An Evolutionary Approach to Solve a Novel Mechatronic Multiobjective Optimization Problem

Efrén Mezura-Montes[1], Edgar A. Portilla-Flores[2], Carlos A. Coello Coello[3], Jaime Alvarez-Gallegos[4] and Carlos A. Cruz-Villar[4]

[1] Laboratorio Nacional de Informática Avanzada (LANIA A.C.)
Rébsamen 80, Centro, Xalapa, Veracruz, 91000 MEXICO
emezura@lania.mx

[2] Autonomous University of Tlaxcala Engineering and Technology Department
Calz. Apizaquito s/n Km. 15, Apizaco Tlax. 90300 MEXICO
eportilla@ingenieria.uatx.mx

[3] CINVESTAV-IPN (Evolutionary Computation Group)
Departamento de Ingeniería Eléctrica, Sección Computación
Av. IPN No. 2508, Col. San Pedro Zacatenco, México D.F. 07360, MEXICO
ccoello@cs.cinvestav.mx

[4] CINVESTAV-IPN Departamento de Ingeniería Eléctrica, Sección de Mecatrónica
Av. IPN No. 2508, Col. San Pedro Zacatenco, México D.F. 07360, MEXICO
jalvare,cacruz@cinvestav.mx

Summary. In this chapter, we present an evolutionary approach to solve a novel mechatronic design problem of a pinion–rack continuously variable transmission (CVT). This problem is stated as a multiobjective optimization problem, because we concurrently optimize, the mechanical structure and the controller performance, in order to produce mechanical, electronic and control flexibility for the designed system. The problem is solved first with a mathematical programming technique called goal attainment method. Based on some shortcomings found, we propose a differential-evolution-based approach to solve the aforementioned problem. The performance of both approaches (the goal attainment and the modified DE) are compared and discussed, based on quality, robustness, computational time and implementation complexity. We also highlight the interpretation of the solutions obtained in the context of the problem. Finally, some conclusions are established and the future work is described.

1 Introduction

Solving real-world optimization problems is usually a challenging task. These problems arise with relative facility, and its complexity may increase when the problem is relatively unknown since the current systems are more complex.

Currently, several systems can be considered as a mechatronic system due to the integration of the mechanical and electronical elements in such systems. Reason why, it is necessary to use new design methodologies, which consider integral aspects of the systems.

The traditional approach for the design of mechatronic systems, considers the mechanical behavior and the dynamic performance separately. Therefore, the design of mechanical elements involves kinematic and static behaviors while the design of the control system uses only the dynamic behavior. This design approach from a dynamic point of view cannot produce an optimal system behavior [14][23]. Recent works on mechatronic systems design propose a concurrent design methodology which considers jointly the mechanical and control performances.

In this concurrent design concept, some approaches have been proposed. However, these concurrent approaches are based on an iterative process. There, the mechanical structure is obtained in a first step and the controller in a second step, if the resulting control structure is very difficult to implement, the first step must be done again.

On the other hand, an alternative approach to formulate the system design problem is to consider it in the dynamic optimization problem [2][3]. In order to do this, the parametric optimal design of the mechatronic systems is stated as a multiobjective 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 allows us to obtain a set of optimal mechanical and controller parameters in only one step, which could produce a simple system reconfiguration.

We present the parametric optimal design of a pinion–rack continuously variable transmission (CVT). The problem is stated as a multiobjective optimization problem. Two approaches are used to solve it. One is based on a mathematical programming technique called goal attainment method [11] and the other is based on an evolutionary algorithm called differential evolution [17]. The chapter is organized as follows: In Section 2 we detail the transformation of the original problem into a multiobjective optimization problem. In Section 3, we present the mathematical programming method, its adaptation to solve the problem and the obtained results. Afterwards, the evolutionary approach is explained and tested in Section 4. Later, in Section 5, we present a discussion of the behavior of both approaches, based on issues like quality and robustness of the approach, computational time and implementation complexity. Finally, our conclusions and future paths of research are presented in Section 6.

2 Multiobjective Problem

In the concurrent design concept, the mechatronic design problem can be stated as 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.

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 in an established range. There are many CVT's 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 CVT's is well known and there is little to modify regarding its basic operation principles. However, research efforts go on with the controller design and the CVT instrumentation side. Different CVT's types have been used in different industrial applications; the Van Doorne belt or V-belt CVT is the most studied mechanism [19][20]. This CVT is built with two variable radii pulleys and a chain or metal-rubber belt. Due to its friction-drive operation principle, the speed and torque losses of rubber V-belt are a disadvantage. The Toroidal Traction-drive CVT uses the high shear strength of viscous fluids to transmit torque between an input torus and an output torus. However, the special fluid characteristic used in this CVT becomes the manufacturing process expensive. A pinion-rack CVT which is a traction-drive mechanism is presented in [21], this CVT is built-in with conventional mechanical elements as a gear pinion, one cam and two pair of racks. The conventional CVT manufacture is an advantage over other existing CVT's. However, in the pinion-rack CVT, it has determined that the teeth size of the gear pinion is an important factor in the performance of the system.

Because the gear pinion is the main mechanical element of the pinion-rack CVT, determining the optimal teeth size of such mechanical element to obtain an optimal performance could be not easy. On the other hand, an optimal performance system must consider a low energy consumption in the controller. Therefore, in order to obtain an optimal performance of the pinion-rack CVT, it is necessary to propose the parametric optimal design of such system.

The goals of the parametric optimal design of the pinion-rack CVT are to obtain a maximal mechanical efficiency as well as minimal controller energy. Therefore, a MDOP for the pinion-rack CVT will be proposed.

2.1 Description and dynamic CVT model

In order to state the MDOP to the pinion-rack CVT, it is necessary to develop the dynamic model of such 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". As it were said previously, this CVT is built-in with conventional mechanical elements as a gear pinion, one cam and two pair of racks. 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 [2]. 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, 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 $u\left(t\right)$ is the input voltage to the DC motor. $J_{eq} = J_{c2} + M r_p^2 + n^2 J_{c1}$ is the equivalent mass moment of inertia, $J_{c1}$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, $\theta_R\left(t\right) = \frac{1}{2}\arctan\left[\tan\left(2\Omega t - \frac{\pi}{2}\right)\right]$ is the

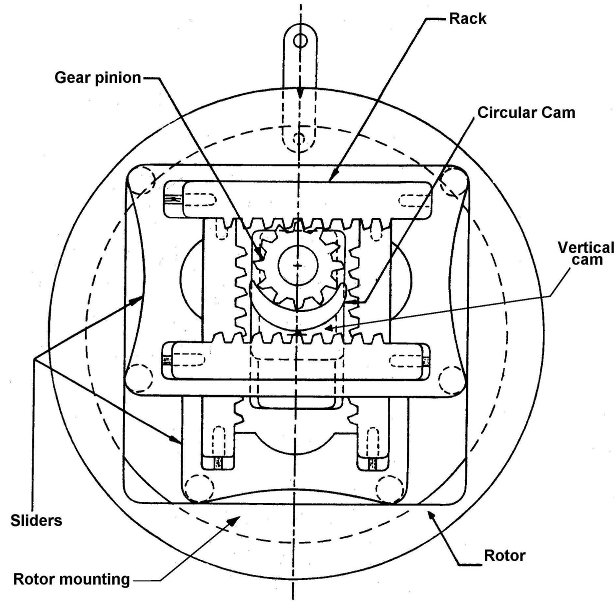

Fig. 1. Main pinion–rack CVT mechanical elements

rack angle meshing. The combined mass to be translated 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.

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

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

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

In order to fulfill the concurrent design concept, the dynamic model of the pinion-rack CVT must be stated with state variables as it indicates in the general problem stated by (1) to (4). With the state variables $x_1 = \dot{\theta}$, $x_2 = i$, $x_3 = e$, $x_4 = \dot{e}$ the dynamic model given by (5) to (7) can be written as:

$$\begin{aligned} \dot{x}_1 &= \frac{\begin{array}{c} AT_m + \left[J_1 A \frac{2x_3}{p_1 p_2} \sin\theta_R\right] x_1^2 - T_L \\ - \left[b_2 + b_1 A^2 + J_1 A \frac{2x_4}{p_1 p_2} \cos\theta_R\right] x_1 \end{array}}{J_2 + J_1 A^2} \\ \dot{x}_2 &= \frac{u\left(t\right) - (\frac{nK_b}{d})x_4 - Rx_2}{L} \\ \dot{x}_3 &= x_4 \\ \dot{x}_4 &= \frac{(\frac{nK_f}{d})x_2 - (b_l + \frac{b_c}{r_p d})x_4 - \frac{T_m}{r_p} \tan\phi \cos\theta_R}{M + \frac{J_{eq}}{d^2}} \end{aligned} \tag{8}$$

Performance criteria and objective functions

The performance of a system is measured by several criteria, one of the most used is the system efficiency because it reflects the energy loss. In the case of the pinion-rack CVT, the mechanical efficiency criterion of the gear systems is used to state the MDOP. This is because the racks and the gear pinion are the principal CVT mechanical elements.

The mathematical equation (9) for mechanical efficiency presented in [22] is used in this work, where μ, N_1, N_2, m, r_1 and r_2 represent the coefficient of sliding friction, the gear pinion teeth number, the spur gear teeth number, the gear module, the pitch pinion radius and the pitch spur gear radius respectively.

$$\eta = 1 - \pi\mu \left(\frac{1}{N_1} + \frac{1}{N_2}\right) = 1 - \frac{\pi\mu}{2m} \left(\frac{1}{r_1} + \frac{1}{r_2}\right) \tag{9}$$

In [2] the speed ratio equation is stated by (10), where ω is the input angular speed and Ω is the output angular speed of the CVT.

$$\frac{\omega}{\Omega} = \frac{R}{r} = 1 + \frac{e}{r}\cos\theta_R \tag{10}$$

Considering $r_1 \equiv r$ and $r_2 \equiv R$, the CVT mechanical efficiency is given by (11).

$$\eta(t) = 1 - \frac{\pi\mu}{N_1}\left(1 + \frac{1}{1 + \frac{e\cos\theta_R}{r}}\right) \tag{11}$$

In order to maximize the mechanical CVT efficiency, $F(\cdot)$ given by (12) must be minimized.

$$F(\cdot) = \frac{1}{N_1}\left(1 + \frac{1}{1 + \frac{e\cos\theta_R}{r}}\right) \tag{12}$$

Equation (12) can be written as (13) which is used to state the MDOP.

$$L_1(\cdot) = \frac{1}{N_1}\left(\frac{2r + e\cos\theta_R}{r + e\cos\theta_R}\right) \tag{13}$$

The second objective function of the MDOP must belong to the dynamic behavior. In order to fulfill that, a proportional and integral (PI) controller structure is used in the MDOP. This is because, despite the development of many control strategies, the PI controller structure remains as one of the most popular approach for industrial processes control due to the adequate performance. Then, in order to obtain the minimal controller energy, the objective function for the MDOP given by (14) is used.

$$L_2(\cdot) = \frac{1}{2}\left[-K_p(x_{ref} - x_1) - K_I\int_0^t (x_{ref} - x_1)dt\right]^2 \tag{14}$$

Objective functions established previously, fulfill the concurrent design concept, since structural and dynamic behaviors will be considered at the same time in the MDOP.

Constraint functions

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

To prevent 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 [16]. Then, in order to avoid 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 [14], 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 [14]. 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 constraint 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 $25.4mm$, then constraint g_9 is imposed.

On the other hand, it can be observed that J_1, J_2 are parameters which are function of the CVT geometry. For this mechanical elements the mass moments of inertia are defined by

$$J_1 = \frac{1}{32}\rho\pi m^4 \left(N + 2\right)^2 N^2 h \tag{15}$$

$$J_2 = \rho h \left[\frac{3}{4}\pi r_c^4 - \frac{16}{6}\left(e_{max} + mN\right)^4 - \frac{1}{4}\pi r_s^4 \right] \tag{16}$$

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.

Design variables

Because the concurrent design concept considers structural an dynamic behaviors at the same time, the vector of the design variables must belong to the mechanical and controller structures. In order to fulfill that, design variables of the mechanical structure related with the standard nomenclature for a gear tooth are used. Moreover, the controller gains K_P and K_I which belong to the dynamic CVT behavior are also used.

Equation (17) establishes a parameter called module m for metric gears, where d is the pitch diameter and N is the teeth number.

$$m = \frac{d}{N} = \frac{2r}{N} \tag{17}$$

On the other hand, 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. Therefore, the vector p^i is proposed in order to establish the MDOP of the pinion-rack CVT.

$$\begin{aligned} p^i &= [p_1^i, p_2^i, p_3^i, p_4^i, p_5^i, p_6^i]^T \\ &= [N, m, h, e_{max}, K_P, K_I]^T \end{aligned} \tag{18}$$

2.2 Optimization problem

In order to obtain the mechanical CVT parameter optimal values, we propose a MDOP given by equations (19) to (26). Where the control signal $u(t)$ is given by (20). As the objective functions must be normalized to the same scale [15], 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 (8) and (20) to (26).

$$\min_{p \in R^6} \Phi(x, p, t) = [\Phi_1, \Phi_2]^T \tag{19}$$

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 the dynamic model stated by (8) and subject to:

$$u(t) = -p_5(x_{ref} - x_1) - p_6 \int_0^t (x_{ref} - x_1) dt \tag{20}$$

$$J_1 = \frac{1}{32} \rho \pi p_2^4 (p_1 + 2)^2 p_1^2 p_3 \tag{21}$$

$$J_2 = \frac{\rho p_3}{4} \left[3\pi r_c^4 - \frac{32}{3} (p_4 + p_1 p_2)^4 - \pi r_s^4 \right] \tag{22}$$

$$A = 1 + \frac{2x_3}{p_1 p_2} \cos \theta_R \tag{23}$$

$$d = r_p \tan \lambda_s \tag{24}$$

$$\theta_R = \frac{1}{2} \arctan \left[\tan \left(2x_1 t - \frac{\pi}{2} \right) \right] \tag{25}$$

$$\begin{aligned}
g_1 &= 0.01 - p_2 (p_1 - 2.5) \leq 0 \\
g_2 &= 6 - \frac{p_3}{p_2} \leq 0 \\
g_3 &= \frac{p_3}{p_2} - 12 \leq 0 \\
g_4 &= p_1 p_2 - p_4 \leq 0 \\
g_5 &= p_4 - \frac{5}{2} p_1 p_2 \leq 0 \\
g_6 &= 12 - p_1 \leq 0 \\
g_7 &= 0.020 - p_3 \leq 0 \\
g_8 &= 0.020 - \left[r_c - \sqrt{2}(p_4 + p_1 p_2) \right] \leq 0 \\
g_9 &= 0.0254 - p_1 p_2 \leq 0
\end{aligned} \tag{26}$$

3 Mathematical Programming Optimization

As we can observe, in a general way, a MDOP is composed by continuous functions given by the dynamic model of the system as well as the objective functions of the problem. In order to find the solution of the MDOP, it must be converted into a Nonlinear Programming Problem (NLP) [4]. Two transcription approaches exist: the sequential and simultaneous approach. In the sequential approach, only the control variables are discretized, this approach are also known as control vector parameterization. In the simultaneous approach the state and control variables are discretized resulting in a large-scale NLP problem which requires of special algorithms of solution [1]. Because of the diversity of algorithms of mathematical programming already established, the transcription of the MDOP into a NLP problem is made by the sequential approach.

The NLP problem which is used to approximate the original problem given by (1) to (4) can be stated as:

$$\min_{p} F(p) \tag{27}$$

subject to:

$$c_i \leq 0 \tag{28}$$

$$c_e = 0 \tag{29}$$

where p is the vector of the design variables, c_i are the inequality constraints and c_e are the equality constraints, respectively. In order to obtain the NLP problem given by (27) to (29), the sequential approach requires the value and the gradient calculation of the objective functions, moreover the gradient calculation of the constraints respect to the design variables must be calculated.

3.1 Gradient calculation and sensitivity equations

To get the gradient calculation for the objective function, we use the following equation:

$$\frac{\partial \Phi_i}{\partial p_j} = \int_{t_0}^{t_f} \left(\frac{\partial L_i}{\partial x} \left[\frac{\partial x}{\partial p_j}(t) \right] + \frac{\partial L_i}{\partial p_j} \right) dt \tag{30}$$

where, it can be seen in the general problem stated by (1) to (4), L_i is the $i-th$ objective function, 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. On the other hand, in order to obtain the partial derivatives $\frac{\partial x}{\partial p_j}$, it is necessary to solve the ordinary differential equations of the sensitivity given by (31).

$$\frac{\partial \dot{x}}{\partial p_j} = \frac{\partial f}{\partial x} \left[\frac{\partial x}{\partial p_j} \right] + \frac{\partial f}{\partial p_j} \tag{31}$$

These sensitivity equations can be obtained taking the time derivatives with respect to p_j of the dynamic model. Due to $\dot{x}$ is a function of the time variable t as well as the design variables p_j (we must consider that p_j are independent of the t), it is fulfilled that:

$$\dot{x} = \frac{dx}{dt} = \frac{\partial x}{\partial t} \tag{32}$$

moreover

$$\frac{d(\frac{\partial x}{\partial p_j})}{dt} = \frac{\partial(\frac{\partial x}{\partial p_j})}{\partial t} = \frac{\partial(\frac{\partial x}{\partial t})}{\partial p_j} = \frac{\partial(\frac{dx}{dt})}{\partial p_j} = \frac{\partial \dot{x}}{\partial p_j} \tag{33}$$

Finally, using the equalities (33) and proposing the following variable:

$$y_j = \frac{\partial x}{\partial p_j} \tag{34}$$

The partial derivatives of x with respect to p_j is now given by the following ordinary differential equations:

$$\dot{y_j} = \frac{\partial f}{\partial x} y_j + \frac{\partial f}{\partial p_j} \tag{35}$$

$$y_j(0) = \frac{\partial x_0}{\partial p_j} \tag{36}$$

3.2 Goal Attainment Method

As we said, in order to transcript the MDOP into a NLP problem the sequential approach is used.The resulting problem is solved using the Goal Attainment Method [11]. In the remaining of the chapter we will refer to it as "MPM" (Mathematical Programming Method). In such technique, a subproblem is obtained as follows:

$$\min_{p,\lambda} G(p,\lambda) \stackrel{\Delta}{=} \lambda \tag{37}$$

subject to:

$$\begin{aligned} g(p) &\leq 0 \\ h(p) &= 0 \\ g_{a1}(p) &= \Phi_1(p) - \omega_1\lambda - \Phi_1^d \leq 0 \\ g_{a2}(p) &= \Phi_2(p) - \omega_2\lambda - \Phi_2^d \leq 0 \end{aligned} \tag{38}$$

where λ is an artificial variable without sign constrain, $g(p)$ and $h(p)$ are the constraints established in the original problem. Moreover, in the last two constraints ω_1 and ω_2 are the scattering vector, Φ_1^d and Φ_2^d are the desired goals for each objective function and Φ_1 and Φ_2 are the evaluated functions.

3.3 Numerical method to solve the NLP problem

In order to solve the resulting NLP problem stated by (37) to (38), the Successive Quadratic programming (SQP) method is used. There, a Quadratic Problem (QP) which is a quadratic approximation to the Lagrangian function optimized over a linear approximation to the constraints is solved. A vector p^i which contains the current parameter values is proposed and the NLP problem given by equations (39) and (40) 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 (39) and (40) 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.

$$\min_{\gamma} QP(p^i) = G\left(p^i\right) + \nabla G^T\left(p^i\right)\gamma + \frac{1}{2}\gamma^T B_i \gamma \tag{39}$$

subject to

$$\begin{aligned} g(p^i) + \nabla g^T\left(p^i\right)\gamma &\leq 0 \\ h(p^i) + \nabla h^T\left(p^i\right)\gamma &= 0 \\ g_{a1}(p^i) + \nabla g_{a1}^T\left(p^i\right)\gamma &\leq 0 \\ g_{a2}(p^i) + \nabla g_{a2}^T\left(p^i\right)\gamma &\leq 0 \end{aligned} \tag{40}$$

3.4 Experiments and Results of the Mathematical Programming Method

In order to carry out the parametric optimal design of the pinion-rack CVT, 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 using Mathlab 6.5.0 Release 13. The system parameters used in numerical simulations were: $b_1 = 1.1 Nms/rad$, $b_2 = 0.05 Nms/rad$, $r = 0.0254m$, $T_m = 8.789Nm$, $T_L = 0Nm$, $\lambda_s = 5.4271$, $\phi = 20$, $M = 10Kg$, $r_p = 4.188E - 03m$, $K_f = 63.92E - 03Nm/A$, $K_b = 63.92E - 03Vs/rad$, $R = 10\Omega$, $L = 0.01061H$, $b_l = 0.015Ns/m$, $b_c = 0.025Nms/rad$ 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 $x_{ref} = 3.2$.

Because 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 (8) and (20) to(26). The optimal solution vector p^1 is shown in Table 1. The goal for Φ_2 was obtained by minimizing this function subject to equations (8) and (20) to(26). The optimal solution vector p^2 for this problem is also shown in Table 1.

Varying the scattering vector can produce different nondominated solutions. In Table 1, two cases are presented: p_A^* is obtained with $\omega = [0.5, 0.5]^T$, p_B^* is obtained with $\omega = [0.4, 0.6]^T$.

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

Table 1. Details of the solutions obtained by the MPM.

Initial search point	Scattering vector
[13.4459, 0.0019, 0.02, 0.0826, 5.000, 0.01]	[0.5, 0.5]
[38, 0.0017, 0.02, 0.0636, 10.000, 1.00]	[0.5, 0.5]
[38, 0.0017, 0.02, 0.0636, 10.000, 1.00]	[0.4, 0.6]
[38, 0.0017, 0.02, 0.0636, 10.000, 1.00]	[0.6, 0.4]
[28.8432, 0.0017, 0.02, 0.0550, 5.024, 0.017]	[0.5, 0.5]
[13.4459, 0.0019, 0.02, 0.0826, 5.000, 0.01]	[0.4, 0.6]
[28.8432, 0.0017, 0.02, 0.0550, 5.024, 0.017]	[0.4, 0.6]
[28.8432, 0.0017, 0.02, 0.0550, 5.024, 0.017]	[0.6, 0.4]
[30.77, 0.0017, 0.02, 0.0694, 5.121, 0.010]	[0.5, 0.5]
[30.77, 0.0017, 0.02, 0.0694, 5.121, 0.010]	[0.4, 0.6]

Table 2. Initial points used for the MPM. Also shown is the corresponding scattering vector.

Run	Time required
1	Diverged
2	23.78 Min
3	Diverged
4	Diverged
5	Diverged
6	Diverged
7	Diverged
8	Diverged
9	Diverged
10	48.5 Min
Average	**36.365 Min**

Table 3. Time required by each run of the MPM. Note that only two runs could converge to a solution. The remaining 8 runs could not provide any result.

As it can be seen in the results in Table 3, 80% of the runs diverged. This behavior shows a high sensitivity of the MPM to the starting point (detailed in Figure 2) 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 MPM per independent run is also summarized in Table 3.

Figure 2 show the mechanical efficiency and the input control of the pinion-rack CVT with both solutions obtained by the MPM (p^1, p^2 and p^*_A respectively). The solution p^*_A was selected because it has the same over achievement of the proposed goal for each objective function.

As we can observe in figure 2, when the teeth number is increased (p_1^*) and their size is decreased (p_2^*), a higher CVT mechanical efficiency is obtained. Also we can observe perturbations in the mechanical efficiency, these are because of a tip-to-tip momentary contact prior to full engagement between teeth can be produced. With the optimal solution this tip-to-tip contact is reduced because a better CVT planetary gear is obtained when the teeth size is decreased. Concluding, the optimal solution implies a lower sensitivity of the mechanical efficiency with respect to reference changes.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. In figure 2 it can be observed that the optimal vector minimizes the initial overshoot of the control input.

Despite the sensitivity of the NLP method, the optimal solutions obtained are good from the mechanical and controller point of view.

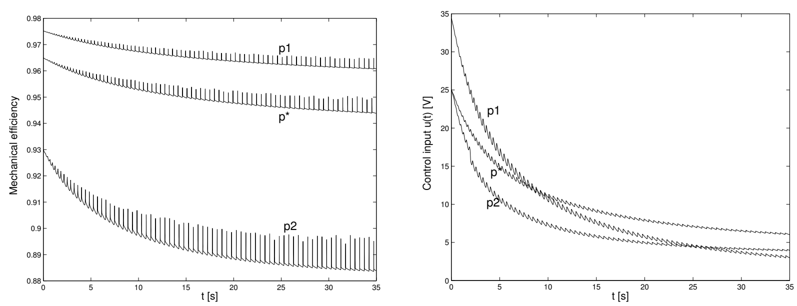

Fig. 2. Mechanical efficiency and Control input for the pinion-rack CVT obtained by the MPM.

4 Evolutionary Optimization

The high sensitivity of the MPM to its initial conditions and its implementation complexity motivated us to solve the problem by using an evolutionary algorithm (EA) because one of its advantages is that competitive results are obtained regardless its initial conditions (i.e. a set of solutions is randomly generated). We selected Differential Evolution [17] because (1) it is an EA which has provided very competitive results when compared with traditional EAs like genetic algorithms, evolution strategies, etc. in real-world problems [6], (2) it is very simple to implement [17] and (3) its parameters for the crossover and mutation operators generally do not require a fine-tuning [13].

DE is an evolutionary direct-search algorithm to solve optimization problems. DE shares similarities with traditional EAs. However it does not use binary encoding as a simple genetic algorithm [8] and it does not use a probability density function to self-adapt its parameters as an Evolution Strategy [18]. Instead, DE performs mutation based on the distribution of the solutions in the current population. In this way, search directions and possible stepsizes depend on the location of the individuals selected to calculate the mutation values.

Several DE variants have been proposed [17]. The most popular is called *"DE/rand/1/bin"*, where "DE" means Differential Evolution, the word "rand" indicates that individuals selected to compute the mutation values are chosen at random, "1" is the number of pairs of solutions chosen to calculate the differences for the mutation operator and finally "bin" means that a binomial recombination is used. A detailed pseudocode of this variant is presented in Figure 3

Four parameters must be defined in DE: (1) the population size, (2) the number of generations, (3) the factor $F \in [0.0, 1.0]$ which scales the value of the differences computed from randomly selected individuals (typically three, where two are used to compute the difference and the other is only added) from the population (row 11 in Figure 3). A value of $F = 1.0$ indicates that the complete difference value is used; and finally, (4) the $CR \in [0.0, 1.0]$ parameter, which controls the influence of the parent on its corresponding offspring; a value of $CR = 0.0$ means that the offspring will take its values from its parent instead of taking their values from the mutation values generated by the combination of the differences of the individuals chosen at random (rows 9-15 in Figure 3).

DE was originally proposed to solve global optimization problems. Moreover, as other EA's, DE lacks a mechanism to handle the constraints of a given optimization problem. Hence, we decided to modify DE to solve constrained multiobjective optimization problems. It is worth remarking that the goal when performing these modifications was to maintain the simpleness of DE.

Three modifications were made to the original DE:

1. The selection criterion between parent and its corresponding offspring was modified in order to handle multiobjective optimization problems.
2. A constraint-handling technique to guide the approach to the feasible region of the search space.
3. A simple external archive to save those nondominated solutions found during the process.

4.1 Selection Criterion

We changed the original criterion to select between parent and offspring (rows 16-20 in Figure 3) based only on the objective function value. As in multiobjective optimization we are looking for a set of trade-off solutions, we used, as traditionally adopted in Evolutionary Multiobjective Optimization [5], Pareto

```
Begin
    G=0
    Create a random initial population X_{i,G} ∀i, i = 1,...,NP
    Evaluate f(X_{i,G}) ∀i, i = 1,...,NP
    For G=1 to MAX_GEN 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(U_{i,G+1}) ≤ f(X_{i,G})) Then
                X_{i,G+1} = U_{i,G+1}
            Else
                X_{i,G+1} = X_{i,G}
            End If
        End For
        G = G + 1
    End For
End
```

Fig. 3. "DE/rand/1/bin" 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_GEN", "CR" and "F" are user-defined parameters. "D" is the dimensionality of the problem.

Dominance as the criterion to select between the parent and its corresponding offspring. The aim is to keep those nondominated solutions in the current population.

A vector $\mathbf{U} = (u_1, \ldots, u_k)$ is said to dominate $\mathbf{V} = (v_1, \ldots, v_k)$ (denoted by $\mathbf{U} \preceq \mathbf{V}$) if and only if $\mathbf{U}$ is partially less than $\mathbf{V}$, 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:

$$\mathcal{P}^* := \{x \in \mathcal{F} \mid \neg\exists\, x' \in \mathcal{F} \;\; \mathbf{F}(x') \preceq \mathbf{F}(x)\}. \qquad (41)$$

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

4.2 Constraint-Handling

The most popular approach to incorporate the feasibility information to the fitness function of an EA is the use of a penalty function. The aim is to decrease the fitness value of those infeasible individuals (which do not satisfy the constraints of the problem). In this way, feasible solutions will have more probabilities to be selected and the EA will approach the feasible region of the search space. However, the main drawback of penalty functions if that they require the definition of penalty factors. These factors will determine the degree of penalization. If the penalty value is very high, the feasible region will be approached mostly at random and the feasible global optimum will be hard to get. On the other hand, if the penalty is too low, the probability of not reaching the feasible region will be high. Based on the aforementioned disadvantage, we decided to avoid the use of a penalty function. Instead, we incorporate a set of criteria based on feasibility originally proposed by Deb [7] and extended to other EA's [9, 10, 12]:

- 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.

We combine the Pareto dominance and also the set of feasibility rules into one selection criterion which substitutes rows 12-16 in Figure 3 as presented in Figure 4:

If ($\mathbf{U}^i_{G+1}$is better than $\mathbf{X}^i_G$ (based on the three selection criteria)) **Then**
    $\mathbf{X}^i_{G+1} = \mathbf{U}^i_{G+1}$
Else
    $\mathbf{X}^i_{G+1} = \mathbf{X}^i_G$
End If

Fig. 4. Modified selection mechanism added to DE to solve our multiobjective optimization problem.

4.3 External Archive

One of the features that distinguish a second generation evolutionary multiobjective optimization algorithm is the concept of elitism [5]. In our modified DE, we included an external archive which includes the set of nondominated solutions found during the evolutionary process. This archive is updated at each generation in such a way that all nondominated solutions from the population will be included in the archive. After that, a nondominance checking is performed with all solutions (the newcomers and also the particles in the archive). Those nomdominated solutions among them will remain in the archive. When the search ends, the set of nondominated solutions in the archive will be reported as the final set of solutions obtained by the approach.

4.4 Results of the EA Approach

The following experiments were performed: 10 independent runs. A fixed set of values for the parameters was used in all runs and they were defined as follows: Population size $NP = 200$, $MAX\_GENERATIONS = 100$; parameters F and CR were randomly generated within an interval. The parameter F was generated per generation in the range $[0.3, 0.9]$ (the differences can be scaled in different proportions without affecting the performance of the approach) and CR was generated per run in the range $[0.8, 1.0]$ (more influence of the mutation operator instead the parent when generating the offspring). These values were empirically derived. This way to define the values for F and CR shows that they do not require to be fine-tuned. We will refer to the evolutionary approach as "EA" (Evolutionary Algorithm).

The experiments were performed in the same platform where the goal attainment experiments were carried out. This was done to have a common point of comparison for the computational time required by each approach.

In table 4 we present the number of nondominated solutions and also the time required per run.

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

Table 4. Time required and number of nondominated solutions found at each independent run by the EA.

The 10 different Pareto fronts obtained are presented in Figure 5.

In order to help the decision maker, we filtered the 10 different set of solutions in order to obtain the final set of nondominated solutions. The final Pareto front obtained from the 10 runs contains 28 nondominated points and it is presented in Figure 6. Finally, the details of the 28 solutions are presented in Table 5.

Figure 7 shows the mechanical efficiency and the input control of the pinion-rack CVT with the optimal solution obtained in the MPM and the solu-

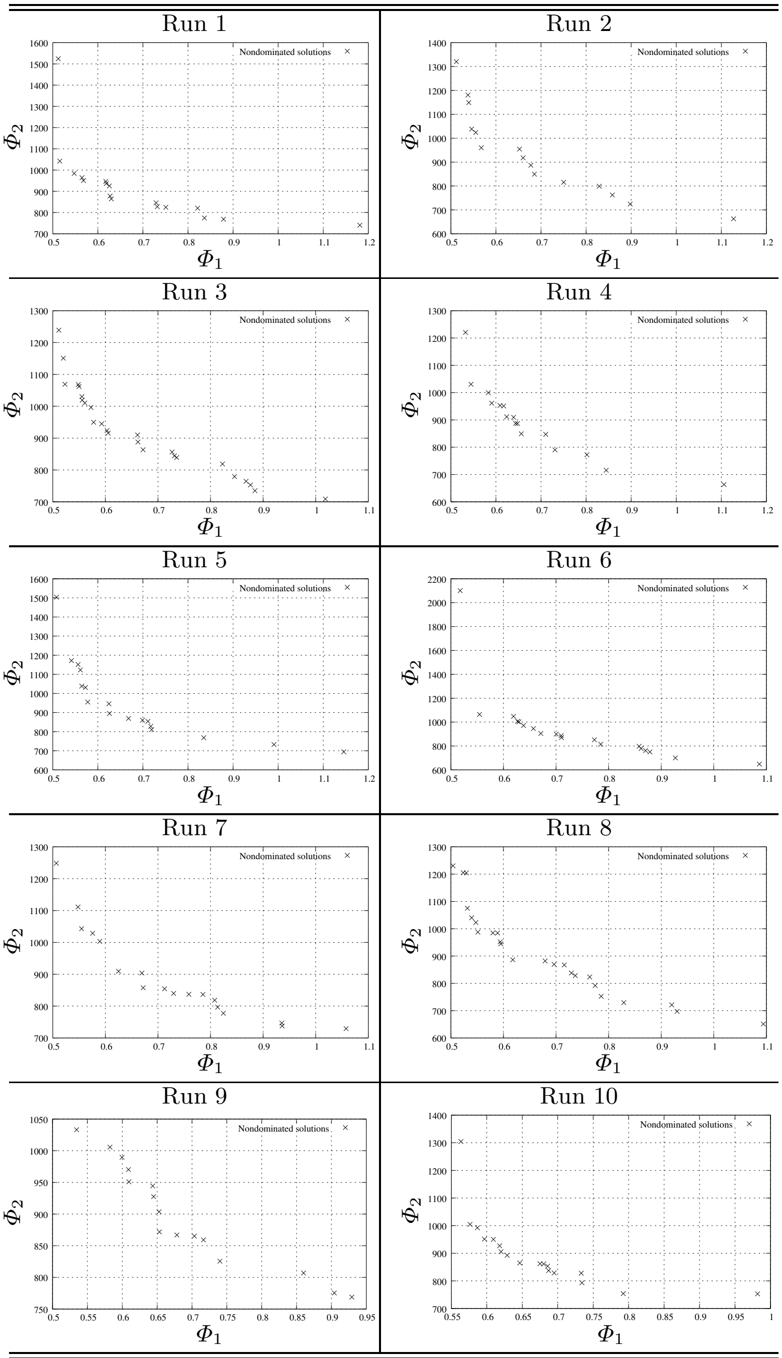

Fig. 5. Different Pareto fronts obtained by the EA in 10 independent runs.

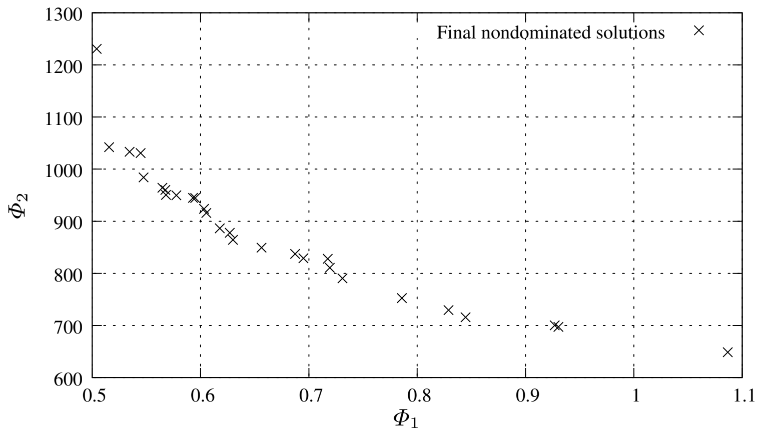

Fig. 6. Final set of solutions obtained by the EA in 10 independent runs

$[N^*, m^*, h^*, e^*_{max}, K^*_P, K^*_I]$	$[\mathbf{\Phi_1(\bullet)}, \mathbf{\Phi_2(\bullet)}]$
[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 5. Details of the trade-off solutions found by the EA. All solutions are feasible.

tion ([30.185435, 0.017, 0.020075, 0.059569, 5.133269,0.019914]) in the middle of the filtered Pareto front obtained with the EA (Figure 6). We can observe that the mechanical efficiency found by the EA is better than MPM solution. We can also see a smooth behavior of the mechanical efficiency for the EA, maintaining a more compact CVT size for the EA solution. However the initial overshoot of the input control is greater than MPM's. These behaviors are obtained with all solutions in the middle of the Pareto front, because a higher teeth number and a corresponding smaller size are obtained (p_1^* was increased and p_2^* was decreased) whereas the input energy controller is greater (p_5^* and p_6^* were increased) in these optimal solutions. In conclusion, from a mechanical point of view, solutions in the middle of the Pareto front, offer many possible system reconfigurations of the pinion-rack CVT.

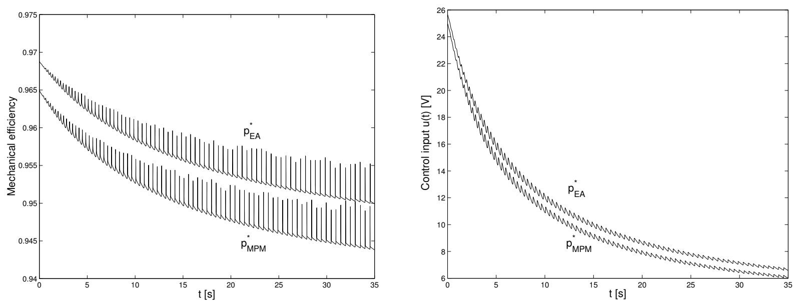

Fig. 7. Mechanical efficiency and Control input for the pinion-rack CVT. obtained by the EA approach

5 Advantages and Disadvantages of Both Approaches

5.1 Quality and Robustness

As we can see, the results provided by the EA were as good as the obtained by the MPM method because the solutions of the last were also nondominated with respect to those found by the EA. However, the EA was not sensitive to the initial conditions (always a randomly generated set of solutions were used). Then, the EA approach provided a more robust behavior than that showed by the MPM. Despite the fact that the results obtained by both approaches are considered similar (from a mechanical and control point of view), as the EA obtains several solutions from a single run, it gives the designer the chance to select from them, the best choice based on his preferences.

5.2 Computational Cost

It is clear, based on the results shown in Tables 3 and 4 for the MPM and the EA approaches respectively, that the EA is the most expensive. However, as it was pointed out in Table 4, the EA obtains a set of nondominated solutions per single run. In contrast, the MP method always returns a single solution per run. Therefore, based on the average time (18.78 Hrs). and the average number of solutions obtained (18.5 solutions), approximately, one solution per hour is obtained. On the other hand, the MPM obtained a solution in spending approximately 36 minutes. Then, we can conclude that the EA requires, roughly, twice the time used by the MPM to find a single solution.

5.3 Implementation Issues

As it was mentioned in Section 3.3, in order to solve the multiobjective optimization problem by using the MPM, a sequential quadratic programming method is used. There, a quadratic programming problem, which is an approximation to the original CVT problem is solved. Based on this issues, some difficulties were detected.

- This method requires the gradient calculation, sensitivity equations and gradient equations of the constraints. In a general way, the number of sensitivity equations is the product between the number of state variables and the number of the design variables. Moreover, gradient equations are related with the number of the design variables. Summarizing, we must calculate: two equations of the objective functions, twenty four sensitivity equations, six gradient equations and fifty four gradient equations of the constraints. On the other hand, with the EA only two equations of the objective functions must be calculated. Therefore, the EA presents an easy reconfiguration.
- Due to the QP problem is an approximation to the original problem and that the constraints are a linear approximation, this problem might be unbounded or infeasible, whereas the original problem is not. With the EA, the original problem is solved. Therefore, the search of the optimal solution is performed indeed in the feasible region of the search space, directly. In this way, in the case of the EA, new structural parameters can be obtained when additional mechanical constraints to the design problem are added. These mechanical constraints could be consider directly in the constraint-handling of the algorithm and no further changes are needed.

It is worth reminding that another additional step related with the use of the MPM is that it requires the minimization process of each objective function independently. This is because the goal attainment method requires a goal for each function to be optimized. This step is not required by the EA. Finally, the EA showed no significant sensitivity to its parameters.

5.4 Goal Attainment to Refine Solutions

It is important to mention that we carried out a set of runs of the MPM by using a nondominated solution obtained by the EA as an starting point. However, the approach was unable to improve the solution in all runs.

6 Conclusions and Future Work

We have presented the multiobjective optimization of a pinion–rack continuously variable transmission (CVT). The aim is to maximize the mechanical efficiency and to mininize the corresponding control. The problem is subject to geometric and strength conditions for the gear pinion of the CVT. Two different approaches were used to solve the problem: A Mathematical Programming method called Goal Attainment and also an evolutionary algorithm. The first one was very sensitive to the initial point of the search (the point must be given by the user and it must be selected carefully), but the computational time required was about 30 minutes to provide a solution. On the other hand, the evolutionary algorithm, which in our case was differential evolution, showed no sensitivity to the initial conditions i.e. a set of solutions generated randomly were used. Besides, the approach did not shown any sensitivity to the values of the parameters related to the crossover and mutation operators. Furthermore, the EA returned a set of solutions in each single run, which gave the designer more options to select the best solutions, based on his preferences. The computational time required for the EA was about 60 minutes to find a solution. The obtained results from both approaches were similar based on quality, but the EA was more robust (in each single run it obtained feasible results). Finally, the EA was clearly easier to implement, which was one of the most clear advantages of the approach.

Our future work consists on designing a preferences-handling mechanism, in order to let the EA to concentrate the search on those regions of the Pareto front where the most convenient solutions must be located. Besides, we will solve other mechatronic problems by using our method.

Acknowledgments

The third author gratefully acknowledges support from CONACyT project number 45683.

References

1. Cervantes A. and Biegler L.T. Large–scale dae optimization using a simultaneous NLP formulation. *AIChe Journal*, 44:1038–1050, 1998.
2. J. Alvarez-Gallegos, C. Cruz-Villar, and E. Portilla-Flores. Parametric optimal design of a pinion–rack based continuously variable transmission. In *Proceedings of the 2005 IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, pages 899–904, Monterey California, 2005.
3. Jaime Alvarez-Gallegos, Carlos Alberto Cruz Villar, and Edgar Alfredo Portilla Flores. Evolutionary Dynamic Optimization of a Continuously Variable Transmission for Mechanical Efficiency Maximization. In Alexander Gelbukh, Álvaro de Albornoz, and Hugo Terashima-Marín, editors, *MICAI 2005: Advances in Artificial Intelligence*, pages 1093–1102, Monterrey, México, November 2005. Springer. Lecture Notes in Artificial Intelligence Vol. 3789,.
4. J.T. Betts. *Practical methods for optimal control using nonlinear programming.* SIAM, Philadelphia, USA, 2001.

5. Carlos A. Coello Coello, David A. Van Veldhuizen, and Gary B. Lamont. *Evolutionary Algorithms for Solving Multi-Objective Problems*. Kluwer Academic Publishers, New York, June 2002. ISBN 0-3064-6762-3.
6. I.L. Lopez Cruz, L.G. Van Willigenburg, and G. Van Straten. Parameter Control Strategy in Differential Evolution Algorithm for Optimal Control. In M.H. Hamza, editor, *Proceedings of the IASTED International Conference Artificial Intelligence and Soft Computing (ASC 2001)*, pages 211–216. Cancun, México, ACTA Press, May 2001. ISBN 0-88986-283-4.
7. Kalyanmoy Deb. An Efficient Constraint Handling Method for Genetic Algorithms. *Computer Methods in Applied Mechanics and Engineering*, 186(2/4):311–338, 2000.
8. David E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Publishing Co., Reading, Massachusetts, 1989.
9. Fernando Jiménez and José L. Verdegay. Evolutionary techniques for constrained optimization problems. In Hans-Jürgen Zimmermann, editor, *7th European Congress on Intelligent Techniques and Soft Computing (EUFIT'99)*, Aachen, Germany, 1999. Verlag Mainz. ISBN 3-89653-808-X.
10. Jouni Lampinen. A Constraint Handling Approach for the Diifferential Evolution Algorithm. In *Proceedings of the Congress on Evolutionary Computation 2002 (CEC'2002)*, volume 2, pages 1468–1473, Piscataway, New Jersey, May 2002. IEEE Service Center.
11. G.P Liu, J. Yang, and J. Whidborne. *Multiobjective optimisation and control*. Research Studies Press, 2003.
12. Efrén Mezura-Montes and Carlos A. Coello Coello. A Simple Multimembered Evolution Strategy to Solve Constrained Optimization Problems. *IEEE Transactions on Evolutionary Computation*, 9(1):1–17, February 2005.
13. Efrén Mezura-Montes, Jesús Velázquez-Reyes, and Carlos A. Coello Coello. Promising Infeasibility and Multiple Offspring Incorporated to Differential Evolution for Constrained Optimization. In H.-G. Beyer and et al., editors, *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'2005)*, volume 1, pages 225–232, New York, June 2005. ACM Press.
14. R. Norton. *Machine Design. An integrated approach*. Prentice-Hall Inc., 1996.
15. A. Osyczka. *Multicriterion optimization in engineering*. John Wiley and Sons, 1984.
16. P. Papalambros and D. Wilde. *Principles of optimal design. Modelling and computation*. Cambridge University Press, 2000.
17. Kenneth V. Price. An introduction to differential evolution. In David Corne, Marco Dorigo, and Fred Glover, editors, *New Ideas in Optimization*, pages 79–108. Mc Graw-Hill, UK, 1999.
18. Hans-Paul Schwefel, editor. *Evolution and Optimization Seeking*. John Wiley & Sons, New York, 1995.
19. P. Setlur, J. Wagner, D. Dawson, and B. Samuels. Nonlinear control of a continuously variable transmission (cvt). In *Transactions on Control Systems Technology, Vol 11, pp. 101-108*. IEEE, 2003.
20. E. Shafai, M. Simons, U. Neff, and H. Geering. Model of a continuously variable transmission. In *First IFAC Workshop on Advances in Automotive Control, pp. 575-593*, 1995.
21. C. De Silva, M. Schultz, and E. Dolejsi. Kinematic analysis and design of a continuously variable transmission. *Mechanism and Machine Theory*, 29(1):149–167, 1994.

22. M. Spotts. *Mechanical design analysis.* Prentice Hall Inc, 1964.
23. H. van Brussel, Németh I. Sas P, P.D. DFonseca, and P. van den Braembussche. Towards a mechatronic compiler. *IEEE/ASME Transactions on Mechatronics*, 6(1):90–104, 2001.