}{d^2}}
\end{aligned} \tag{7}$$

with PI controller

$$u(t) = -p_5 (x_{ref} - x_1) - p_6 \int_0^t (x_{ref} - x_1) dt \tag{8}$$

and constraints

$$J_1 = \frac{1}{32}\rho\pi p_2^4 \left(p_1 + 2\right)^2 p_1^2 p_3 \tag{9}$$

$$J_2 = \frac{\rho p_3}{4}\left[3\pi r_c^4 - \frac{32}{3}\left(p_4 + p_1 p_2\right)^4 - \pi r_s^4\right] \tag{10}$$

$$A = 1 + \frac{2x_3}{p_1 p_2}\cos\theta_R \tag{11}$$

$$d = r_p \tan\lambda_s \tag{12}$$

$$\theta_R = \frac{1}{2}\arctan\left[\tan\left(2x_1 t - \frac{\pi}{2}\right)\right] \tag{13}$$

$$\begin{aligned}
g_1 &= 0.01 - p_2\left(p_1 - 2.5\right) \leq 0 \\
g_2 &= 6 - \frac{p_3}{p_2} \leq 0 \\
g_3 &= \frac{p_3}{p_2} - 12 \leq 0 \\
g_4 &= p_1 p_2 - p_4 \leq 0 \\
g_5 &= p_4 - \frac{5}{2} p_1 p_2 \leq 0 \\
g_6 &= 12 - p_1 \leq 0 \\
g_7 &= 0.020 - p_3 \leq 0 \\
g_8 &= 0.020 - \left[r_c - \sqrt{2}(p_4 + p_1 p_2)\right] \leq 0 \\
g_9 &= 0.0254 - p_1 p_2 \leq 0
\end{aligned} \tag{14}$$

5.2. Five-bar parallel robot

5.2.1. Mechatronic System

Many robots have their links sequentially connected starting from a fixed base. The last link in the chain is connected from one end to a previous link but is free from the other end, resulting in an open link chain or *open kinematic chain*. It is common that each joint of the links is connected by actuators (actuated joints) in order to provide the motion of the robot. Those robots are generally known as *open-chain robots* or *serial robots*. Other robot kinematic configurations have their links connected in serial as well as in parallel combinations forming one or more closed-link loops. In that configuration, not all joints are actuated. Those robots are called *closed-chain robots* or *parallel robots*.

Parallel robots have the advantages of high stiffness, speed, acceleration, good dynamic characteristics and precise positioning capabilities (Hunt, 1983). Nevertheless, their disadvantages due to their parallel configurations include limited workspace and singular configurations[1] as well as the lack of well developed tools for the analysis, synthesis, control and trajectory planning. So, it is a well-known fact that the parallel robot design while optimizing performance is a huge challenge due to the highly nonlinear system dynamics and the presence of many singularities. Consequently, it is not an easy task to find the mechanical and the control parameters in such a way that they simultaneously optimize the system performance of the mechanical system and the control system. Here, a multi-objective dynamic optimization problem is stated to design both the mechanical and control system of a five-bar parallel robot.

The five-bar parallel robot is the parallel robot with the minimal degrees of freedom (DoFs) in the field, which can be used for positioning of a desired point or for tracking of a desired trajectory on a region of a plane that is known as the workspace. The five-bar parallel robot consists of five links connected end to end by five revolute joints, two of which are actuated and are connected to the base of the robot. Figure 8 shows the five-bar parallel robot built at Cinvestav-IPN. A detailed explanation about the kinematic and dynamic model of the five-bar parallel robot can be obtained from (Liu et al., 2006) and (Ghorbel et al., 2000), respectively.

5.2.2. Optimization problem.

The multi-objective dynamic optimization problem consists of finding the optimal design variable vector $p^* \in R^{51}$ which involves the optimal geometries of the parallel robot's links (structure design) and the optimal PID controller gains (control design), that simultaneously minimize a performance function vector (16), subject to constraints at the parallel robot dynamic model (18), the limits of the motor torque (19), the Grashof criterion (20), the geometric limits of the links (21), the Cartesian position $\bar{c} = [\bar{x}_p, \bar{y}_p]^T$ and velocity $\dot{\bar{c}} = [\dot{\bar{x}}_p, \dot{\bar{y}}_p]^T$ of the desired trajectory to be executed by the end effector (22).

[1]A singular configuration is a position of the robot where the subsequent behavior cannot be predicted or becomes nondeterministic. Hence, the robot can not be further controlled.

Figure 8: Five-bar parallel robot

The performance function vector includes the manipulability measure Φ_1 and the position error Φ_2. The position error of the robot is required in order to follow a desired trajectory in the workspace of the parallel robot and the manipulability measure is included in order to move away from singularity configurations. When Φ_1 is minimized, the parallel robot moves away from singularity configurations. When $\Phi2$ is minimized, the position error of the end-effector to follow the desired trajectory is minimized.

In Fig. 9 the geometric parameters of the $i-th$ link are shown. When the geometric parameters are modified, the link shape can be changed. The thickness of the $i-th$link is represented by e_{s_i}.

In this paper both the structure design and the control design are simultaneously considered in order to get an appropriate system performance. Hence, the geometric parameters $p_s \in R^{45}$ of the links (Figure 9) and the PID control gains $p_c \in R^6$ are the design variable vector p^*. So, the design variable vector p^* is stated as:

$$p = [p_1, p_2, ..., p_{50}, p_{51}]^T = [p_s,\ p_c]^T \in R^{51} \tag{15}$$

where:

$$\begin{aligned} p_s &= [\ a_{s_1} \cdots a_{s_5},\ \ b_{s_1} \cdots b_{s_4},\ \ c_{s_1} \cdots c_{s_4},\ \ d_{s_1} \cdots d_{s_4},\ \ e_{s_1} \cdots e_{s_4},\ \ f_{s_1} \cdots f_{s_4}, \\ &\quad\ g_{s_1} \cdots g_{s_4},\ \ h_{s_1} \cdots h_{s_4},\ \ i_{s_1} \cdots i_{s_4},\ \ j_{s_1} \cdots j_{s_4},\ \ k_{s_1} \cdots k_{s_4}\ ]^T \in R^{45} \\ p_c &= [\ k_{p_1},\ \ k_{p_2},\ \ k_{i_1},\ \ k_{i_2},\ \ k_{d_1},\ \ k_{d_2}\ ]^T \in R^6 \end{aligned}$$

Therefore, the multi-objective dynamic optimization problem can be formulated as follows:

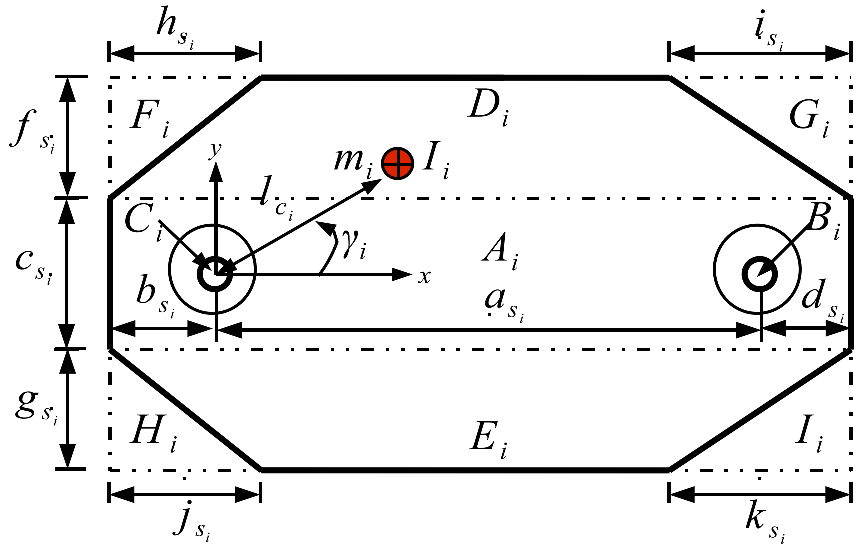

Figure 9: Schematic representation of a link for the five-bar parallel robot.

$$\min_{p\in R^{51}} \Phi(x,p,t) = [\Phi_1, \Phi_2]^T \tag{16}$$

where

$$\Phi_1 = \int_{t_0=0}^{5} \frac{2\left|\det J^{-1}\right|}{tr\left(J^{-1}J^{-T}\right)}dt$$

$$\Phi_2 = \int_{t_f=0}^{5} e^T e dt$$

$$J = \begin{bmatrix} -\frac{p_1(\sin(q_2+q_4))\sin q_3}{\sin(q_1-q_2+q_3-q_4)} & \frac{p_2(\sin(q_1+q_3))\sin q_4}{\sin(q_1-q_2+q_3-q_4)} \\ \frac{p_1(\cos(q_2+q_4))\sin q_3}{\sin(q_1-q_2+q_3-q_4)} & -\frac{p_2(\cos(q_1+q_3))\sin q_4}{\sin(q_1-q_2+q_3-q_4)} \end{bmatrix} \tag{17}$$

subject to

1.- The system dynamics (Ghorbel et al., 2000) and the PID controller:

$$\begin{array}{l} \dot{x} = \frac{dx}{dt} = f\left(x, \bar{x}, p, u\right) \\ u_1 = p_{46}e_1 + p_{47}\int_0^{t_f} e_1 dt + p_{48}\dot{e}_1 \\ u_2 = p_{49}e_2 + p_{50}\int_0^{t_f} e_2 dt + p_{51}\dot{e}_2 \\ x(0) = [1.7453, 0, 0, 0, 0, 0]^T \end{array} \tag{18}$$

2.- The maximum applied torque for the motor:

$$\begin{array}{ll} g_1 : |u_1(t)| \leq 5 & 0 \leq t \leq 5 \\ g_2 : |u_2(t)| \leq 5 & 0 \leq t \leq 5 \end{array} \tag{19}$$

3.- The Grashof criterion:

$$\begin{array}{l} g_3 : a_{s_5} + p_1 + p_2 - p_3 - p_4 < 0 \\ g_4 : p_1 + p_2 - p_3 < 0 \\ g_5 : p_1 + p_2 - p_4 < 0 \\ g_6 : p_4 - p_5 < 0 \\ g_7 : p_3 - p_5 < 0 \\ g_8 : p_5 - 0.5 \leq 0 \\ g_9 : -p_1 + 0.035 < 0 \\ g_{10} : -p_2 + 0.035 < 0 \end{array} \tag{20}$$

4.- The geometric limits of the links:

$$\begin{array}{ll} g_{10+i} : 0.01905 \leq p_{5+i} \leq 0.3 & \\ g_{14+i} : 0.0381 \leq p_{9+i} \leq 0.10 & \\ g_{18+i} : 0.3 \leq p_{13+i} \leq 0.3 & \\ g_{22+i} : 0.00635 \leq p_{17+i} \leq 0.03 & \\ g_{26+i} : 0 \leq p_{21+i} \leq 0.1 & \text{for } i = 1, \ ..., \ 4 \\ g_{30+i} : 0 \leq p_{25+i} \leq 0.1 & \\ g_{34+i} : p_{29+i} + p_{33+i} \leq p_{5+i} + p_i + p_{13+i} & \\ g_{38+i} : p_{37+i} + p_{41+i} \leq p_{5+i} + p_i + p_{13+i} & \end{array} \tag{21}$$

5.- The desired trajectory $\bar{c} = [\bar{x}_p, \bar{y}_p]^T$ and velocity $\dot{\bar{c}} = [\dot{\bar{x}}_p, \dot{\bar{y}}_p]^T$:

$$\begin{array}{l} h_1 : \bar{x}_p = 0.15 + 0.1 \cos(1.2566t) \\ h_2 : \bar{y}_p = 0.3 + 0.1 \sin(1.2566t) \\ h_3 : \dot{\bar{x}}_p = -0.1256 \sin(1.2566t) \\ h_4 : \dot{\bar{y}}_p = 0.1256 \cos(1.2566t) \end{array} \tag{22}$$

Therefore q_1, q_2 and q_3, q_4 are the actuated and unactuated angles of the parallel robot, respectively, $x = [q_1, \ q_2, \ \dot{q}_1, \ \dot{q}_2, \ \int_0^{t_f} e_1 dt, \ \int_0^{t_f} e_2 dt]^T = [x_1, ..., x_6]^T \in R^6$ and $\bar{x} = [\bar{x}_1, \bar{x}_2, \bar{x}_3, \bar{x}_4, \bar{x}_5, \bar{x}_6,]^T \in R^6$ are the current and desired state variable vectors, $u = [u_1, u_2]^T$ is the input torque vector, $e = [e_1, e_2]^T = [\bar{x}_1 - x_1, \bar{x}_2 - x_2]^T$ and $\dot{e} = [\dot{e}_1, \dot{e}_2]^T = [\bar{x}_3 - x_3, \bar{x}_4 - x_4]^T$

are the angular position and velocity error vectors of the actuated angles, respectively, t is the time, t_0, t_f are the initial and final time, and J is the Jacobian matrix of the five-bar parallel robot.

The Cartesian position $\bar{c} = [\bar{x}_p, \bar{y}_p]^T$ and velocity $\dot{\bar{c}} = [\dot{\bar{x}}_p, \dot{\bar{y}}_p]^T$ of the desired trajectory must be transformed into the joint position and velocity of the parallel robot because the PID controller u_i in equation (18) requires the desired trajectory and velocity in the joint space ($\bar{x}$). This transformation is expressed in equation (23).

$$\begin{aligned} \bar{x}_i &= \tan^{-1}\left(\frac{\sigma_i\sqrt{1-(A_i)^2}}{A_i}\right) + q_p \text{ for } i = 1, 2 \\ [\dot{\bar{x}}_3, \dot{\bar{x}}_4]^T &= J\dot{\bar{c}} \end{aligned} \tag{23}$$

where $A_i = \frac{-p_{i+2}^2 + p_i^2 + \|\vec{a}_p\|^2}{2p_i\|\vec{a}_p\|}$, $q_p = \tan^{-1}\left(\frac{\bar{y}_p - O_{i_y}}{\bar{x}_p - O_{i_x}}\right)$, $\|\vec{a}_p\| = \|c - O_i\|$, $O_1 = [0, 0]^T$, $O_2 = [p_5, 0]^T$ and $\sigma_i = \pm 1$.

In the optimization problem (16)-(22) there are fifty one design variables 15, forty five design variables corresponding to the structure design p_s and the other six design variables corresponding to the control design p_c. In addition, there are forty inequality static linear constraints (20) and (21), two inequality smooth dynamic nonlinear constraints (19) and four equality smooth dynamic nonlinear constraints (22) (the dynamic constraints depend on the time). There is another highly nonlinear dynamic constraint due to the dynamic behavior of the parallel robot (18). This constraint is represented by smooth nonlinear differential equations which are solved by numerical methods such as the Runge-Kutta method. The Runge-Kutta method requires an integration step Δt to solve the nonlinear differential equations. In this problem, an integration step of $\Delta t = 0.005$ is used. So, the time interval $[t_0, t_f] = [0, 5]$ is divided into $n_{\Delta t}$ ($n_{\Delta t} = \frac{t_f - t_0}{\Delta t} = 1000$) intervals ($t = 0, t = 0.005, ..., t = 4.995, t = 5$). It is important to mention that in this problem there are dynamic constraints which must be evaluated at $n_{\Delta t}$ different times. So, each dynamic constraint must be evaluated $n_{\Delta t}$ times for each individual in the population.

A detailed explanation to obtain the performance criteria and objective functions, constraints functions and design variables is available in (Villarreal-Cervantes et al., 2010).

5.3. Summary of features of the concurrent design problems

For the CVT problem there are four design variables for the structure and two design variables corresponding to the controller. Moreover, the whole set of inequality constraints are static: eight nonlinear constraints and one linear constraint. Finally, there is one equality constraint.

For the five-bar parallel robot there are forty five design variables for the structure and six design variables corresponding to the controller. In addition, there are four equality dynamic nonlinear constraints, forty inequality static linear constraints and two inequality dynamic nonlinear constraints.

On the other hand, the singularity configurations of the robot comprise a problem when the design of a parallel robot is formulated as an optimization problem. These configurations are particular positions of the end-effector, for which parallel robots lose their inherent infinite rigidity and in which 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 (see Fig. 10). In the singularity regions neither the dynamic model (NLDEs) nor the Jacobian matrix can be computed. Hence, the problems of using NLPTs arise when the initial condition is inside the singularity region (see Fig. 10a) or when the search direction goes to a singularity region (see Fig. 10b), since neither the gradient nor the sensitivity can be computed in order to get the next search direction. The only way to avoid the singularity regions (there is not only one single singularity region) is using barrier functions (Bazaraa and Sherali, 1993) to reject the search direction from the singularity configuration. Nevertheless, the main problem of using barrier function is that the initial solution must not be in the singularity region and in addition, there is a possibility that the search direction goes to another solution (see Fig. 10c). Therefore, the NLPTs, which are single shooting approaches, can not guarantee convergence to a solution. In addition, it is a difficult task to find the initial conditions where the system dynamics is well defined and without singularities.

It is important to remark that in both optimization problems the dynamic model of the system is included. The dynamic model of the system is represented by nonlinear differential equations. In order to solve this dynamic model, a numerical method is utilized. In this numerical method an integration step (Δt) is required to divide the time interval $[t_0, t_f]$ into finite intervals. So, the dynamic constraints must be evaluated at each time interval, requiring $n_{\Delta t}$ ($n_{\Delta t} = \frac{t_f - t_0}{\Delta t}$) evaluations for each dynamic constraint

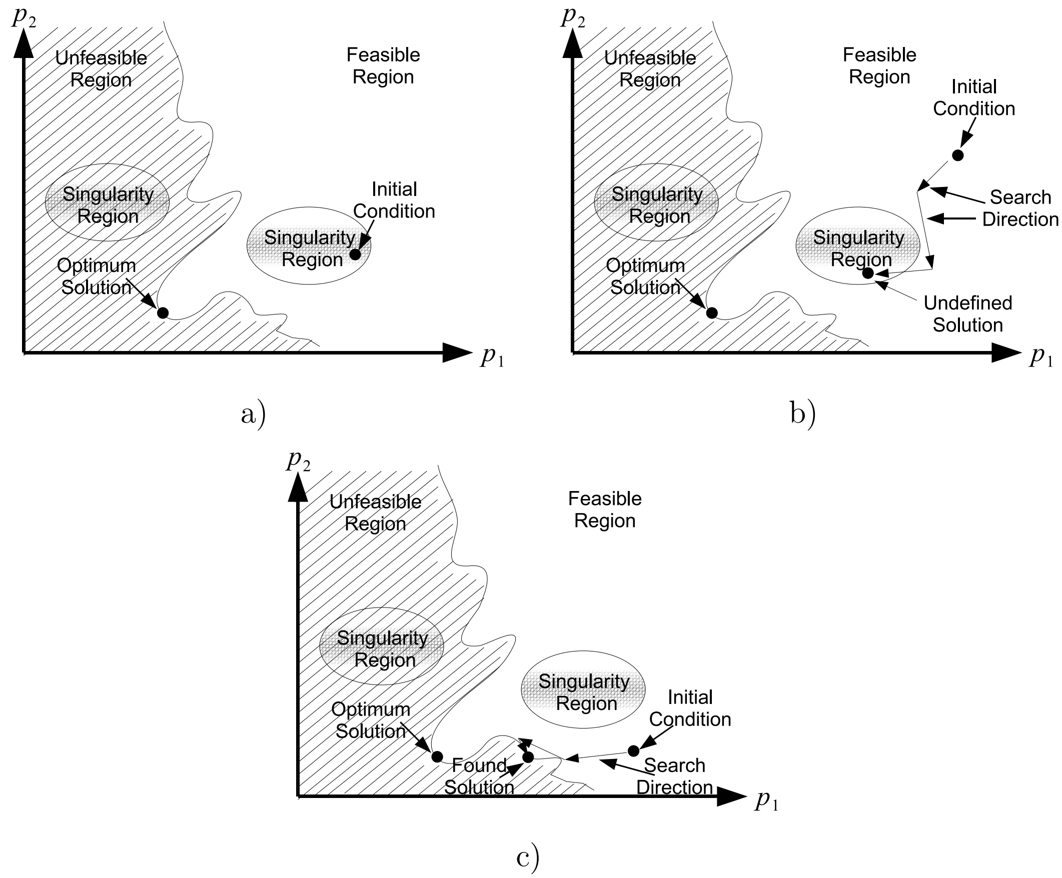

Figure 10: Objective function of a two-dimensional design space.

of each individual in the population. The dynamic constraints are only included into the dynamic optimization problem stated for the five-bar parallel robot. Moreover, the dynamics of the five-bar parallel robot is more complex than that of the CVT. Taking everything into account, an evaluation of the dynamic model of the five-bar parallel robot requires more time than the CVT, which is reflected in the results obtained in the numerical experiments presented in the next Section.

6. Numerical results

6.1. Experimental design

In order to assess the performance of the proposed approach, which aims to favor reconfigurability in concurrent optimal mechatronic design by the introduction of an evolutionary-based method , the two problems described in Section 5 are solved by the adapted DE algorithm detailed in Section 4. Based on the fact that the evaluation of one single solution takes several minutes, 5 independent runs were performed for each experiment and the time in hours per run is reported. All independent runs were performed on the same computer platform: a PC with a 2.8GHz Pentium IV processor with 1 GB of memory using Matlab® 7.6.0 Release 2008*a*. Five experiments were carried out for each mechatronic design. In each one of them, the same parameter values were used with the exception of the NS parameter, which was varied aiming to analyze the influence of the crowding distance factor in the parametric reconfiguration improvement of the two real-world mechatronic concurrent designs under study.

The fixed parameter values were the following for the CVT design: $NP = 200$, $MAX\_GEN = 100$, F was generated at random at each generation within the following range [0.3,0.9] and CR was also generated at random at each generation within the range [0.8,1.0]. The aim to generate the F and CR values at random at each generation was to promote the generation of diverse search directions and also diverse combinations between the target and mutant vectors. The parameters for the five-bar parallel robot system were equal for F and CR, just NP and MAX_GEN were modified adopting values of 100 and 6000, respectively. It is clear from the parameter definition that the second problem required more time to be solved due to its complexity. As a consequence, each single run required more time to finish as it will be detailed later in the paper.

The NS parameter took the following values: $NS = 0.1$ in Experiment 1, which implies a intensive use of the selection from the archive based on the crowding distance factor value, i.e., low exploration of promising solutions and a very high reconfigurability promotion, $NS = 0.3$ in Experiment 2, which means a more frequent use of the selection based on the crowding factor, i.e., moderate exploration of promising solutions and a high reconfigurability promotion, $NS = 0.5$ in Experiment 3, which means a similar use of the normal selection from the current population and the selection from the archive based on the crowding distance, i.e., equal exploration of promising solutions and reconfigurability promotion and, finally, $NS = 0.8$ in Experiment 4, which means a low use of the selection by crowding distance, i.e., a high exploration of promising solutions and a low reconfigurability promotion.

A fifth experiment consisted on comparing the results of the previous experiments with a version without the crowding mechanism, i.e., no reconfigurability promotion. In order to allow a fair comparison, the same exact evolutionary algorithm with the same parameter values was utilized.

The results obtained for each mechatronic concurrent design are presented in the next subsections.

6.2. CVT System

Table 1 contains the numerical results obtained in the first four experiments where the NS parameter value was varied. The number of non-dominated solutions per single run and the time required are also included. Finally, the average number of non-dominated solutions and the average time per run per experiment are calculated.

Regarding Experiment 5, where the DE algorithm without the reconfigurability promotion mechanism is tested, the obtained results are presented in Table 2.

For a better visualization of the Pareto fronts obtained in each experiment, in Figure 11 the five fronts obtained in each single run per experiment are filtered into a single one and the number of non-dominated solutions in each filtered front is given in Table 3.

A first observation of the overall results is that all the final solutions obtained in the five experiments for the CVT design problem were feasible.

From the summary of results in Table 1, the first four fronts in Figure 11 and the first four rows in Table 3 different findings were observed:

1. The average number of non-dominated solutions per single run increased (almost twice) from Experiment 4 ($NS = 0.8$) to Experiment 1 ($NS = 0.1$), while the average time required was almost the same (see Table 1).
2. In the same regard, the number of solutions in the filtered Pareto front in Experiment 1 was more than twice the number of solutions in the filtered front in Experiment 4 (see Table 3 and Figure 11).
3. There was an increase in the average number of non-dominated solutions in Experiments 2 ($NS = 0.3$) and 3 ($NS = 0.5$) with respect to

CVT design. Experiment 1 $NS = 0.1$

Run	Non-dominated solutions	Time/Hrs
1	57	12.64
2	51	13.04
3	53	12.90
4	49	13.65
5	41	13.21
Average	50.2	13.08

CVT design. Experiment 2 $NS = 0.3$

Run	Non-dominated solutions	Time/Hrs
1	26	13.90
2	35	15.49
3	48	16.09
4	56	15.96
5	36	15.99
Average	40.2	15.68

CVT design. Experiment 3 $NS = 0.5$

Run	Non-dominated solutions	Time/Hrs
1	30	12.61
2	44	12.54
3	43	13.08
4	41	12.65
5	43	12.59
Average	40.2	12.69

CVT design. Experiment 4 $NS = 0.8$

Run	Non-dominated solutions	Time/Hrs
1	35	12.98
2	23	13.17
3	31	13.14
4	32	12.85
5	21	13.04
Average	28.4	13.03

Table 1: Number of non-dominated solutions and time required to find them at each independent run in the four experiments for the CVT concurrent design.

CVT Design. Experiment 5

Run	Non-dominated solutions	Time/Hrs
1	19	14.79
2	18	15.71
3	15	16.16
4	17	15.08
5	20	15.36
Average	17.8	15.42

Table 2: Number of non-dominated solutions and time required to find them at each independent run in the DE version without the reconfigurability promotion mechanism for the CVT concurrent design.

Number of non-dominated solutions per experiment for the CVT design.

Experiment	Non-dominated solutions
1	108
2	78
3	75
4	41
5	28

Table 3: Number of non-dominated solutions in the filtered Pareto fronts per each experiment for the CVT design.

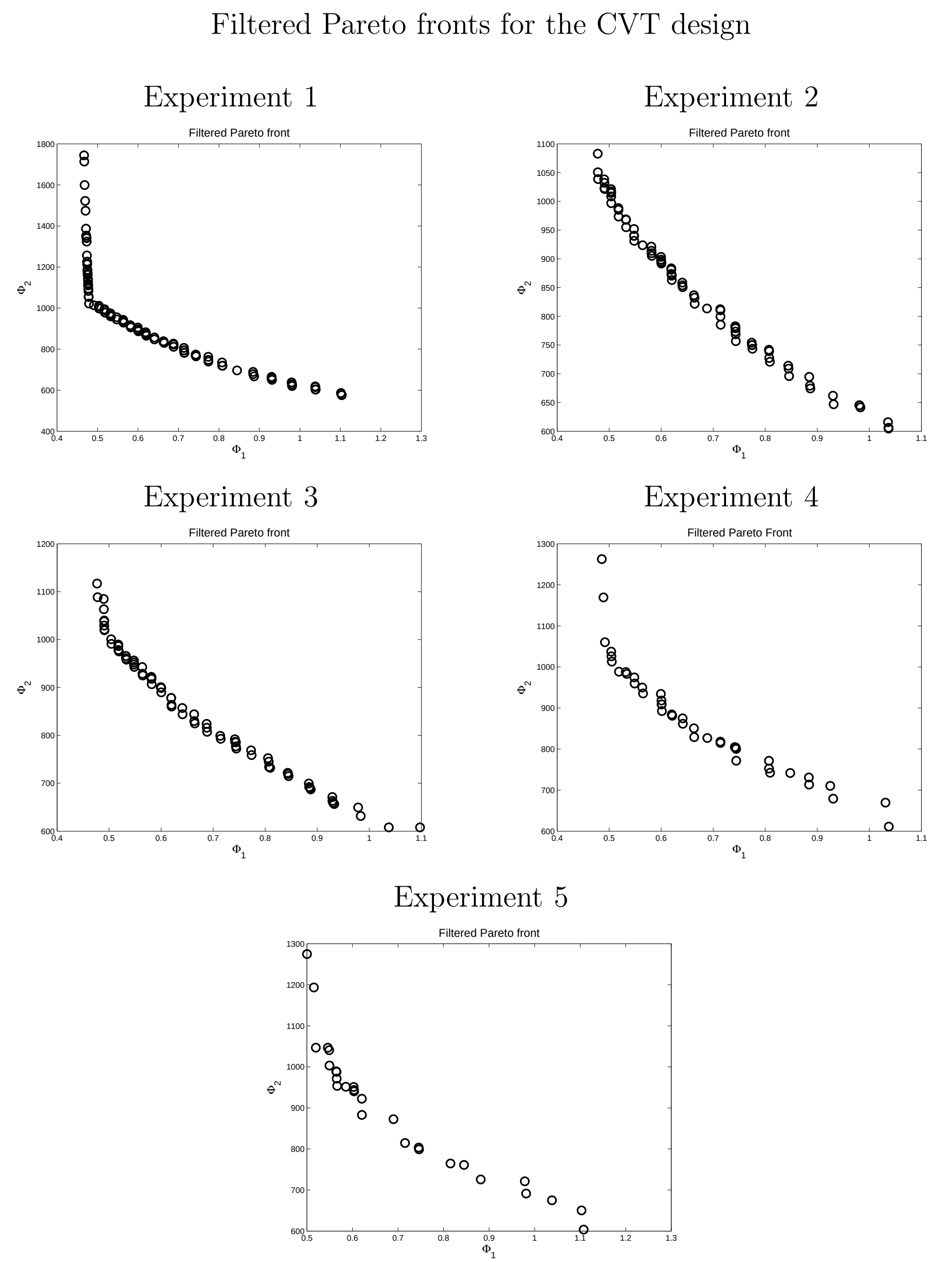

Figure 11: Filtered Pareto fronts per each one of the five experiments for the CVT design.

Experiment 4 ($NS = 0.8$). On the other hand, the average time in Experiment 2 was the highest (15.68 Hours) of the four experiments whereas the corresponding average time in Experiment 3 was the lowest (12.69 Hours).

4. Experiment 1 showed the convenience of the massive usage of the reconfigurability mechanism based on crowding distance. As a result of the higher number of non-dominated solutions (see Table 3) in Figure 11, for Experiment 1, a wider range of values for objective $\Phi_2 \in [600, 1800]$ was obtained. In contrast, none of the remaining experiments (2, 3 and 4) was able to generate solutions in the upper left part of the Pareto front of this design problem.

The results obtained with the DE algorithm without the reconfigurability promotion mechanism (Experiment 5) presented in Table 2, in the last filtered Pareto front in Figure 11 and in Table 3 suggest that the performance of this version was surpassed by that of the previous four experiments, except for the average time, which is slightly higher in Experiment 2 ($NS = 0.3$). Furthermore, The filtered Pareto fronts in Figure 11 showed that the lack of the reconfigurability mechanism affects the capability of the algorithm to generate solutions with a lower value for objective function Φ_1.

It is worth noticing that the filtered Pareto front from Experiment 1 in Figure 11 provides the design engineer a very rich set of possible solutions because, for a similar mechanical design (e.g., objective $\Phi_1 \approx 0.5$) there is a rich set of solutions with different controller values (objective $\Phi_2 \in [1000, 1800]$). Taking into account that mechanical elements are more expensive (i.e., a gear pinion must be built by a CNC machine) than controller implementation, it is better to have a large set of controller gains, maintaining a constant size of mechanical elements. Therefore, from the design engineer's point of view, solutions in the mentioned area represent a smaller investment on the final prototype.

As a final conclusion for this mechatronic design, the massive use of a selection based on the crowding distance value allowed the DE algorithm to generate solutions which favor the reconfigurability of the design without increasing the computational time required by the approach.

6.3. Five-bar Parallel Robot System

The results provided in the first four experiments are presented in Table 4, where each one of them considered a different NS value. As in the case of the CVT design, the number of non-dominated solutions per single run and the time required are included. However, for this problem the number of feasible solutions in the final Pareto front obtained is also included because not all of them satisfied the constraints of the design problem. The average numbers for these three measures are calculated in the same Table 4.

The results of the DE algorithm without the reconfigurability promotion mechanism for this second design problem are presented in Table 5.

The five filtered Pareto fronts (one per each experiment) are presented in Figure 12 and the number of non-dominated solutions in each one of them are given in Table 6. Based on the fact that Figure 12 does not completely reflect the differences in behaviors of each experiment, in Figure 13 a zoom view of the five filtered fronts is plotted.

Interesting performances were found in the results of the first four experiments:

1. The highest average number of non-dominated solutions was obtained in Experiment 2 ($NS = 0.3$), and it was almost twice the average value reported in Experiment 4 ($NS = 0.8$). In these two experiments, the time required by the DE algorithm is very similar (46.66 hours in Experiment 2 and 46.27 hours in Experiment 4) and the percentage of feasible solutions is slightly worst in Experiment 2 because one run was unable to converge to the feasible region of the search space (see Table 4).
2. The lowest NS value in Experiment 1 ($NS = 0.1$) caused the generation of a lower number of non-dominated solutions and the number of feasible solutions was also decreased. Although the average time spent was slightly less than the reported in the rest of the experiments (see Table 4).
3. The number of non-dominated solutions in the filtered Pareto front of Experiment 2 ($NS = 0.3$) was the highest among all the remaining experiments (see Table 6 and Figure 12).
4. Even though Experiment 4 ($NS = 0.8$) in Figure 12 seems to provide more diverse solutions in the upper left part of the Pareto front generated with respect to Experiment 2 ($NS = 0.3$), Figure 13 shows that the non-dominated solutions from Experiment 2 dominate those of Experiment 4.
5. The intensive use of the reconfigurability promotion mechanism in Experiment 2 provides a considerable number of different values for ob-

Robot design. Experiment 1 $NS = 0.1$

Run	Non-dominated solutions	Time/Hrs	% Feasible Solution in $\vec{s}_{MAX\_GEN}$
1	113	44.35	51
2	196	44.71	100
3	185	45.89	100
4	0	43.53	0
5	162	43.72	19
Average	131.2	44.43	54

Robot design. Experiment 2 $NS = 0.3$

1	244	47.90	100
2	195	47.63	100
3	0	46.26	0
4	267	45.36	100
5	255	46.20	100
Average	192.2	46.66	80

Robot design. Experiment 3 $NS = 0.5$

1	153	46.33	100
2	189	46.45	100
3	167	47.45	100
4	161	46.93	100
5	164	46.91	100
Average	166.8	46.81	100

Robot design. Experiment 4 $NS = 0.8$

1	96	45.60	100
2	109	46.23	100
3	113	46.06	100
4	93	45.72	100
5	126	47.78	100
Average	107.4	46.27	100

Table 4: Number of non-dominated solutions, time required to find them and percentage of feasible solutions at each independent run in the four experiments for the parallel robot concurrent design.

Robot design. Experiment 5

Run	Non-dominated solutions	Time/Hrs	% Feasible Solution in $\vec{s}_{MAX\_GEN}$
1	24	46.76	100
2	35	48.60	100
3	44	48.88	100
4	35	46.08	100
5	33	46.23	100
Average	34.2	47.3	100

Table 5: Number of non-dominated solutions, time required to find them and percentage of feasible solutions in the final front at each independent run in the DE version without the reconfigurability promotion mechanism for the robot concurrent design.

jective $\Phi_1 \in [-0.95, -0.88]$, whereas the other three experiments were less competitive in this regard.

Experiment 5 showed that the use of the reconfigurability promotion mechanism is highly important to provide diverse solutions to the designer (see the very low number of non-dominated solutions reported in Tables 5 and 6 and the poorly distributed Pareto fronts in filtered front in Figures 12 and 13). It is important to remark that the approximation to the feasible region is affected by the use of the selection based on crowding distance because the feasibility of solutions is not directly related with this criteria. However, with a correct value for the NS parameter as in Experiment 2, this shortcoming can be controlled.

An interesting finding in this non-iterative concurrent design of the five-bar parallel robot was that the DE algorithm was able to provide the design engineer a diverse set of mechanical and controller designs that satisfy different trade-offs between both objectives e.g., the mechanical performance (objective $\Phi_1 \in [-0.93, -0.89]$) and the controller performance (objective $\Phi_2 \in [0, 1.5]$), see Figure 12. From a designer point of view, a design methodology where a variety of designs fulfills different trade-offs between the position error and the manipulability measure, is recommended i.e., solutions near to $[\Phi 1, \Phi 2] = [-1, 0]$. So, the design engineer could find the solution that satisfies the required design specification, i.e., the solution must fulfill the required design specification. A design that exceeds the required design specification tends to raise the cost of the final product, whereas a design that is below of the required design specification tends to fail for the entrusted task.

The control performance (Φ_2) indicates the accuracy of the trajectory tracking. Then, a value near to zero means that the position error is almost zero. On the other hand, the mechanical performance (Φ_1) indicates a distance measure from singularity configurations of the parallel robot. Then, a value near to minus one (-1) means that the parallel robot configuration is away from singularities.

The overall results of the proposed DE algorithm in this five-bar parallel robot non-iterative concurrent design, which was more difficult to solve compared to the CVT design problem previously discussed, suggest that the intensive use of the reconfigurability mechanism allows the algorithm to generate more non-dominated solutions. However, its use must be calibrated in order to keep the original capability of the approach to generate feasible designs.

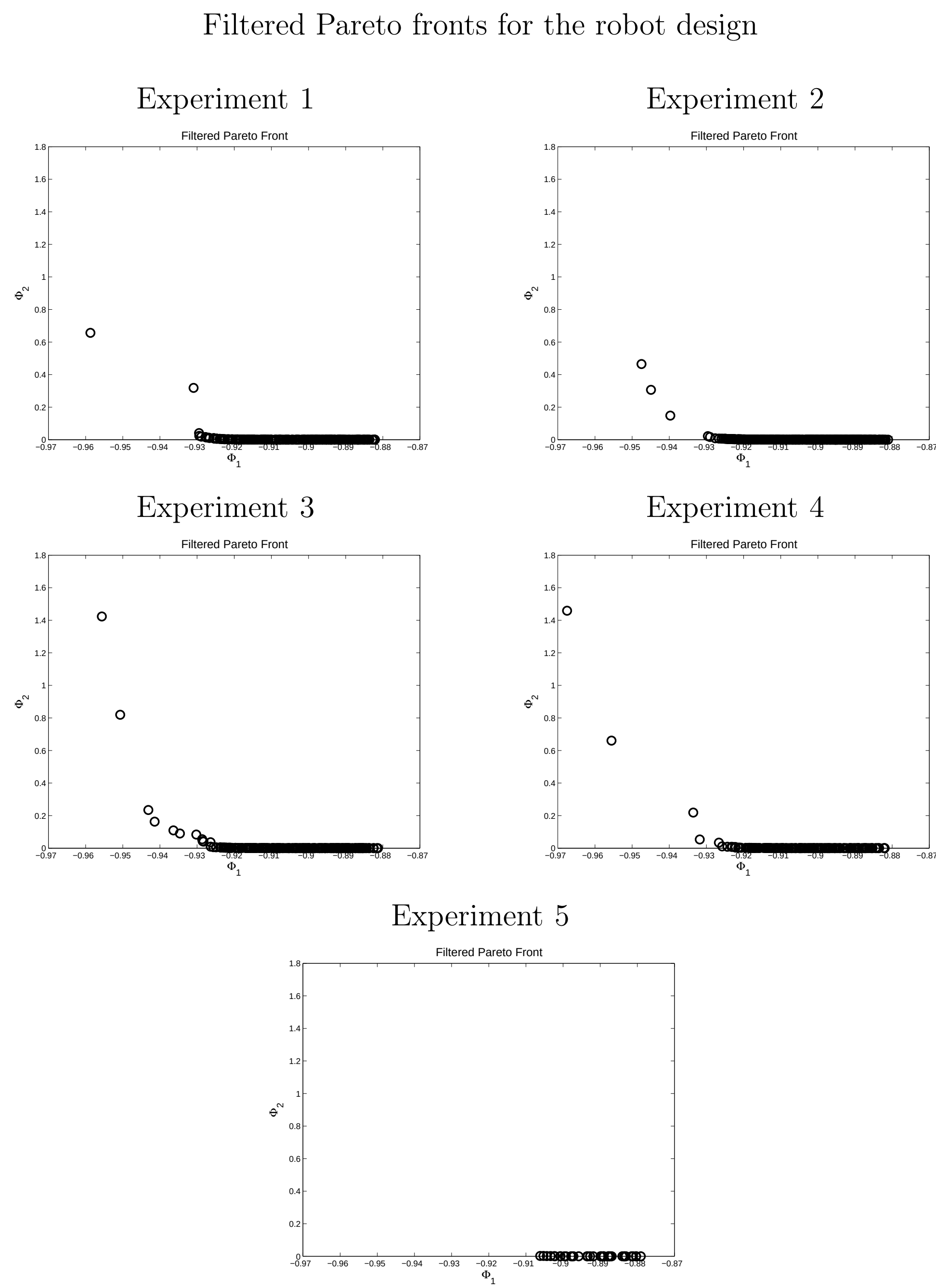

Figure 12: Filtered Pareto fronts per each one of the five experiments for the robot design.

Number of non-dominated solutions per experiment for the robot design.	
Experiment	Non-dominated solutions
1	206
2	296
3	222
4	145
5	38

Table 6: Number of non-dominated solutions in the filtered Pareto fronts per each experiment for the robot design.

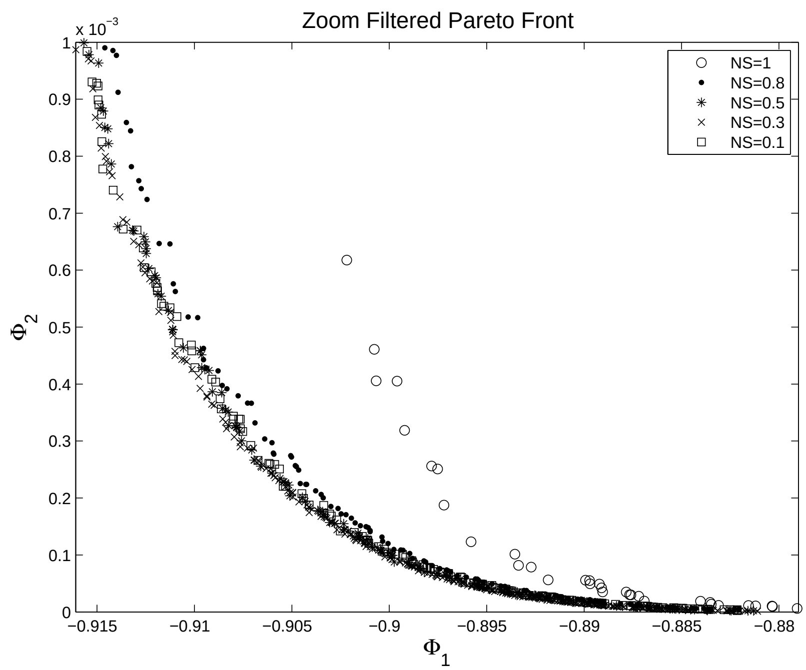

Figure 13: Zoom of the set of Filtered Pareto fronts from Figure 12

7. Conclusions and Future Work

In this work the reconfigurability feature of the non-iterative concurrent mechatronic design methodology was improved by using an evolutionary-based approach. The use of the crowding distance, Pareto dominance concepts coupled with a simple but effective constraint-handling technique and an external archive were added to a Differential Evolution variant called DE/rand/1/bin. The selection of the vectors to generate the trial vector was done in two different ways based on the value of a user-defined parameter called Normal Selection (NS) which determined the percentage of generations where selections were made as in traditional DE/rand/1/bin (three randomly chosen vectors from the current population). The remaining percentage $(1 - NS)$ selections were made from the archive where the non-dominated solutions are stored by using the crowding distance as a criterion. In other words, the NS parameter controlled the time dedicated within the search process to look for feasible and high-quality solutions before switching to improving the parametric reconfiguration of the design, i.e., an extended and well-distributed Pareto front.

The overall expected effect was to find a trade-off between the search of a good approximation to a sub-optimal Pareto front and a good distribution of solutions within it. This combination of effects precisely improved the parametric reconfiguration property of the design in order to provide the designer with an adequate set of possibilities.

The proposed algorithm was used to optimize the design of two complex mechatronic systems: A CVT and a Five-bar parallel robot. In both designs the mechanic and also the controller design were considered in a bi-objective optimization problem.

The performance observed by the proposed algorithm showed that the reconfigurability mechanism (controlled by the NS parameter) must be frequently used (i.e., low NS values are preferred) in order to promote the generation of more non-dominated solutions in the final Pareto front obtained without affecting the computational time required. However, if the problem to be solved is highly constrained, the use of the reconfigurability mechanism must be used with a lower frequency (i.e., slightly higher NS values are required) in order to allow the traditional DE selection mechanism to generate feasible mechatronic designs.

The future paths of research consider the definition of an improved set of criteria which considers feasibility in the reconfigurability promotion mechanism. Moreover, we will test our approach in other types of mechatronic systems in which more than two objectives (besides mechanical and controller designs) are considered.

Acknowledgments

The first author acknowledges support from COFAA and SIP from IPN. The second author acknowledges support from the Mexican Consejo Nacional de Ciencia y Tecnología (CONACyT) through project No. 79809. The third and fifth authors acknowledge support from CONACyT, project 084060. The fourth author acknowledges support from CONACyT project No. 103570. The last author acknowledges support from CONACyT through a scholarship to pursue a PhD degree at CINVESTAV-IPN's Electrical Engineering Department.

References

References

Alvarez-Gallegos, J., Cruz-Villar, C., Portilla-Flores, E., 2005a. Parametric Optimal Design of a pinion-rack based Continuously Variable Transmission. In: Proceedings of the 2005 IEEE/ASME International Conference on Advanced Intelligent Mechatronics. Monterey, California, USA, pp. 899–904.

Alvarez-Gallegos, J., Cruz-Villar, C. A., Portilla-Flores, E. A., 2005b. Evolutionary Dynamic Optimization of a Continuously Variable Transmission for Mechanical Efficiency Maximization. In: A. Gelbukh, A. d. A., Terashima-Marín, H. (Eds.), MICAI 2005: Advances in Artificial Intelligence, Lectures Notes in Artificial Intelligence Vol. 3789. Springer, pp. 1093–1102.

Bazaraa, M. S., Sherali, C. M., 1993. Nonlinear programming. John Wiley & Sons.

Coello Coello, C. A., Lamont, G. B., Van Veldhuizen, D. A., September 2007. Evolutionary Algorithms for Solving Multi-Objective Problems, 2nd Edition. Springer, New York, iSBN 978-0-387-33254-3.

Cruz-Villar, C., Alvarez-Gallegos, J., Villarreal-Cervantes, M., March 2009. Concurrent redesign of an underactuated robot manipulator. Mechatronics 19 (2), 178–183.

De Silva, C., Schultz, M., Dolejsi, E., 1994. Kinematic analysis and design of a continuously variable transmission. Mech. Mach. Theory 29 (1), 149–167.

Deb, K., 2000. An Efficient Constraint Handling Method for Genetic Algorithms. Computer Methods in Applied Mechanics and Engineering 186 (2/4), 311–338.

Deb, K., Pratap, A., Agarwal, S., Meyarivan, T., April 2002. A Fast and Elitist Multiobjective Genetic Algorithm: NSGA–II. IEEE Transactions on Evolutionary Computation 6 (2), 182–197.

Eiben, A., Smith, J. E., 2003. Introduction to Evolutionary Computing. Natural Computing Series. Springer Verlag.

Fonseca, C. M., Fleming, P. J., 1996. Nonlinear System Identification with Multiobjective Genetic Algorithms. In: Proceedings of the 13th World Congress of IFAC. San Francisco, California, pp. 187–192.

Ghorbel, F., Chételat, O., Gunawardana, R., Longchamp, R., September 2000. Modeling and set point control of closed-chain mechanisms: theory and experiment. IEEE Transactions on Control Systems Technology 8 (5), 801–815.

Huang, V. L., Qin, A. K., Suganthan, P. N., July 2006. Self-adaptative Differential Evolution Algorithm for Constrained Real-Parameter Optimization. In: 2006 IEEE Congress on Evolutionary Computation (CEC'2006). IEEE, Vancouver, BC, Canada, pp. 324–331.

Hunt, K., December 1983. Structural kinematics of in-parallel-actuated robot-arms. Journal of Mechanism, Transmissions and Automation in Design 105 (4), 705–712.

Kukkonen, S., Lampinen, J., July 2006. Constrained Real-Parameter Optimization with Generalized Differential Evolution. In: 2006 IEEE Congress on Evolutionary Computation (CEC'2006). IEEE, Vancouver, BC, Canada, pp. 911–918.

Laumanns, M., Thiele, L., Zitzler, E., Deb, K., July 2002. Archiving with Guaranteed Convergence and Diversity in Multi-Objective Optimization. In: Langdon, W., Cantú-Paz, E., Mathias, K., Roy, R., Davis, D., Poli, R., Balakrishnan, K., Honavar, V., Rudolph, G., Wegener, J., Bull, L., Potter, M., Schultz, A., Miller, J., Burke, E., Jonoska, N. (Eds.), Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'2002). Morgan Kaufmann Publishers, San Francisco, California, pp. 439–447.

Li, Q., Zhang, W., Chen, L., 2001. Design for control-A concurrent engineering approach for mechatronics systems design. IEEE/ASME Transactions on Mechatronics 6 (2), 161–168.

Liu, X.-J., Wang, J., Pritschow, G., February 2006. Kinematics, singularity and workspace of planar 5r symmetrical parallel mechanisms. Mechanism and Machine Theory 41 (2), 145–169.

Merlet, J. P., 2001. Parallel robots. Kluwer Academic Publishers.

Mezura-Montes, E., Muñoz-Dávila, L., Coello Coello, C. A., 2008a. A preliminary study of fitness inheritance in evolutionary constrained optimization. In: Krasnogor, N., Nicosia, G., Pavone, M., Pelta, D. (Eds.), Nature Inspired Cooperative Strategies for Optimization. Springer, Berlin, pp. 1–14, iSBN 978-3-540-78986-4.

Mezura-Montes, E., Portilla-Flores, E. A., Coello Coello, C. A., Alvarez-Gallegos, J., Cruz-Villar, C. A., 2008b. An Evolutionary Approach to Solve a Novel Mechatronic Multiobjective Optimization Problem. In: Siarry, P., Michalewicz, Z. (Eds.), Advances in Metaheuristic Methods for Hard Optimization. Springer, Berlin, pp. 329–351, iSBN 978-3-540-72959-4.

Mezura-Montes, E., Velázquez-Reyes, J., Coello Coello, C. A., July 2006. Comparing Differential Evolution Models for Global Optimization. In: 2006 Genetic and Evolutionary Computation Conference (GECCO'2006). Vol. 1. pp. 485–492.

Miettinen, K. M., 1999. Nonlinear Multiobjective Optimization. Kluwer Academic Publishers, Boston, Massachusetts.

Norton, R., 1996. Machine Design, An integrated approach. Prentice Hall Inc.

Oyama, A., Shimoyama, K., Fujii, K., May 2007. New constraint-handling method for multi-objective and multi-constraint evolutionary optimization. Transactions of the Japan Society for Aeronautical and Space Sciences 50 (167), 56–62.

Pil, A., Asada, H., March 1996. Integrated structure/control design of mechatronics systems using a recursive experimental optimization method. IEEE/ASME Transactions on Mechatronics 1 (1), 191–203.

Portilla-Flores, E. A., Mezura-Montes, E., Álvarez Gallegos, J., Coello-Coello, C. A., Cruz-Villar, C. A., August 2007. Integration of Structure and Control Using an Evolutionary Approach: An Application to the Optimal Concurrent Design of a CVT. International Journal for Numerical Methods in Engineering 71 (8), 883–901.

Price, K., Storn, R., Lampinen, J., 2005. Differential Evolution: A Practical Approach to Global Optimization. Natural Computing Series. Springer-Verlag.

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

Saravanan, R., Ramabalan, S., 2008. Evolutionary Minimum Cost Trajectory Planning for Industrial Robots. J. Intell. Robot Syst. 52, 45–77.

Van Brussel, H., Sas, P., Németh, I., Fonseca, P., Van den Braembussche, P., 2001. Towards a Mechatronic Compiler. IEEE/ASME Transactions on Mechatronics 6 (1), 90–104.

Villarreal-Cervantes, M., Cruz-Villar, C., Alvarez-Gallegos, J., Portilla-Flores, E., June 2010. Differential Evolution Techniques for the Structure-control Design of a Five-bar Parallel Robot. Engineering Optimization 42 (6), 535–565.

Zhang, W., Li, Q., Guo, L., 1999. Integrated design of a mechanical structure and control algorithm for a programmable four-bar linkage. IEEE/ASME Transactions on Mechatronics 4 (4), 354–362.

Zielinski, K., Laur, R., 2008. Stopping criteria for differential evolution in constrained single-objective optimization. In: Chakraborty, U. K. (Ed.), Advances in Differential Evolution. Springer, Berlin, pp. 111–138, iSBN 978-3-540-68827-3.