INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN ENGINEERING
Int. J. Numer. Meth. Engng 2000; **00**:1–6 *Prepared using* nmeauth.cls *[Version: 2002/09/18 v2.02]*

Integration of Structure and Control Using an Evolutionary Approach: An Application to the Optimal Concurrent Design of a CVT

Portilla-Flores Edgar A.*, †Mezura-Montes Efren, §Alvarez-Gallegos Jaime, ‡Coello-Coello Carlos A., §Cruz-Villar Carlos A.

Autonomous University of Tlaxcala, Engineering and Technology Department, †LANIA A.C., CINVESTAV-IPN, ‡Computer Science Section, §Mechatronic Section

SUMMARY

In this paper, a concurrent design methodology to formulate the mechatronic design problem is presented. The methology proposes to state the mechatronic design problem as a dynamic optimization problem. In order to explain the methodology, the concurrent design of pinion–rack continuously variable transmission (CVT) is carried out. Two optimization techniques, one based on a mathematical programming method and the other based on a novel evolutionary approach are used to solve it. The behavior of both approaches are compared, based on quality and the computational time required by each of them in a common platform.

*Correspondence to: Autonomous University of Tlaxcala, Engineering and Technology Department, Calz. Apizaquito s/n Km. 15, 90300 Apizaco Tlax. Mexico, email: eportilla@ingenieria.uatx.mx, phone: (+52) 241 4172544

Received XXX
Revised XXX

KEY WORDS: *Mechatronic system, Parametric optimal design, Multiobjective Optimization, Continuously variable transmission, Evolutionary Algorithms*

1. INTRODUCTION

Optimization arises by the need to design or to improve systems according to the requirement under which systems operate. There are several criteria that can help to quantify the system performance; however, these criteria are often in conflict with each other since frequently, the structural objectives of design require hard conditions for the control system. Therefore, the design problem is usually considered as a multiobjective design problem in order to obtain better systems. However, the traditional approach for the design of mechatronic systems, considers the mechanical behavior and the dynamic performance separately. Usually, the design of the mechanical elements involves kinematic and static behaviors while the design of the control system uses only the dynamic behavior; therefore, from a dynamic point of view, this approach cannot produce an optimal system behavior [1], [2]. Recent research in the area of mechatronic systems raises the need of a concurrent design methodology for mechatronic systems. This methodology must integrate mechanical structure and the design of the control in order to produce mechanical, electronical and control flexibility for the designed system [2], [3].

In this paper, a concurrent design methodology to formulate the mechatronic design problem is proposed. The methodology aims to state it within a dynamic optimization problem, because the goal is to fulfill the dynamic and control behaviors of the mechatronic system, besides to minimize some performance criteria [4]. This methodology allows us to obtain a set of optimal mechanical structure and controller parameters in only one step, which can produce a simple system reconfiguration.

In order to explain the methodology, the concurrent design of a pinion-rack continuously variable transmission (CVT) is carried out. This means that both the parametrical optimal design and the proportional and integral (PI) controller gains of the CVT are obtained by means of a multiobjective dynamic optimization problem (MDOP). In this case, both the kinematic and dynamic models of the mechanical structure and the dynamic model of the controller are jointly considered besides system performance criteria.

In the multiobjective optimization framework, a typical approach consists on transforming the original problem into an equivalent single objective problem using a weighted sum of the original objectives. In most of the cases, this single objective problem will be easier to solve than the original multiobjective problem. However, the weakness of the weighted method is that not all of the nondominated solutions can be found unless the problem is convex [5, 6]. A MDOP can be solved by converting it into a nonlinear programming (NLP) problem [7], [8] and using the Goal Attainment method [9] for the resulting problem. Two transcription methods exist for the MDOP: the sequential and the simultaneous methods [10]. In the sequential method, only the control variables are discretized; this method is also known as the control vector parameterization. In the simultaneous method the state and control variables are discretized resulting in a large-scale NLP problem which usually requires special solution strategies. However, these common methods need a point to initialize the optimization search, and consequently the convergence of the algorithm depends on the chosen point. Moreover, the nonlinear programming approach is able to produce only one possible solution.

Because of the above reasons, in this work we propose the use of an evolutionary-based approach. The main advantage in solving the problem with an evolutionary algorithm (EA) is that it always works with several initial starting points (called population) which are usually generated at random. In this way, we can avoid the sensitivity of the approach to the initial search point. Furthermore, we are able to discuss advantages and disadvantages of two different approaches used to solve the problem.

We wanted to emphasize the lack of sensitivity of the optimization approach to the initial conditions and also to use these initial solutions as a reference to find promising areas of the search space. Thus, we selected an EA which makes special emphasis in finding new search directions based on the distribution of the current solutions. These search directions are mostly random at the beginning and, once the process advances, they are based on the best solutions found so far. Differential Evolution (DE) was adopted as our search engine, because it fulfills the previously stated requirements.

DE is a population-based evolutionary algorithm with an special recombination operator that performs a linear combination of a number of individuals (normally three) and one parent (which is subject to be replaced) to create one child. The selection is deterministic between the parent and the child. The best of them remain in the next population. DE shares similarities with traditional EAs. However it does not use binary encoding as a simple genetic algorithm [11] and it does not use a probability density function to self-adapt its parameters as an Evolution Strategy [12]. Furthermore, DE has been successfully applied to mechanical design optimization tasks [13], [14]. However, in these previous works, the mechanical design problem is stated as a static optimization problem.

This paper is organized as follows: The concurrent design methodology is presented in Section 2. The application of a mathematical programming method to solve the problem is given in Section 3. The new evolutionary algorithm proposed in this work is explained in Section 4. The discussion of results is provided in Section 5 and finally, some conclusions are drawn in Section 6.

2. CONCURRENT OPTIMAL DESIGN

As it was discussed in Section 1, the mechatronic design problem must be stated on a concurrent way. Therefore we propose the following general problem:

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

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

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 (1) to (4): p is a vector of the design variables which belongs to the mechanical and control structure, x is the vector of the state variables and t is the time variable. On the other hand, some performance criteria L must be selected for the mechatronic system. The dynamic model (2) describes the state vector x at time t. Also, the design constraints of the mechatronic system must be developed and proposed, respectively. Therefore, the parameter vector p which is a solution of the previous problem will be an optimal set of structure and controller parameters which minimize the performance criteria selected for the mechatronic system and subject to the constraints imposed by the dynamic model and the design.

2.1. Dynamic model of the CVT

Current research efforts in the field of power transmission of rotational propulsion systems, are dedicated to obtain low energy consumption with high mechanical efficiency. An alternative solution to this problem is the so called continuously variable transmission (CVT), whose transmission ratio can be continuously changed inside an established range. There are many CVTs configurations built in industrial systems, especially in the automotive industry due to the requirements to increase the fuel economy without decreasing the system performance. The mechanical development of CVTs is well known and there is little to modify regarding its basic operation principles. However, research efforts continue with the controller design and the CVT instrumentation side. A pinion-rack CVT which is a traction-drive mechanism is presented in [15]. This CVT is built-in with conventional mechanical elements such as a gear pinion, one cam and two pairs of racks. The conventional CVT manufacture is an advantage over other existing CVTs.

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-in with a lead screw attached by a nut to the vertical transport cam. Fig. 1 depicts the main mechanical CVT components.

The dynamic model of a pinion-rack CVT is presented in [16]. Ordinary differential equations (5), (6) and (7) describe the CVT dynamic behavior. In equation (5): T_m is the input torque,

J_1 is the mass moment of inertia of the gear pinion, b_1 is the input shaft coefficient viscous damping, r is the gear pinion pitch circle radius, η is the mechanical CVT efficiency, T_L is the CVT load torque, J_2 is the mass moment of inertia of the rotor, R is the planetary gear pitch circle radius, b_2 is the output shaft coefficient viscous damping and θ is the angular displacement of the rotor. In equations (6) and (7): L, R_m, K_b, K_f and n represent the armature circuit inductance, the circuit resistance, the back electro-motive force constant, the motor torque constant and the gearbox gear ratio of the DC motor, respectively. Parameters r_p, λ_s, b_c and b_l denote the pitch radius, the lead angle, the viscous damping coefficient of the lead screw and the viscous damping coefficient of the offset mechanism, respectively. The control signal $u(t)$ is the input voltage to the DC motor. $J_{eq} = J_{c2} + Mr_p^2 + n^2 J_{c1}$ is the equivalent mass moment of inertia, where J_{c1} is the mass moment of inertia of the DC motor shaft, J_{c2} is the mass moment of inertia of the DC motor gearbox and $d = r_p \tan \lambda_s$, is a lead screw function. Moreover, $\theta_R(t) = \frac{1}{2}\arctan\left[\tan\left(2\dot{\theta}t - \frac{\pi}{2}\right)\right]$ is the rack angle meshing and $\frac{R}{r} = 1 + \frac{e}{r}\cos\theta_R$ is the CVT transmission ratio. The combined mass to be translated inside the rotor by the offset mechanism is denoted by M and $P = \frac{T_m}{r_p}\tan\phi\cos\theta_R$ is the loading on the gear pinion teeth, where ϕ is the pressure angle.

$$\left(\frac{R}{r}\right)\eta T_m - T_L = \left[J_2 + J_1\eta\left(\frac{R}{r}\right)^2\right]\ddot{\theta} - \left[J_1\eta\left(\frac{R}{r}\right)\frac{e}{r}\sin\theta_R\right]\dot{\theta}^2 + \left[\begin{array}{c} b_2 + b_1\eta\left(\frac{R}{r}\right)^2 \\ +J_1\eta\left(\frac{R}{r}\right)\frac{\dot{e}}{r}\cos\theta_R \end{array}\right]\dot{\theta} \quad (5)$$

$$L\frac{di}{dt} + R_m i = u(t) - \left[\frac{nK_b}{d}\right]\dot{e} \quad (6)$$

$$\left[\frac{nK_f}{d}\right] i - P = \left[M + \frac{J_{eq}}{d^2}\right]\ddot{e} + \left[b_l + \frac{b_c}{r_p d}\right]\dot{e} \tag{7}$$

2.2. Performance criteria and objective functions.

The performance of a system is measured by several criteria. One of the most used criteria is the system efficiency because it reflects the energy loss. In this work, the mechanical efficiency criterion of the gear systems is used in the concurrent design methodology. This is because the racks and the gear pinion are the principal CVT mechanical elements.

In [16] the mechanical CVT efficiency is given by (8) where μ, N_1 r and e represent the coefficient of sliding friction, the gear pinion teeth number, the pitch pinion radius and the offset, respectively.

$$\eta(t) = 1 - \frac{\pi\mu}{N_1}\left(1 + \frac{1}{1 + \frac{e\cos\theta_R}{r}}\right) \tag{8}$$

In order to maximize the mechanical CVT efficiency, $F(\cdot)$ given by (9) must be minimized.

$$F(\cdot) = \frac{1}{N_1}\left(1 + \frac{1}{1 + \frac{e\cos\theta_R}{r}}\right) \tag{9}$$

Equation (9) can be written as (10) which is used to state the design problem objective function.

$$L_1(\cdot) = \frac{1}{N_1}\left(\frac{2r + e\cos\theta_R}{r + e\cos\theta_R}\right) \tag{10}$$

On the other hand, in order to obtain the minimal controller energy, the design problem objective function given by (11) is used.

$$L_2(\cdot) = \frac{1}{2}\left[-K_p(x_{ref} - x_1) - K_I \int_0^t (x_{ref} - x_1)\right]^2 dt \tag{11}$$

2.3. Constraint functions.

The design constraints for the concurrent design of the CVT are proposed according to geometric and strength conditions for the gear pinion of the CVT.

To prevent the fracture of the annular portion between the axe bore and the teeth root on the gear pinion, the pitch circle diameter of the pinion gear must be greater than the bore diameter by at least 2.5 times the module [17]. Then, in order to avoid the fracture, the constraint g_1 must be imposed. To achieve a load uniform distribution on the teeth, the face width must be 6 to 12 times the value of the module [1]; this is ensured with constraints g_2 and g_3. To maintain the CVT transmission ratio in the range $[2r, 5r]$, constraints g_4, g_5 are imposed. Constraint g_6 ensures a teeth number of the gear pinion equal or greater than 12 [1]. A practical constraint requires that the gear pinion face width must be equal or greater than $20mm$, in order to ensure that constraint g_7 is imposed. To constrain the distance between the corner edge in the rotor and the edge rotor, constraint g_8 is imposed. Finally, to ensure a practical design for the pinion gear, the pitch circle radius must be equal or greater than $25.4mm$, which is what g_9 imposes.

On the other hand, it can be observed that J_1, J_2 are parameters which are a function of the CVT geometry. For these mechanical elements, the mass moments of inertia are defined by (12), where ρ, m, N, h, e_{max}, r_c and r_s are the material density, the module, the teeth number of the gear pinion, the face width, the highest offset distance between axes, the rotor radius and the bearing radius, respectively.

$$J_1 = \frac{1}{32}\rho\pi m^4 \left(N+2\right)^2 N^2 h; \quad J_2 = \rho h \left[\frac{3}{4}\pi r_c^4 - \frac{16}{6}\left(e_{max} + mN\right)^4 - \frac{1}{4}\pi r_s^4\right] \qquad (12)$$

2.4. Design variables.

One of the most important facts in the methodology proposed in this work is the vector of the design variables selected, since these must belong to the mechanical and the controller structure. Therefore, for the concurrent design of the CVT, design variables of the mechanical structure related with the standard nomenclature for a gear tooth are used. That is because, the rotor size of the CVT will not improve only the internal mechanical elements.

Equation (13) states a parameter called module m for metric gears, where d_p is the pitch diameter and N is the teeth number.

$$m = \frac{d_p}{N} = \frac{2r}{N} \tag{13}$$

The face width h, which is the distance measured along the axis of the gear and the highest offset distance between axes e_{max} are parameters which define the CVT size. The above design variables belong to the mechanical structure.

On the other hand, a proportional and integral (PI) controller structure is used, this is because despite the development of many control strategies, this controller structure remains as one of the most popular approach for industrial processes control due to the adequate performance in most of such applications. Therefore, controller gains K_P and K_I belong to the dynamic CVT behavior. Finally, the vector p^i which considers mechanical and dynamic design variables is proposed in order to carry out the concurrent design of the CVT.

$$p^i = [p^i_1, p^i_2, p^i_3, p^i_4, p^i_5, p^i_6]^T = [N, m, h, e_{max}, K_P, K_I]^T \tag{14}$$

2.5. Optimization problem.

In order to obtain the mechanical CVT parameter optimal values, we propose a multiobjective dynamic optimization problem given by equations (15) to (23). 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 (17). As the objective functions must be normalized to the same scale [5], the corresponding factors $W = [0.4397, 563.3585]^T$ were obtained using the algorithm from Section 3 by minimizing each objective function subject to constraints given by equations (16) to (23).

$$\min_{p \in R^6} \Phi(x, p, t) = [\Phi_1, \Phi_2]^T \tag{15}$$

where

$$\Phi_1 = \frac{1}{W_1} \int_0^{10} \left[\frac{1}{p_1} \left(\frac{p_1 p_2 + x_3 \cos\theta_R}{\frac{p_1 p_2}{2} + x_3 \cos\theta_R} \right) \right] dt$$

$$\Phi_2 = \frac{1}{W_2} \int_0^{10} u^2 dt$$

subject to

$$\begin{aligned}
\dot{x}_1 &= \frac{AT_m + \left[J_1 A \frac{2x_3}{p_1 p_2} \sin\theta_R \right] x_1^2 - T_L - \left[b_2 + b_1 A^2 + J_1 A \frac{2x_4}{p_1 p_2} \cos\theta_R \right] x_1}{J_2 + J_1 A^2} \\
\dot{x}_2 &= \frac{u(t) - (\frac{nK_b}{d}) x_4 - R x_2}{L} \\
\dot{x}_3 &= x_4 \\
\dot{x}_4 &= \frac{(\frac{nK_f}{d}) x_2 - (b_l + \frac{b_c}{r_p d}) x_4 - \frac{T_m}{r_p} \tan\phi \cos\theta_R}{M + \frac{J_{eq}}{d^2}}
\end{aligned} \tag{16}$$

$$u(t) = -p_5(x_{ref} - x_1) - p_6 \int_0^t (x_{ref} - x_1)dt \tag{17}$$

$$J_1 = \frac{1}{32}\rho\pi p_2^4 \left(p_1 + 2\right)^2 p_1^2 p_3 \tag{18}$$

$$J_2 = \frac{\rho p_3}{4}\left[3\pi r_c^4 - \frac{32}{3}\left(p_4 + p_1 p_2\right)^4 - \pi r_s^4\right] \tag{19}$$

$$A = 1 + \frac{2x_3}{p_1 p_2}\cos\theta_R \tag{20}$$

$$d = r_p \tan\lambda_s \tag{21}$$

$$\theta_R = \frac{1}{2}\arctan\left[\tan\left(2x_1 t - \frac{\pi}{2}\right)\right] \tag{22}$$

$$\begin{aligned}
g_1 &= 0.01 - p_2\left(p_1 - 2.5\right) \leq 0 \\
g_2 &= 6 - \frac{p_3}{p_2} \leq 0 \\
g_3 &= \frac{p_3}{p_2} - 12 \leq 0 \\
g_4 &= p_1 p_2 - p_4 \leq 0 \\
g_5 &= p_4 - \frac{5}{2}p_1 p_2 \leq 0 \\
g_6 &= 12 - p_1 \leq 0 \\
g_7 &= 0.020 - p_3 \leq 0 \\
g_8 &= 0.020 - \left[r_c - \sqrt{2}(p_4 + p_1 p_2)\right] \leq 0 \\
g_9 &= 0.0254 - p_1 p_2 \leq 0
\end{aligned} \tag{23}$$

3. MATHEMATICAL PROGRAMMING METHOD

The resulting problem stated by (15)-(23) is solved using the goal attainment method. The resulting problem is stated in equations (24) and (25) subject to equations (2) to (4), where $\omega = [w_1, w_2]^T$ is the scattering vector [5], $\Phi^d = [1, 1]^T$ are the desired goals for each objective function and $\Phi_1(p)$ and $\Phi_2(p)$ are the evaluated function.

$$\min_{p,\lambda} G\left(p, \lambda\right) \triangleq \lambda \tag{24}$$

subject to:

$$\begin{aligned} g(p) &\leq 0 \\ g_{a1}(p) &= \Phi_1\left(p\right) - \omega_1 \lambda - \Phi_1^d \leq 0 \\ g_{a2}(p) &= \Phi_2\left(p\right) - \omega_2 \lambda - \Phi_2^d \leq 0 \end{aligned} \tag{25}$$

A vector p^i which contains the current parameter values is proposed and the NLP problem given by equations (26) and (27) is obtained, where B_i is the Broyden–Fletcher–Goldfarb–Shanno updated (BGFS) positive definite approximation of the Hessian matrix, and the gradient calculation is obtained using sensitivity equations. Hence, if γ solves the subproblem given by (26) and (27) and $\gamma = 0$, then the parameter vector p^i is an original problem optimal solution. Otherwise, we set $p^{i+1} = p^i + \gamma$ and with this new vector the process is repeated again.

$$\min_{\gamma \in R^{j+1}} QP(p^i) = G\left(p^i\right) + \nabla G^T\left(p^i\right)\gamma + \frac{1}{2}\gamma^T B_i \gamma \tag{26}$$

subject to

$$
\begin{aligned}
g(p^i) + \nabla g^T \left(p^i\right) \gamma &\leq 0 \\
g_{a1}(p^i) + \nabla g_{a1}^T \left(p^i\right) \gamma &\leq 0 \\
g_{a2}(p^i) + \nabla g_{a2}^T \left(p^i\right) \gamma &\leq 0
\end{aligned} \tag{27}
$$

The NLP approach requires the gradient calculation, therefore the sensitivity equations must be solved in order to obtain the necessary information to establish a subproblem. The number of sensitivity equations is the product between the number of state variables and the number of the design variables of the design vector. In this case, the number of sensitivity equations are twenty four. Additionally, it is necessary to solve six gradient equations and two other equations in order to obtain the values of the objective functions. Additionally, fifty four gradient equations of the constraints must be calculated.

The gradient calculation (28) is obtained solving ordinary differential equations of the sensitivity equations stated by (29).

$$
\frac{\partial \Phi_i}{\partial p_j} = \int_{t_0}^{t_f} \left(\frac{\partial L_i}{\partial x} \left[\frac{\partial x}{\partial p_j}(t) \right] + \frac{\partial L_i}{\partial p_j} \right) dt \tag{28}
$$

$$
\frac{d}{dt} \left[\frac{\partial x}{\partial p_j} \right] = \frac{\partial f}{\partial x} \left[\frac{\partial x}{\partial p_j} \right] + \frac{\partial f}{\partial p_j} \tag{29}
$$

where, as it can be seen in the general problem stated by (1) to (4), L_i is the $i-th$ objective function, f is the dynamic model of the mechatronic system, x is the vector of the state variables, p_j is the $j-th$ element of the vector of the design variables and t is the time variable.

3.1. NLP results

The system parameters used in the optimization procedure were: $b_1 = 1.1Nms/rad$, $b_2 = 0.05Nms/rad$, $r = 0.0254m$, $T_m = 8.789Nm$, $T_L = 0Nm$, $\lambda_s = 5.4271$, $\phi = 20$, $M = 10Kg$, $r_p = 4.188E - 03m$, $K_f = 63.92E - 03Nm/A$, $K_b = 63.92E - 03Vs/rad$, $R = 10\Omega$, $L = 0.01061H$, $b_l = 0.015Ns/m$, $b_c = 0.025Nms/rad$ and $n = ((22 * 40 * 33)/(9 * 8 * 9))$. The initial conditions vector was $[x_1(0), x_2(0), x_3(0), x_4(0)]^T = [7.5, 0, 0, 0]^T$ and the output reference was considered to be $x_{ref} = 3.2$.

The goal attainment method requires the goal for each one of the objective functions. The goal for Φ_1 was obtained by minimizing this function subject to equations (16)-(23). The optimal solution vector p^1 is shown in Table I. The goal for Φ_2 was obtained by minimizing this function subject to equations (16)-(23). The optimal solution vector p^2 for this problem is also shown in Table I.

Varying the scattering vector can produce different nondominated solutions. In Table I, two cases are presented: p^*_A is obtained with $\omega = [0.5, 0.5]^T$, p^*_B is obtained with $\omega = [0.4, 0.6]^T$.

We performed 10 independent runs, all of them by using a PC with a 2.8 GHz Pentium IV processor with 1 GB of Memory.

As it can be seen in the results in Table II, 80% of the runs diverged. This behavior shows a high sensitivity of the goal attainment method to the starting point because it must be carefully chosen in order to allow the approach to reach a good solution. The information about the time required by the goal attainment approach per independent run is also summarized in Table II.

In the mechanical CVT efficiency equation, it can be observed that a higher efficiency is produced when a bigger gear pinion teeth number is used. From the nondominated solution it can be observed that, when the teeth number is increased (p_1^*) and their sizes are decreased (p_2^*), a higher CVT mechanical efficiency is obtained. On the other hand, a more compact CVT size is obtained since (p_3^*) is decreased. Furthermore, a minimal controller energy is obtained when the controller gains (p_5^*) and (p_6^*) are decreased. Despite the sensitivity of the NLP method, the optimal solutions obtained are good from the mechanical and controller point of view.

4. Evolutionary Optimization

Based on the considerable sensitivity of the goal attainment approach to its initial search point, which we argue that may lead the search to a local optimum solution, we decided to experiment with an evolutionary approach to solve the optimization problem of our interest. The main advantage that we observed in solving the problem with an evolutionary algorithm (EA) is that it always works with several initial starting points (called population) usually generated at random. In this way, we expected to avoid the sensitivity of the approach to the initial point provided. Furthermore, by using and EA in addition to the Goal Attainment method, we are able to discuss advantages and disadvantages of using each of them in our problem.

We use the standard version of the differential evolution algorithm called DE/rand/1/bin [18] and its algorithm [18] is presented in Figure 2.

The "CR" parameter controls the influence of the parent in the generation of the offspring. Higher values mean less influence of the parent. The "F" parameter scales the influence of two of the three individuals selected at random to generate the offspring.

We propose a novel DE-based approach to solve our multiobjective optimization problem. Therefore, we added to the original DE, specific mechanisms required to solve the current problem:

1. The selection criterion was modified in order to handle a multiobjective problem.
2. A mechanism was added to handle the constraints of the problem.

4.1. Multiobjective optimization

The criterion to select the fittest solution between parent and offspring in traditional DE is based on the value of the objective function. However, in multiobjective optimization we are looking for trade-off solutions. Therefore, we propose to use, as in other evolutionary algorithms for multiobjective optimization [19], Pareto dominance as a selection criterion between parent and offspring. In this way, nondominated solutions will remain in the current population. We understand the nondominance criterion as follows:

A vector $\vec{u} = (u_1, \ldots, u_k)$ is said to dominate $\vec{v} = (v_1, \ldots, v_k)$ (denoted by $\vec{u} \preceq \vec{v}$) if and only if $\vec{u}$ is partially less than $\vec{v}$, i.e. $\forall i \in \{1, \ldots, k\},\ u_i \leq v_i \wedge \exists i \in \{1, \ldots, k\} : u_i < v_i$. If we denote the feasible region of the search space as $\mathcal{F}$, the evolutionary multiobjective algorithm will look for the Pareto optimal set ($\mathcal{P}^*$) defined as:

$$\mathcal{P}^* := \{x \in \mathcal{F} \mid \neg\exists\ x' \in \mathcal{F}\ \ \vec{f}(x') \preceq \vec{f}(x)\}. \tag{30}$$

In our case, $k = 2$ as we are optimizing two objectives.

Additionally, we also added an external memory [20] (a data structure in the implementation of the approach), in order to store all the nondominated solutions found during the evolutionary process. The external memory is implemented as follows: At each generation, all the nondominated solutions in the current population are inserted in the external memory. Then, nondominance checking is performed per each solution and those individuals nondominated in this external memory will remain and the dominated solutions will be eliminated. At the end of the process, our approach reports the nondominated solutions that remain in the external storage.

4.2. Constraint handling

The mechanism to deal with constraints are three simple selection criteria which guide the algorithm to the feasible region of the search space:

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

These criteria are applied when the child is compared against the parent subject to be replaced. The idea of using selection criteria based on feasibility to deal with constraints in genetic algorithms was originally proposed by Deb [21] and was extended to other evolutionary algorithms in other approaches [22, 23, 24], which have been mainly used to solve global optimization problems.

The algorithm of our modified DE (MDE) is presented in Figure 3:

4.3. MDE Results

We performed 10 independent runs with the same set of parameters for the MDE approach: Population number $NP = 200$, $MAX\_GENERATIONS = 100$; parameters F and CR were randomly generated. The parameter F was generated per generation in the range $[0.3, 0.9]$ and CR was generated per run in the range $[0.8, 1.0]$. These values were empirically derived. We observed that adopting higher values for the "CR" parameter (i.e. the offspring will inherit more features from the random solutions selected from the population than those inherited from its parent) provided a better performance for the MDE. Furthermore, we observed that, allowing the "F" parameter to vary within all its allowable interval except for the extreme values, led to better results. As the experiments performed with the Goal Attainment method, all tests took place in the same platform where the mathematical programming method was tested.

The number of nondominated solutions at each run and the time required to obtain them by the MDE are presented in Table III.

It is worth reminding that, unlike the mathematical programming method, MDE always starts with a set of solutions generated at random using a uniform distribution. To promote a fair comparison, the system parameters used in the MDE are exactly the same used in the previous experiments performed with the goal attainment approach.

The 10 different Pareto fronts generated by each run are presented in Figure 4. They show that the performance of the MDE in each single run is very consistent. All runs could reach the same regions of the Pareto Front despite starting from random points.

Based on the 10 outputs of each run (i.e. our set of trade-off solutions), we merged all of them and we obtained the global set of nondominated solutions. The Pareto front obtained is presented in Figure 5.

As we can observe in the set of nondominated solutions obtained, solutions in the middle of the Pareto front present a good performance both from a mechanical and from a controller point of view. With these nondominated solutions we have a higher mechanical efficiency since the teeth number (p_1^*) was increased beyond 25 and its size (p_2^*) was decreased. It can be noted that in all nondominated solutions, a more compact CVT size was obtained since (p_3^*) has a value closer to 0.02. On the other hand, with the gain controllers obtained, a minimal controller energy can be implemented.

As a final experiment, to test the capabilities of the NLP method with a nondominated solution as a starting point, we decided to use one of the solutions found by the MDE as an initial point for the goal attainment method to evaluate its performance when trying to improve this solution. However, the NLP method was unable to generate a better solution in all cases. We tried with five different solutions taken from both extremes of the Pareto front plotted in Figure 5 and also five solutions located in the middle of it.

5. Discussion of results

5.1. Implementation Difficulty

For the NLP approach, the whole system equation was simultaneously solved to establish the subproblem until the stop criteria of the subproblem were satisfied. The initial point to search the optimum is p^2 for both scattering vectors. This initial point was selected because the optimal solutions p_A^* and p_B^* were only reached when the search started from this point. The number of evaluations of the whole system equation were 6 and 12 for each scattering vector, respectively.

On the other hand, the MDE approach only uses the evaluated functions, which are obtained by solving the four differential equations of the dynamic system and the two objective function equations.

Based on these issues, we can say that the MDE approach is easier to implement than the NLP method.

5.2. Performance

Based on the performance obtained by both optimization techniques used to solve the problem, we highlight the following issues:

- The goal attainment method was very sensitive to the initial point. In fact, to get an adequate initial point for this method, a global optimization technique was required in order to optimize each objective separately, and from both solutions obtained (one per objective), one of them was selected as a starting point for the method. On the other hand, as indicated before, the MDE always started the search from purely random solutions.
- Some points of the final Pareto front obtained by the MDE approach were used like an initial point for the NLP approach but this method was unable to improve them in all cases. It is worth noting that, with specific modifications to the NLP method, it will be possible to find more nondominated solutions. However, our experimental design was set to compare two different approaches in similar conditions (with no more modifications) in order to evaluate their strenghts and shortcomings.
- Despite using random solutions at the beginning, the MDE was able to converge in all

the 10 independent runs performed to a feasible Pareto front.

- The average number of points obtained by the MDE per run was: 18.5.
- The solutions obtained with the goal attainment method are also part of the Pareto front found by the MDE (they are nondominated with respect to each other). Therefore, the quality of solutions obtained by both approaches can be considered similar. However, based on the mechanical efficiency objective values, the solutions found by the MDE are preferred (as discussed in Section 4.3).
- The average time required by the goal attainment method is clearly lower than the average time used by the MDE to get competitive results. However, as the MDE obtains an average of 18.5 solutions in an average of 18.7 Hrs, it approximately obtained one solution per hour. On the other hand, the goal attainment method obtains one solution in about half an hour. Then, we can conclude that the MDE requires twice the time used by the goal attainment method to obtain a solution.

All these issues lead us to conclude that the MDE approach is clearly a better option than the Goal Attainment method (based on the simplicity to implement it, its lack of sensitivity to the initial conditions and its overall performance), assumming that enough time is available to find an optimal set of solutions. Otherwise, the NLP method is useful when a solution is required in a short period of time, assuming a more complicated implementation and a careful definition of the initial conditions.

6. Conclusions

In this paper, we have presented a suitable parametric optimal design methodology for a mechatronic system. The advantage of this methodology is that the parametric optimal design can be stated as a MDOP where kinematic and dynamic behaviors are considered besides two performance criteria. The design methodology fulfills the concurrent design concept, since the control–structure optimization are integrated and solved in only one stage.

The problem was initially solved by using a mathematical programming method for multiobjective optimization. However, the results were not as good as expected. Therefore, an evolutionary approach called Differential Evolution was used to address the problem. Some simple modifications were made to the original Differential Evolution Algorithm in order to solve a multiobjective optimization problem. The performance of the evolutionary approach was, by far much better than that provided by the goal attainment approach and it was not sensitive to the initial set of solutions. However, the computational cost associated with the evolutionary approach was much higher than the mathematical programming approach.

7. Future work

As future paths of research, we plan to propose new design constraints. These constraints must consider stress conditions and bounding of the state variables. On the other hand, another objective function of the pinion-rack CVT overall mechanical efficiency, including the offset mechanism and lead screw constraints could be considered in the parametric optimal design. Furthermore, new design variables could be considered. These design variables would be related to the lead screw of the offset mechanism. Finally, we plan to design an hybrid approach which would combine the advantages of both methods compared here in order to obtain even a better performance.

Acknowledgements

The fourth author acknowledges support from CONACyT project no. 45683-Y.

REFERENCES

1. Norton, R., *Machine Design. An integrated approach*, Prentice-Hall Inc., 1996.
2. van Brussel, H., I. Sas P, N., DFonseca, P., and van den Braembussche, P., "Towards a Mechatronic Compiler," *IEEE/ASME Transactions on Mechatronics*, Vol. 6, No. 1, 2001, pp. 90–104.
3. Zhang, W., Li, Q., and Guo, L., "Integrated design of a mechanical structure and control algorithm for a programmable four-bar linkage," *Transactions on Mechatronics*, Vol. 4, No. 4, 1999, pp. 354–362.
4. Bryson, A., *Dynamic Optimization*, Addison-Wesley, 1999.
5. Osyczka, A., *Multicriterion optimization in engineering*, John Wiley and Sons, 1984.
6. Das, I. and Dennis, J., "A Closer Look at Drawbacks of Minimizing Weighted Sums of Objectives for Pareto Set Generation in Multicriteria Optimization Problems," *Structural Optimization*, Vol. 14, No. 1, 1997, pp. 63–69.
7. Kraft, D., "On converting optimal control problems into nonlinear programming problems," *Computational Mathematical Programming*, Springer-Verlag, 1985, pp. 261–280.
8. Goh, C. and Teo, K., "Control parametrization: a unified approach to optimal control problems with general constraints," *Automatica*, Vol. 24, No. 1, 1988, pp. 3–18.
9. Liu, G., Yang, J., and Whidborne, J., *Multiobjective optimisation and control*, Research Studies Press, 2003.
10. Betts, J., *Practical methods for optimal control using nonlinear programming*, SIAM, Philadelphia, USA, 2001.
11. Goldberg, D. E., *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley, Mass. USA, 1989.
12. Schwefel, H.-P., editor, *Evolution and Optimization Seeking*, John Wiley & Sons, New York, 1995.

13. Deb, K. and Kain, S., "Multi-Speed Gearbox Design Using Multi-Objective Evolutionary Algorithms," *Journal of Mechanical Design*, Vol. 125, No. 3, September 2003, pp. 609–619.

14. Shiakolas, P., Koladiya, D., and Kebrle, J., "On the optimum synthesis of six–bar linkages using differential evolution and the geometric centroid of precision positions technique," *Mechanism and Machine Theory*, Vol. 40, No. 3, March 2005, pp. 319–335.

15. Silva, C. D., Schultz, M., and Dolejsi, E., "Kinematic analysis and design of a continuously variable transmission," *Mechanism and Machine Theory*, Vol. 29, No. 1, 1994, pp. 149–167.

16. Alvarez-Gallegos, J., Cruz-Villar, C., and Portilla-Flores, E., "Parametric optimal design of a pinion–rack continuously variable transmission," *Proceedings of the 2005 IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, Monterey California, 2005, pp. 899–904.

17. Papalambros, P. and Wilde, D., *Principles of optimal design. Modelling and computation*, Cambridge University Press, 2000.

18. Price, K. V., "An Introduction to Differential Evolution," *New Ideas in Optimization*, edited by D. Corne, M. Dorigo, and F. Glover, Mc Graw-Hill, UK, 1999, pp. 79–108.

19. Coello Coello, C. A., Van Veldhuizen, D. A., and Lamont, G. B., *Evolutionary Algorithms for Solving Multi-Objective Problems*, Kluwer Academic Publishers, New York, June 2002, ISBN 0-3064-6762-3.

20. Knowles, J. D. and Corne, D. W., "Approximating the Nondominated Front Using the Pareto Archived Evolution Strategy," *Evolutionary Computation*, Vol. 8, No. 2, 2000, pp. 149–172.

21. Deb, K., "An Efficient Constraint Handling Method for Genetic Algorithms," *Computer Methods in Applied Mechanics and Engineering*, Vol. 186, No. 2/4, 2000, pp. 311–338.

22. Jiménez, F. and Verdegay, J. L., "Evolutionary techniques for constrained optimization problems," *7th European Congress on Intelligent Techniques and Soft Computing (EUFIT'99)*, edited by H.-J. Zimmermann, Verlag Mainz, Aachen, Germany, 1999, pp. 182–183, ISBN 3-89653-808-X.

23. Lampinen, J., "A Constraint Handling Approach for the Diifferential Evolution Algorithm," *Proceedings of the Congress on Evolutionary Computation 2002 (CEC'2002)*, Vol. 2, IEEE Service Center, Piscataway, New Jersey, May 2002, pp. 1468–1473.

24. Mezura-Montes, E. and Coello Coello, C. A., "A Simple Multimembered Evolution Strategy to Solve Constrained Optimization Problems," *IEEE Transactions on Evolutionary Computation*, Vol. 9, No. 1, February 2005, pp. 1–17.

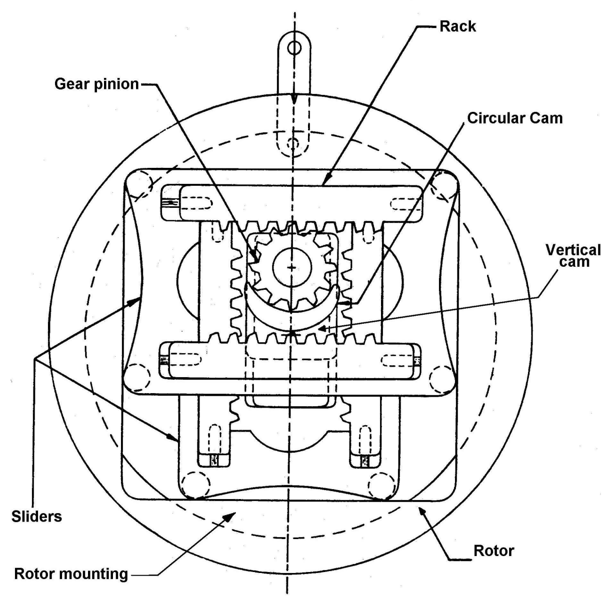

Figure 1. Main CVT mechanical components

$[N^*, m^*, h^*, e^*_{max}, K^*_P, K^*_I]$	$\Phi_N(\bullet) = [\Phi_1(\bullet), \Phi_2(\bullet)]$	$\Phi(\bullet) = [\Phi_1(\bullet), \Phi_2(\bullet)]$
$p^1 = [38, 0.0017, 0.02, 0.0636, 10.000, 1.00]$	$\Phi_N(p^1) = [1.0000, 4.7938]$	$\Phi(p^1) = [0.4397, 2700.6279]$
$p^2 = [13.4459, 0.0019, 0.02, 0.0826, 5.000, 0.01]$	$\Phi_N(p^2) = [2.8017, 1.0000]$	$\Phi(p^2) = [1.2319, 563.3585]$
$p^*_A = [26.7805, 0.0017, 0.02, 0.0826, 5.000, 0.01]$	$\Phi_N(p^*_A) = [1.4696, 1.4696]$	$\Phi(p^*_A) = [0.6461, 827.9116]$
$p^*_B = [29.0171, 0.0017, 0.02, 0.0789, 5.000, 0.01]$	$\Phi_N(p^*_B) = [1.3646, 1.5469]$	$\Phi(p^*_B) = [0.6000, 871.4592]$

Table I. MDOP solutions

Run	Time required	Initial search point	Scattering vector
1	23.78 Min.	[13.4459, 0.0019, 0.02, 0.0826, 5.000, 0.01]	[0.5,0.5]
2	Diverged.	[38, 0.0017, 0.02, 0.0636, 10.000, 1.00]	[0.5,0.5]
3	Diverged.	[38, 0.0017, 0.02, 0.0636, 10.000, 1.00]	[0.4,0.6]
4	Diverged.	[38, 0.0017, 0.02, 0.0636, 10.000, 1.00]	[0.6,0.4]
5	Diverged.	[28.8432, 0.0017, 0.02, 0.0550, 5.024, 0.017]	[0.5,0.5]
6	48.5 Min.	[13.4459, 0.0019, 0.02, 0.0826, 5.000, 0.01]	[0.4,0.6]
7	Diverged.	[28.8432, 0.0017, 0.02, 0.0550, 5.024, 0.017]	[0.4,0.6]
8	Diverged.	[28.8432, 0.0017, 0.02, 0.0550, 5.024, 0.017]	[0.6,0.4]
9	Diverged.	[30.77, 0.0017, 0.02, 0.0694, 5.121, 0.010]	[0.5,0.5]
10	Diverged.	[30.77, 0.0017, 0.02, 0.0694, 5.121, 0.010]	[0.4,0.6]
Average	**36.365 Min**		

Table II. Time required by each run of the Goal attainment approach. Note that only two runs could converge to a solution. The remaining 8 runs could not provide any result.

Run	Nondominated solutions found	Time required
1	17	18.53 Hrs.
2	15	20.54 Hrs.
3	25	18.52 Hrs.
4	16	18.63 Hrs.
5	17	18.55 Hrs.
6	19	17.57 Hrs.
7	18	18.15 Hrs.
8	24	18.47 Hrs.
9	16	18.67 Hrs.
10	18	20.24 Hrs.
Average	**18.5 solutions**	**18.78 Hrs**

Table III. Number of nondominated solutions found at each independent run by the MDE and the time required by each one of them.

```
Begin
    G=0
    Create a random initial population \vec{x}_G^i ∀i, i = 1, ..., NP
    Evaluate f(\vec{x}_G^i) ∀i, i = 1, ..., NP
    For G=1 to MAX_GENERATIONS Do
        For i=1 to NP Do
            Select randomly r_1 ≠ r_2 ≠ r_3 :
            j_rand = randint(1, D)
            For j=1 to D Do
                If (rand_j[0, 1) < CR or j = j_rand) Then
                    u^i_{j,G+1} = x^{r_3}_{j,G} + F(x^{r_1}_{j,G} − x^{r_2}_{j,G})
                Else
                    u^i_{j,G+1} = x^i_{j,G}
                End If
            End For
            If (f(\vec{u}^i_{G+1}) ≤ f(\vec{x}^i_G)) Then
                \vec{x}^i_{G+1} = \vec{u}^i_{G+1}
            Else
                \vec{x}^i_{G+1} = \vec{x}^i_G
            End If
        End For
        G = G + 1
    End For
End
```

Figure 2. DE 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. “NP”, “MAX_GENERATIONS”, “CR” and “F” are user-defined parameters.

```
     Begin
         G=0
         Initialize the external memory EM_0 = ∅
         Create a random initial population x⃗^i_G ∀i, i = 1,...,NP
         Evaluate each x⃗^i_G in each objective function and in each constraint ∀i, i = 1,...,NP
         For G=1 to MAX_GENERATIONS Do
             For i=1 to NP Do
                 Select randomly r1 ≠ r2 ≠ r3 :
                 j_rand = randint(1, D)
                 For j=1 to D Do
                     If (rand_j[0,1) < CR or j = j_rand) Then
                         u^i_{j,G+1} = x^{r3}_{j,G} + F(x^{r1}_{j,G} − x^{r2}_{j,G})
                     Else
                         u^i_{j,G+1} = x^i_{j,G}
                     End If
                 End For
                 Evaluate u⃗^i_{G+1} in each objective function and in each constraint
⟹               If (u⃗^i_{G+1} is better than x⃗^i_G (based on the three selection criteria)) Then
                     x⃗^i_{G+1} = u⃗^i_{G+1}
                 Else
                     x⃗^i_{G+1} = x⃗^i_G
                 End If
             End For
             G = G + 1
⟹           Find the nondominated solutions ND_G from the current population: x⃗^i_G ∀i, i = 1,...,NP
⟹           Update EM_G with ND_G by performing nondominance checking
         End For
     End
```

Figure 3. MDE algorithm. The modified steps are marked with an arrow. 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. “NP”, “MAX_GENERATIONS”, “CR” and “F” are user-defined parameters

$[N^*, m^*, h^*, e^*_{max}, K^*_P, K^*_I]$	$[\boldsymbol{\Phi_1}(\bullet), \boldsymbol{\Phi_2}(\bullet)]$
[32.949617, 0.001780, 0.020413, 0.063497, 5.131464, 0.022851]	[0.534496, 1033.243548]
[25.022005, 0.001699, 0.020103, 0.052385, 5.087026, 0.024991]	[0.687214, 837.167059]
[24.764331, 0.001723, 0.020662, 0.048119, 5.104801, 0.011072]	[0.694969, 828.856396]
[32.203853, 0.001793, 0.021356, 0.066703, 5.033164, 0.012833]	[0.547385, 984.149814]
[30.774167, 0.001710, 0.020092, 0.069459, 5.129618, 0.010260]	[0.568131, 950.480089]
[34.231339, 0.001756, 0.020974, 0.065426, 5.104461, 0.023469]	[0.515604, 1042.009590]
[31.072336, 0.001760, 0.020295, 0.072332, 5.018621, 0.024963]	[0.564775, 964.310541]
[27.647589, 0.001685, 0.020151, 0.069264, 5.001687, 0.031805]	[0.627021, 877.670407]
[27.548056, 0.001696, 0.020083, 0.067970, 5.006868, 0.017859]	[0.629913, 864.206663]
[30.866972, 0.001735, 0.020305, 0.058766, 5.002777, 0.032694]	[0.567519, 960.120458]
[28.913492, 0.001747, 0.020478, 0.058322, 5.021887, 0.027174]	[0.603222, 923.771423]
[28.843277, 0.001764, 0.020282, 0.055027, 5.024443, 0.017157]	[0.605340, 915.753294]
[30.185435, 0.001700, 0.020075, 0.059569, 5.133269, 0.019914]	[0.577733, 949.842309]
[29.448640, 0.001755, 0.020601, 0.063276, 5.019318, 0.033931]	[0.593085, 944.906551]
[20.002905, 0.001697, 0.020098, 0.053235, 5.114809, 0.018447]	[0.844657, 715.605541]
[26.373053, 0.001718, 0.020176, 0.068410, 5.031773, 0.014986]	[0.656264, 849.215816]
[32.227085, 0.001764, 0.020567, 0.070369, 5.178989, 0.026127]	[0.544721, 1030.722785]
[23.476167, 0.001731, 0.020618, 0.057264, 5.050345, 0.010533]	[0.730990, 790.412654]
[23.853314, 0.001696, 0.020054, 0.063646, 5.097374, 0.040464]	[0.717403, 827.978369]
[23.936736, 0.001767, 0.020179, 0.054081, 5.026456, 0.013965]	[0.719347, 810.685134]
[18.094865, 0.001754, 0.020097, 0.033930, 5.263513, 0.012051]	[0.926890, 700.251032]
[15.287561, 0.001836, 0.020539, 0.065247, 5.001634, 0.077960]	[1.086582, 648.563140]
[20.410186, 0.001689, 0.020082, 0.067889, 5.0055020.046545]	[0.828891, 729.481066]
[29.319668, 0.001754, 0.020557, 0.057790, 5.140154, 0.012875]	[0.595073, 944.511281]
[28.165197, 0.001722, 0.020449, 0.069922, 5.035457, 0.013965]	[0.617721, 886.468167]
[34.733111, 0.001738, 0.020849, 0.064827, 5.470063, 0.078838]	[0.504179, 1230.655492]
[18.028162, 0.001753, 0.021026, 0.075356, 5.185506, 0.027797]	[0.930299, 697.362827]
[21.642511, 0.001694, 0.020196, 0.061009, 5.040619, 0.029378]	[0.785859, 752.464167]

Table IV. Details of the trade-off solutions found by the MDE. All solutions are feasible.

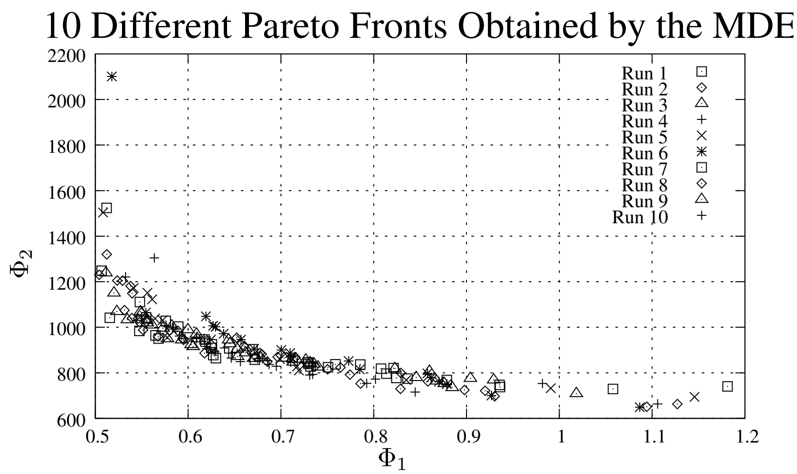

Figure 4. 10 Pareto Fronts obtained in the 10 independent runs performed.

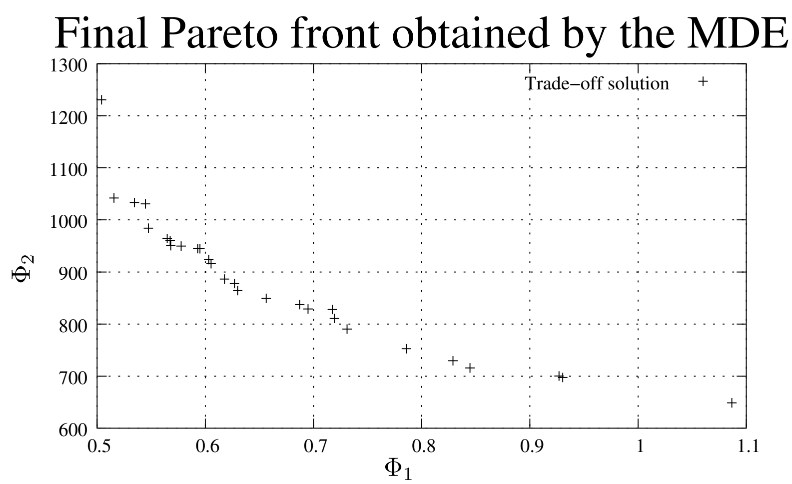

Figure 5. Final set of solutions obtained by the MDE in 10 independent runs