Dynamic Optimization of First-Order Systems via Static Parametric Programming: Application to Electrical Discharge Machining

P. Huguenin*, B. Srinivasan*, F. Altpeter, R. Longchamp***
* Laboratoire d'Automatique, Ecole Polytechnique Fédérale de Lausanne, CH-1015 Lausanne, Switzerland
** Charmilles Technologies S.A., Applied Research, rue du Pré-de-la-Fontaine 8-10, CH-1217 Meyrin, Switzerland

SYNOPSIS

In this paper, it is shown that dynamic optimization problems of first-order systems can be transformed into a static parametric programming problem, where the state plays the role of the parameter. Thus, an optimal feedback law is obtained. This concept is applied to the die-sinking Electrical Discharge Machining, a highly time varying industrial process which necessitates adaptation of machining settings during operation. It is shown that the minimum-time operation of this process is equivalent to choosing the manipulated variables that maximizes the speed of machining at every position.

1 INTRODUCTION

Optimization has gained interest in the industrial community, since it provides an appropriate framework for improving efficiency and cutting production costs. The main bottleneck is that most real-life systems are dynamic, time varying, and uncertain and dynamic optimization of uncertain systems is not straightforward. So, the use of measurements in the optimization framework need to envisaged (1,2).

For first-order systems, this paper exploits the one-dimensional nature of the problem to transform the dynamic optimization problem into a static parametric programming problem. Parametric programming is a concept recently introduced (3,4). The idea is to obtain the optimal solution for different values of the parameters. An interesting twist occurs when the state variables are used as parameters (5). A similar idea, where the state takes the role of the parameter, will be used here.

The methodology is applied to die-sinking Electrical Discharge Machining (EDM) (6,7). The EDM process consists in eroding hard metals by applying high frequency sparks between an electrode and the workpiece, both immersed in dielectric fluid. When machining progresses, it is more difficult for the dielectric to enter the machined cavity to clean it. This leads to debris accumulation which modifies the dielectric properties. Thus, EDM is a highly time varying industrial process and so, the machining settings have to be adapted along the machining operation.

Applying the theoretical developments to EDM indicates that minimum-time operation can be achieved by maximizing the machining speed. The parametric programming approach is used to maximize the speed, with the electrode position acting as the parameter. Thus, a static feedback law, *i.e.* a relationship between position and the manipulated variables, is obtained. Also, issues regarding performing local or global optimization on-line are discussed.

The paper is organized as follows. In Section 2, certain preliminaries concerning constrained dynamic optimization and parametric programming are presented. The main result of transforming a dynamic optimization problem of first-order systems into a static parametric programming problem is discussed in Section 3. Finally, application of the proposed methodology to the EDM process and experimental results are presented in Section 4.

2 PRELIMINARIES

2.1 Constrained Dynamic Optimization

The dynamic optimization problem under constraints can be formulated as follows (1,2,3) :

$$\min_{t_f,u(t)} \square(x(t_f),t_f) \qquad (1)$$

$$\text{s.t. } \dot{x} = f(x,u),\ x(0) = x_0 \qquad (2)$$

$$S(x,u) \square 0,\ T(x(t_f)) \square 0 \qquad (3)$$

where $\square(x(t_f),t_f)$ is the objective function to be minimized, $x \square$ the state, $u \square\ ^{m}$ ($m \geq 1$) the input, $S(x,u) \square\ ^{\square}$ contains the path constraints, and $T(x(t_f)) \square\ ^{\square}$the terminal constraints. The final time t_f may be specified of free. If t_f is free, it acts as an additional decision variable of the optimization, as is the case in (1). All developments will be done for free terminal time.

Here, a special case of the objective function that depends only on the final state $x(t_f)$ is considered. The more general case with an integral term is not addressed here. Equation (2) describes the system dynamics, $f(x,u)$ starting at the state value $x(0) = x_0$. $f(x,u)$ is a scalar function at least once continuously differentiable ($f(x,u) \square C^1$) which can be linear or nonlinear. Note that only first-order systems are treated here.

By means of the Pontryagin's Minimum Principle (PMP), optimization of a *scalar objective function* (1)–(3) can be transformed into the optimization of the *Hamiltonian function* $H(t)$ (8):

$$\min_{t_f,u(t)} H(t) = \square(t) f(x,u) + \square^T S(x,u) \quad (4)$$

$$\text{s.t. } \dot{x} = f(x,u), \ x(0) = x_0 \quad (5)$$

$$\dot{\square} = \square \frac{\partial f}{\partial x} \square \square \square^T \frac{\partial S}{\partial x}; \ \square(t_f) = \left.\frac{\partial \square}{\partial x}\right|_{t_f} + \square^T \left.\frac{\partial T}{\partial x}\right|_{t_f} \quad (6)$$

$$\square f(x,u)\big|_{t_f} + \square^T S(x,u)\big|_{t_f} + \left.\frac{\partial \square}{\partial x}\right|_{t_f} = 0 \quad (7)$$

where $\square(t) \square$ is the adjoint variable (Lagrange multiplier for the dynamic equation), $\square$ the vector of Lagrange multipliers for path constraints and $\square$ the vector of Lagrange multipliers for terminal constraints. Equation (7), termed the *transversality condition*, is needed only when the terminal time is a decision variable. The necessary conditions of optimality are given by:

$$\frac{\partial H}{\partial u} = \frac{\partial f}{\partial u} \square + \square^T \frac{\partial S}{\partial u} = 0 \quad (8)$$

The problem represented by Equations (4) to (7) can be solved by classical dynamic optimization techniques. But, as will be discussed later, the above mentioned problem can be solved using the Static Parametric Programming methodology.

2.2 Static Parametric Programming

Parametric programming is a way of including variations in the optimization scheme. In the context of parametric programming, a parameter is an abstract variable, and the optimal solution needs to be calculated for various values of this abstract variable. In the case of static nonlinear programming, the parametric optimization problem can be formulated as follows (4) :

$$\begin{aligned} &\square^*(\square) = \arg\min_{\square} \square(\square,\square) \\ &\text{s.t. } h(\square,\square) = 0 \\ &\quad\ \ g(\square,\square) \square 0 \end{aligned} \quad (9)$$

where $\square(\square,\square)$ is the cost function whose value depends on the manipulated variable $\square$ and on the vector of parameters $\square$. $h(\square,\square)$ is a set of equality constraints and $g(\square,\square)$ is a set of inequality contraints (including bounds on the parameters). The functions $\square$, g and h are nonlinear.

Parametric programming can be better understood by contrasting it with other optimization approaches:

- *Standard nominal optimization* – no variation in parameters. The parameters $\square^*$ are known and the optimal solution $\square^*$ is sought.
- *Robust optimization* – parameters vary but the variations cannot be measured or estimated. The parameters are known to lie in a region of the parameters space. Then, a solution $\square$ that does not violate the constraints even in the worst case and minimizes the expected cost is sought.

- *Parametric programming* – parameters vary and their variations can be measured or estimated. Then, an optimal solution as a function of the parameter value is sought, *i.e.* $□^*(□)$. In other words a law linking the optimal solution and parameters needs to be computed.

In parametric programming, the law $□^*(□)$ is based on the knowledge of the system's dynamics and the constraints. For most systems, but the simplest ones, getting a general control law turns out to be very difficult. So, typically, the optimal solutions are computed off-line for various values of the parameters and stocked in a look-up table. Then, the on-line implementation only requires table readings of the precomputed solution of the optimization problem (5). Thus, parametric programming can be considered as an indirect adaptive technique similar to gain scheduling.

In the above development, the parameter was left as an abstract variable. An interesting case arises when the state of the system takes the role of the parameter. Such is the case with predictive control (5) and a similar idea will be used here.

3 TRANSFORMATION OF THE DYNAMIC OPTIMIZATION PROBLEM

In this section, the two concepts presented in the preceding section will be used to reformulate the dynamic optimization of first-order systems.

Theorem 1 *The dynamic optimization problem (1)–(3) for first-order dynamic systems can be reformulated into the following static parametric programming problem with the state acting as the parameter:*

$$\begin{aligned} &u^*(x) = \arg\min_u \operatorname{sign}(□(t_f))\, f(x,u) \\ &\text{s.t.} \quad S(x,u)\, □\, 0 \end{aligned} \tag{10}$$

Proof

First, it will be shown that the adjoint variable $□(t)$ does not change sign. Let the input u_i be determined by the constraint S_j. Then, from the necessary condition of optimality (8), the Lagrange multiplier $□_j$ is given by,

$$□_j = □□□\frac{\partial f}{\partial u_i}□ \Big/ □\frac{\partial S_j}{\partial u_i}□ \tag{11}$$

So, each one of the Lagrange multipliers $□_j$ are (i) either zero if the constraint is not active, or (ii) proportional to $□$ as in (11) if the constraint is active. This means that $□^T \frac{\partial S}{\partial x} = □_j □_j \frac{\partial S_j}{\partial x} = k(x,u)\, □$, where $k(x,u)$ is an appropriate function.

This leads to the following dynamic equation for the adjoint variable:

$$\dot{\lambda} = -\lambda\left(\frac{\partial f}{\partial x} + k(x,u)\right) \tag{12}$$

For $\lambda(t)$ to change sign, $\lambda(t)$ should be zero at some time instant. However, equation (12) shows that if $\lambda(t)$ is zero at some time instant, $\dot{\lambda}$ is zero as well. Then $\lambda(t) = 0 \ \forall t$ implying that (4) is not a meaningful problem. Therefore $\lambda(t)$ cannot change sign and has to remain positive or negative for all t. This leads to:

$$\text{sign}(\lambda(t)) = \text{sign}(\lambda(t_f)), \forall t \tag{13}$$

Since $\text{sign}(\lambda(t))$ is determined by $\text{sign}(\lambda(t_f))$, there is no need to integrate the adjoint equations.

The second point to note is that, since $\lambda(t)$ is scalar, minimizing the Hamiltonian $H(t) = \lambda(t) f(x,u)$ under the constraints $S(x,u) \le 0$ is equivalent to minimizing $\text{sign}(\lambda(t)) f(x,u)$ under the same constraints. Since it has been proved in the earlier part that the $\text{sign}(\lambda(t))$ does not change, the optimization of the scalar function (1)–(3), which is equivalent to (4)–(7), can now be written as:

$$\begin{aligned} u^*(x) &= \begin{cases} \arg\min\limits_{u} f(x,u) & \text{if } \text{sign}(\lambda(t_f)) > 0 \\ \arg\max\limits_{u} f(x,u) & \text{if } \text{sign}(\lambda(t_f)) < 0 \end{cases} \\ \text{s.t.} \quad & S(x,u) \le 0 \end{aligned} \tag{14}$$

where the state x takes the role of the parameter. Equation (14) is just a more explicit form of Equation (10), stressing the two possible values for $\text{sign}(\lambda(t_f))$. ■

The crucial point of the result is that λ is a scalar and does not change sign in the considered time interval. So, the dynamic optimization is being converted into a static optimization and the control variable $u(x)$ is a function of the state x only (no switching times are expected). For the same reason it is possible to get rid of the Lagrange multipliers in the optimization formulation and to simply choose the relevant scenario according to $\text{sign}(\lambda(t_f))$.

It follows, for the transformed problem (4)–(7), that the objective function is $f(x,u)$ and not $\phi(x(t_f),t_f)$. In accordance to Equation (6), the sign of $\lambda(t_f)$ depends on $\phi(x(t_f),t_f)$, the objective function of the original problem and the terminal constraints $T(x(t_f),t_f)$. Another interesting aspect is that even if the terminal time is free, it is no longer a decision variable of the transformed problem (4)–(7). The terminal time is either determined by the terminal constraint or through the optimization criterion.

The solution of the static optimization problem (14), has the form of a static feedback law that expresses the control variables as functions of the state, this is what was searched for (9). This methodology relies on the fact that measurement or estimation of the state has to be available.

4 APPLICATION TO DIE-SINKING ELECTRICAL DISCHARGE MACHINING

4.1 System description

Die-sinking Electrical Discharge Machining consists in generating electrical erosive sparks between an electrode and a workpiece in order to shape a well-defined cavity. The space between the electrode and the workpiece, refered to as *the gap*, is about a few micrometers in average. The current and voltage profiles are controlled by dedicated spark generators.

The EDM process runs on two time scales (Figure 1). The first time scale has order of magnitude of microseconds and deals with single erosive *sparks* (Figure 1a). A spark contains three phases, the gap ionization phase (t_d); the discharge phase (t_e), where current flows through the gap and, finally, the off-time phase (t_o) to let the gap recover its dielectric properties. The second time scale (Figure 1b) is much slower since its typical order of magnitude is the second. It is composed of two phases, the active machining time t_u, where a train of sparks is generated as described earlier, and the cleaning phase lasting $2t_r$ where the electrode is lifted up — t_r seconds — and brought back down for an other t_r. The latter phase is meant to clean the gap by flushing through a large vertical displacement. This cleaning phase is necessary to get rid of the machining debris that accumulated during machining phases. This debris accumulation phenomenon is called *contamination* (6).

Among all those time parameters, only t_o and t_u are considered to be manipulated.

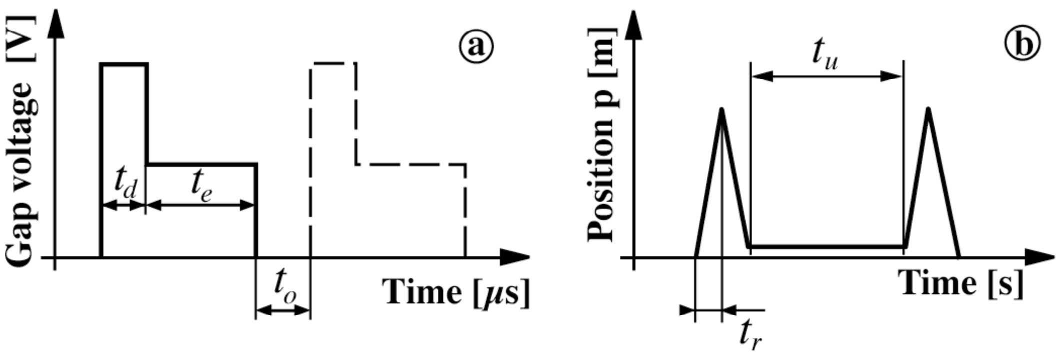

Figure 1 (a) Sketch of a typical spark voltage profile in the fast time scale, with the delay time t_d, the discharge time t_e and the off-time t_o (7). (b) graphical representation, in the slow time scale, of the electrode movements during the active machining phase t_u and the cleaning phase $2t_r$.

The state-space representation of the EDM process can be written as:

$$\dot{p} = V(p,t_o,t_u), \;\; p(0) = 0 \tag{15}$$

where $p(t)$ is the position of the electrode and $V(p,t_o,t_u)$ the machining speed, can be given by:

$$V(p,t_o,t_u) = v_{ref}\,\frac{t_e}{t_d + t_e + t_o}\,\big(1 \square \exp(\square t_o / t_u)\big)\frac{t_u}{t_u + 2t_r}\,\square(p,t_u) \tag{16}$$

where v_{ref} is a normalizing constant with dimension m/s. The first ratio represents the efficiency in the fast time scale. It is balanced by an exponential-like function governed by a relaxation time constant $t_{\square}$. The second ratio represents the efficiency in the slow time-scale. $\square(p,t_u)$ is a nonlinear function of the position p and the machining time t_u. The shape and value of this function are obtained using a nonlinear mean square regression of the data on a function of the form:

$$\square(p,t_u) = (a_1 + a_2 t)\tanh(a_3 p + a_4) + (a_5 + a_6 t);\ \ a_i = a_i(t_u),\ \ i = 1, 2, \ldots, 6 \tag{17}$$

The $a_i(t_u),\ i = 1, 2, \ldots, 6$, in Equation (17) are interpolated. The regression is performed on the data collected from machining operation of a 2.5 mm deep hole. The electrode position is recorded every 10 seconds and the machining speed obtained through numerical differentiation.

4.2 Optimization problem

With the state-space definition of EDM presented in Equation (15) and Equations (1)–(3) where $\square(x(t_f),t_f) = t_f$ as well as $f(x,u) = V(p,t_o,t_u)$, the mathematical description of the general optimization problem for the EDM process can be stated as follows:

$$\min_{t_o(t),t_u(t)} t_f \tag{18}$$

$$\text{s.t. } \dot{p} = V(p,t_o,t_u),\ \ p(0) = 0 \tag{19}$$

$$p(t_f) = p_f > 0 \tag{20}$$

Note that p is the only state variable in the model. It is defined positive in the downward direction and it evolves from 0 at surface to the positive final position p_f fixed *a priori*. With this sign convention, the machining speed $V(p,t_o,t_u)$ is always positive since the material removed from the workpiece is taken away forever.

The optimization searches for t_o and t_u as a function of time.

4.3 Transformation of the optimization problem

In the case of EDM, the changing parameter is the electrode position p. To choose between the two scenarios of Equation (14), it is necessary to find $\text{sign}(\square(t_f))$, *i.e.* the sign of the Lagrange multiplier at final time. Equation (7) is used for this purpose. Since $\square(x(t_f),t_f) = t_f$,

$$\square V(p,t_o,t_u)\Big|_{t_f} + \left.\frac{\partial t}{\partial t}\right|_{t_f} = \square V(p,t_o,t_u)\Big|_{t_f} + 1 = 0 \tag{21}$$

Due to the definition of the machining coordinate, $V(p,t_o,t_u) > 0$ for all p. So $\square(t_f) < 0$. Finally, from Equation (14), the optimization problem applied to EDM reads:

$$t_o(p),t_u(p) = \arg\max_{t_o,t_u} V(p,t_o,t_u) \tag{22}$$

$$p(t = 0) = 0 \tag{23}$$

$$p(t = t_f) = p_f > 0 \tag{24}$$

This demonstrates two things. First, that it is not necessary to choose local suboptimal settings at some places in order to get the overall global optimum (minimum time) and, second, that the minimum time problem leads to a maximum effort solution where the speed has to be *maximized* everywhere along the trajectory. Note that with this transformation, t_o and t_u as functions of p are searched for.

4.4 Results

The problem of maximizing $V(p,t_o,t_u)$ is now considered.

Figure 2 shows the speed at beginning and at end of machining. It is seen that the speed drops down as the hole gets deeper. This behaviour is due to contamination.

The optimal feedback law, $(t_o(p),t_u(p))$, which maximizes machining speed at every point along the machining operation is searched for. The coordinates of the highest point of the speed, as p evolves from zero toward final position p_f, are shown if Figure 3.

The large abrupt change in t_u values, around position 0.8 mm, is particularly worth noticing in Figure 3. Such a behaviour is due to the presence of two local maxima easily distinguishable in Figure 2. The global optimum switches from one local optimum to another. It is obvious in Figure 3 that optimal t_o remains approximately constant for all machining positions.

Since local maxima are present, a question worth asking is: «□What happens if one stays near one of those local maxima, rather than switching from one optimum to the other as machining position increases□?

The answer to this question is found by investigating the three following strategies: t_u □ [1,3] s (global optimization), t_u □ [2,3] s (stay close to local optimum 1) and t_u □ [1,2] s (stay close to local optimum 2). Running the simulation for the three cases yields the results shown in Table 1.

A few comments about those results are to be made. Comparing strategies 1 and 3 in Table 1 makes it clear that the overall machining time for these two cases are almost identical. However, the philosophies behind them are very different.

Strategy 1 needs a global optimizer which is difficult to implement in practice. Such an optimizer has to recognize that optimum 2 it started with is no longer the global optimum, but has been overtaken by another maximum which lies in a different area of the parameter space. So, the optimizer has to move, as fast as possible, toward the new global optimum what can take time and lead to hazardous machining conditions.

In other words, even if a bit faster in theory, the global optimizing policy has to be avoided in practice and replaced by a local optimization strategy over a reduced set of possible machining settings. This reduced set is to be chosen in the neighborhood of the most effective local optimum, even if, in theory, this does not lead to the minimum time realizable strategy.

The results, in terms of machining time for optimal and nonoptimal settings law, are very close to the experimental measurements realized on the EDM machine.

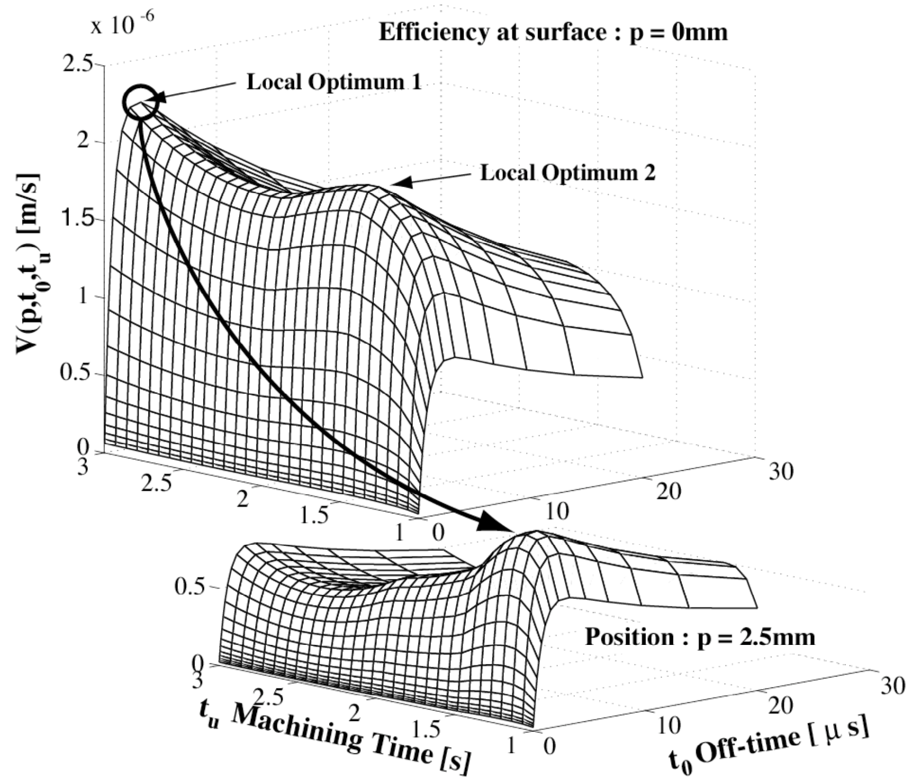

Figure 2 Machining efficiency at initial position $p_0 = 0$ mm and final position $p_f = 2.5$ mm.

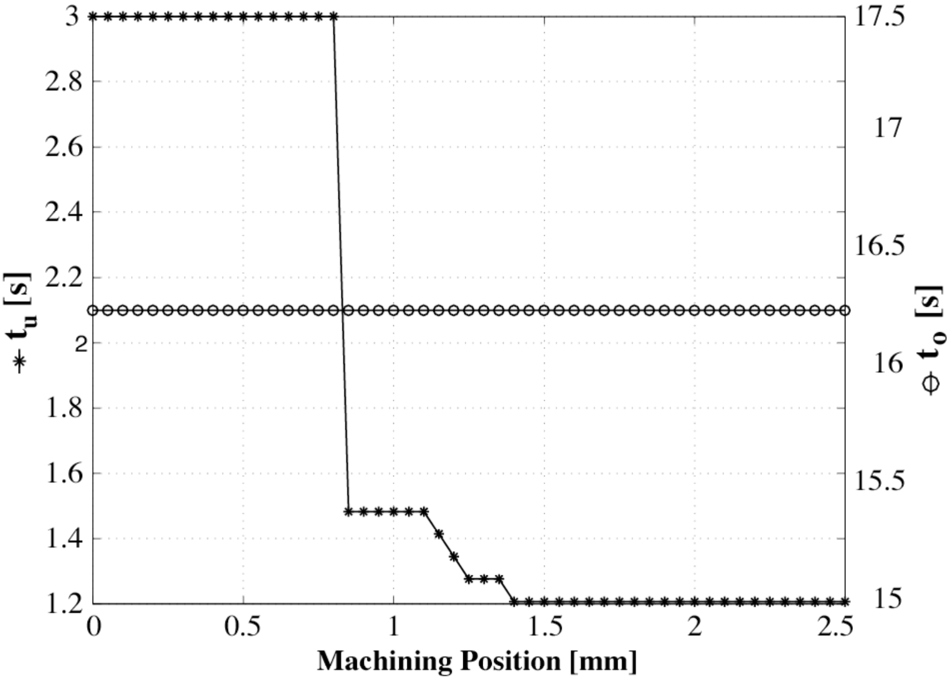

Figure 3 The t_u and t_o profiles, as functions of the position p.

Table 1 Global or local minimum tracking, overall machining time comparison.

Strategy	Range t_u [s]	Resulting t_f [min]	Localisation
1	[1,3]	24.5	Global
2	[2,3]	38.3	Large t_u area
3	[1,2]	24.7	Small t_u area

4 CONCLUSIONS

In this paper, the dynamic optimization problem of a first-order dynamical system subject to variations has been addressed. Using the fact that the adjoint variable does not change sign, the dynamic optimization problem has been transformed into a static parameter programming problem. For the EDM process, it has been shown that a static control law can achieve the minimum time realization by searching for the maximum machining speed at every point of the trajectory.

Remarks have been made to compare the theoretical approach with the experimental one and potential problems arising when switching between local optima have been pointed out. The practical advantages of restraining the search domain of the optimizer to the neighborhood of a *good* local maximum have been studied. This kind of policy leads to a solution very close to the optimal solution in a much safer way than trying to implement a global optimizer to deal with two very different local optima.

REFERENCES

1 Srinivasan, B., Palanki, S. and Bonvin, D. (2003) Dynamic optimization of batch processes : I. Characterization of the nominal solution. *Comput. Chem. Engng.* ,27 , pp. 1-26.

2 Srinivasan, B., Bonvin, D., Visser, E. and Palanki, S. (2003) Role of measurements in handling uncertainty. *Comput. Chem. Engng* ,27 , pp. 27-44.

3 Pesch, H. P. (1994) A practical guide to the solution of real-life optimal control problems. *Control and Cybernetics*, Vol. 23(1-2), pp. 7-60. Special issue on Parametric Optimization.

4 Dua, V. and Pistikopoulos, E. N. (1998) Optimization techniques for process synthesis and material design under uncertainty. *Trans. IchemE,* Vol. 76(A), pp. 408-416.

5 Pistikopoulos, E. N., Dua, V., Bozinis, N. A., Bemporad, A. and Morari, M. (2000) On-line optimization via off-line parametric optimization tools. *Comput. & Chem. Engng*, Vol. 24(2-7), pp. 183-188.

6 Altpeter, F., Cors, J., Kocher, M. and Longchamp, R. (1998) EDM modeling for control. In *12th Int. Symp. for Electromachining, VDI Berichte 1405*, pp. 149-155, Aachen, Germany.

7 Rajurkar, K. P. and Wang, W. M. (1995) Monitoring and control systems for die-sinking and wire EDM processes. *EDM Technology*, Vol. 3, pp. 9-16.

8 Bryson Jr, A. E. and Ho, Y.-C. (1975) Applied Optimal Control : Optimization, Estimation and Control. Taylor & Francis.

9 Knowles, G. (1981) An Introduction to Applied Optimal Control. Academic Press.