Journal of Engineering and Technology for Industrial Applications, 2018. Edition. 13.Vol: 04
https://www.itegam-jetia.org
ISSN ONLINE: 2447-0228
DOI: https://dx.doi.org/10.5935/2447-0228.20180001

New features in dynamic optimization with EcosimPro™

Daniel Navia López[1], Gloria Gutiérrez Rodríguez[2], Pedro C. Herrero[3], César de Prada Moraga[4]

[1,2,4] Departamento de Ingeniería de Sistemas y Automática, Universidad de Valladolid, España, C/Real de Burgos s/n 47011 Valladolid.
[3] EA Internacional, España, C/ San Bernardo 123 Planta 7 28015 Madrid.

Email: daniel.navia@autom.uva.es, gloria@autom.uva.es, pce@ecosimpro.com, prada@autom.uva.es

Received: January 24th, 2018.

Accepted: February 01th, 2018.

Published: March 30th, 2018.

ABSTRACT

The following work shows the application of DASPK in EcosimPro™ with the aim of to calculate the partial derivatives of the state variables with respect to the decision variables (sensitivities) and to improve the dynamic optimization. The implementation of DASPK was tested in three examples: disturbance rejection in a stirred tank with heater, the optimal production policy of hydrogen in a model of a hydrodesulfuration plant and the optimal trajectory of the feeding in a batch bioreactor. The sensitivities with DASPK was compared with another two possible options: obtain the gradients using finite differences and to solve the extended system with the analytical expressions of the sensitivities. The outcomes shows that the implementation of DASPK is a suitable option to ensure optimality in dynamics optimization without the need of writing the sensitivities by hand, nevertheless the CPU time must be improved to become this option attractive.

Keywords: Dynamic optimization, DASPK, Sensitivities analysis, EcosimPro.

I. INTRODUTION

I.1. DYNAMIC OPTIMIZATION

A general problem of dynamic optimization can be written as eq. 1 shows.

$$\begin{aligned}
&\min_{\boldsymbol{u}} J(\boldsymbol{x},\boldsymbol{u},t_f)\\
&s.t.:\\
&\boldsymbol{f}(\dot{\boldsymbol{x}},\boldsymbol{x},\boldsymbol{u},t)=\boldsymbol{0}, \qquad \boldsymbol{x}(t_o)=\boldsymbol{x_0}\\
&\boldsymbol{g}(\dot{\boldsymbol{x}},\boldsymbol{x},\boldsymbol{u},t)\le\boldsymbol{0}\\
&\boldsymbol{e}(\dot{\boldsymbol{x}},\boldsymbol{x},\boldsymbol{u},t_f)\le\boldsymbol{0}
\end{aligned} \tag{1}$$

Where $\boldsymbol{u}\in\mathcal{R}^{n_u}$ represents the decision (or input) variables, $\boldsymbol{x}\in\mathcal{R}^{n_x}$ are the state variables, $\boldsymbol{f}:\mathcal{R}^{n_x}\times\mathcal{R}^{n_u}\to\mathcal{R}^{n_x}$ are the equations of the model (DAE system), $\boldsymbol{g}:\mathcal{R}^{n_x}\times\mathcal{R}^{n_u}\to\mathcal{R}^{n_g}$ are the path constraints and $\boldsymbol{e}\in\mathcal{R}^{n_x}\times\mathcal{R}^{n_u}\to\mathcal{R}^{n_e}\le 0$ are the terminal constraints.

This problem can be solved using an indirect approach, solving the optimal control problem (see Sargent [1] for a summary). This option is not used very often because the difficulty in solving a problem with both bounds undetermined (see Pontryagin's Maximum principle and Dynamic Programming with the Hamilton – Jacobi – Bellman equations [2]).

Instead of this option, the dynamic problem can be discretized in order to use standard NLP techniques to solve it. This is called the direct approach and there are two ways to perform the discretization:

- Simultaneous: the decision and the state variables are discretized transforming the DAE system in a set of algebraic equations using methods of collocation solving the discretized problem with non linear programming techniques [3].

Sequential: only the decision variables are discretized using piece-wise polynomials. With this information, continuous time integration is performed to calculate the state variables solving the DAE system [4] and calculating the value of the objective function and the constraints. If the integration is carried out from the initial to the final time the method is called single shooting, on the other hand if the time is cut in intervals and several integrations are performed is called multiple shooting [5].

A very comprehensive summary of methods for dynamics optimization can be found in the work of Srinivasan and co – workers [6].

In this work we will use the direct approach using the simultaneous method with the single shooting technique, summarized in Figure 1.

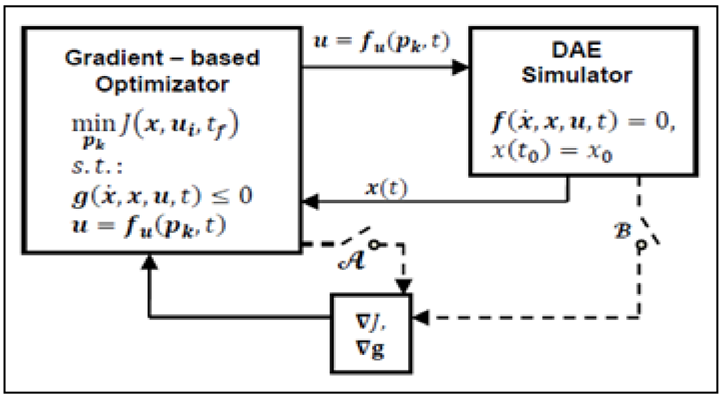


Figure 1: Diagram of the direct sequential method.
Source: Authors, (2018).

Figure 1 shows the parametrization of the decision variables using $p_k \in R^{n_p}$ parameters. Also it can be seen the necessity to estimate the gradients of the objective function and the nonlinear constraints in order to update the value of the decision variables. In this work we will consider only two ways to obtain this information (dashed lines): the optimizator can estimate these gradients using perturbations in the decision variables around the current value and then calculate with finite differences ($\mathcal{A}$ switch ON) or the simulator gives gradient information to the optimizator ($\mathcal{B}$ switch ON).

I.2. SENSITIVITY ANALYSIS

To obtain the gradient information from the simulation, consider the chain rule of the objective function:

$$\nabla J = \left[\frac{\partial J}{\partial u_j}\right], \qquad j: 1 \dots n_u$$
$$\frac{\partial J}{\partial u_j} = \frac{\partial J}{\partial x} S_j, \qquad j: 1 \dots n_u \tag{2}$$

Where $S_j \in \mathcal{R}^{n_x}, j = 1 \dots n_u$ is defined as the vector of sensitivities S_{ij}, which are the partial derivatives of the state variables x_i with respect to the decision variable u_j.

The sensitivities can be obtained from the original DAE system, if a total differentiation is performed.

$$\frac{\partial f}{\partial u_j} = \frac{\partial f}{\partial \dot{x}}\frac{\partial \dot{x}}{\partial u_j} + \frac{\partial f}{\partial x}\frac{\partial x}{\partial u_j} + \frac{\partial f}{\partial p}\frac{\partial p}{\partial u_j} = 0$$
$$\Rightarrow \frac{\partial f}{\partial u_j} = \frac{\partial f}{\partial \dot{x}}\dot{S}_j + \frac{\partial f}{\partial x} S_j + \frac{\partial f}{\partial p}\frac{\partial p}{\partial u_j} = 0 \tag{3}$$

Generating $n_x \cdot n_u$ additional equations that must be solved in each simulation. The entire system to be solved (called extended system), can be summarized merging the model equations f with the sensitivities equations (3), as eq. 4 shows:

$$F(\dot{X}, X, u, t) = 0, \quad X(t_0) = \begin{bmatrix} x_0^T \\ S_{10}^T \\ \vdots \\ S_{n_u 0}^T \end{bmatrix} \tag{4}$$
$$X \in \mathcal{R}^{(n_u+1)\times n_x}$$

I.3. METHODS TO OBTAIN THE SENSITIVITIES

In this work we will use two methods to obtain the value of the sensitivities: Analytical and Numerical. The Analytical Sensitivities approach consists in to simulate the entire system, writing by hand the analytical expressions to the partial derivatives of f with respect to x, which gives a $(n_x + 1)n_u$ DAE set that must be solved in each simulation.

The Numerical Sensitivities method is presented in the works [7]-[8]. The idea is to use the special structure of the extended system: if the Jacobian matrix of the model equations is known, then eq. 5 is a linear ODE system. DASPK is a multi – step integrator that discretizes the time and solves a set of algebraic equations in each step (eq. 5) [9].

$$x_{n+1}^{k+1} = x_{n+1}^{k} - J^{-1}\mathcal{F}\left(x_{n+1}^{k}\right)$$
$$\mathcal{F}(x_{n+1}) = f\left(t_{n+1}, x_{n+1}, \dot{x}_{n+1}^{0} - \frac{\alpha}{h_{n+1}}\left(x_{n+1} - x_{n+1}^{0}\right)\right) \tag{5}$$

It can be seen that the Jacobian of the model is calculated to solve eq. 5, therefore it can be used to obtain the sensitivities in a numerical way. The version 3.0 of DASPK has a utility to get these values, the code allows to choose among three different ways to obtain the sensitivities [8]:

- *Staggered direct method:* At each time step, the states are computed solving eq. 5. With the gradient information, the sensitivities are obtained by solving a linear system. This method is considered inefficient because the Jacobian matrix must be evaluated and factorized at every time step.
- *Simultaneous corrector method:* The state and sensitivity variables are computed simultaneously by solving eq. 5 applied to the extended system from eq 4. This method is more eficiente in terms of evaluating the Jacobian matrix, but has the drawback that a large set of nonlinear equations is solved at each instant.
- *Staggered corrector method:* Similar to the direct corrector, this method first evaluate the state variables and then gets the sensitivities, but the Jacobian isn't updated in every time instant, therefore an additional nonlinear corrector term must be added. This method has the advantage of evaluate and factorize the Jacobian only when is necessary, but has the inconvenience that instead of solving a linear set of equations, a nonlinear one must be solved.

I.4. OBJECTIVE OF THIS WORK

The objective of this work is to test the implementation of DASPK in EcosimPro 4.6 to calculate the sensitivities. EcosimPro is a simulation tool for modelling simple and complex physical processes that can be expressed in terms of: DAE equations or ODE equations and discrete events. Actually this program uses DASSL and 4th order Runge Kutta as numerical integrators [10].

The numerical sensitivities will be used to solve three dynamic optimization problems:

- The rejection of disturbances in a stirred tank that has an electrical heater.

- The optimal policy of hydrogen production in a simplified model of a hydrodesulfuration plant.
- The optimal trajectory of substrate feeding in a batch Bioreactor.

The outcomes obtained with the numerical sensitivities method (DASPK) will be compared with the other options mentioned: finite differences and analytical sensitivities[1]; in terms of optimality, feasibility and CPU time. At last some concluding remarks will be mentioned.

II. APLICATION IN ECOSIMPRO

The following results were obtained in a computer with processor Intel® Core™2 Quad Q9550 @2.83GHz, with 3.25 GB in RAM memory. In all the examples tested the optimizations were performed with a sequential quadratic programming algorithm implemented in the Nag routines [11]: nag_opt_nlp. (e04ucc).

II.1. STIRRED TANK WITH A HEATER

The system is a perfectly agitated tank like the one represented in Figure 2. The temperature (*T*) and the liquid height (*h*) can be modified by changing the valve aperture (*a*) and the voltage of the electrical resistance (V). The caudal and the temperature of the influent (*q*, T_i) are unmeasured disturbances (presented in Figure 3), therefore the manipulated variables (*a*, V) can be modified with the aim of keep the controlled ones (*T*, *h*) closer to their set points ($w_{h,}$, w_t).

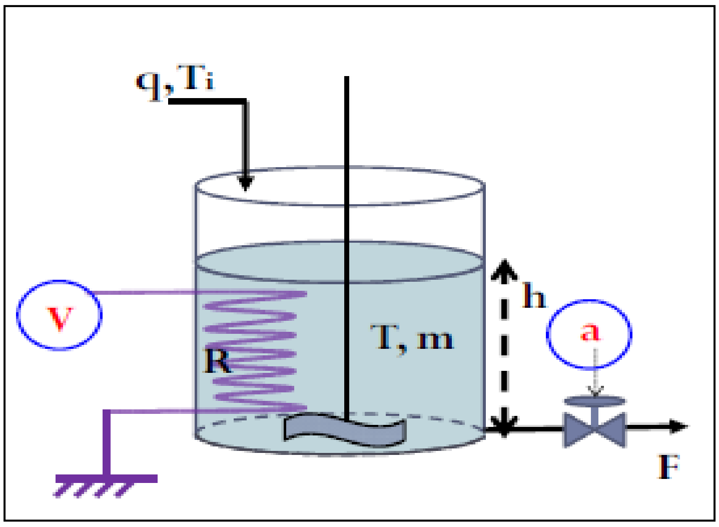


Figure 2: Diagram of the Stirred tank with Heater.
Source: Authors, (2018).

[1]For the examples tested, we will refer to optimization using: *analytical sensitivities* if the sensitivities values are obtained by writing the derivatives of **f** by hand and simulate the extended system; *numerical sensitivities* is referred to obtain the sensitivities by using one of the methods that provides DASPK; *finite differences* if non information of sensitivities is used and the gradients are calculated by perturbations in the decision variables.

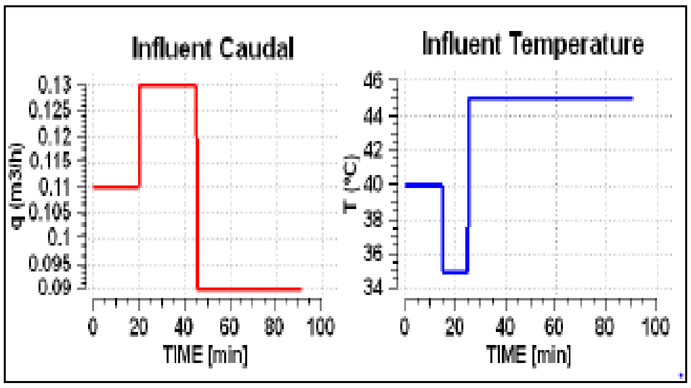


Figure 3: Diagram of the disturbances in the influent.
Source: Authors, (2018).

$$
\begin{aligned}
&\min_{V,a} J(t_f)\\
&s.t.:\\
&\dot{h} = \frac{1}{A}(q - F)\\
&h\,\dot{T} = \frac{q}{A}(T_i - T) + \frac{V^2}{AR\rho C_p} - \frac{U_{env}}{A\rho C_p}(T - T_{env})\\
&F = ak\sqrt{h}\\
&\dot{J} = \alpha_h\left(\frac{h - w_h}{w_h}\right)^2 + \alpha_T\left(\frac{T - w_T}{w_T}\right)^2\\
&V \in \mathcal{V}, \qquad a \in \mathcal{A}
\end{aligned}
\qquad (6)
$$

Where *A* is the area of the tank, ρ and C_p are the density and the heat capacity of the liquid, *k* is a valve constant, U_{env} is a heat transfer coefficient, T_{env} is the outside temperature and α are weights for the objective function.

The dynamic optimization was solved using piece – wise linear parametrization of the manipulated variables with equally spaced intervals. Theprediction horizon was 90 min and the control horizon was 50 min for the voltage and 80 min for the valve aperture. To choose the method to calculate the numerical sensitivities in the optimization (simultaneous, staggered corrector or staggered direct), numerical simulations with DASPK were performed for an increasing number of decision variables (N). These results are in Figure 4.

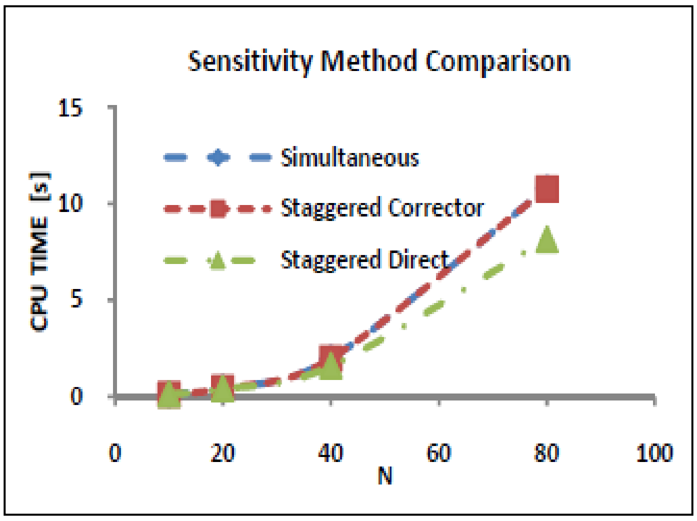
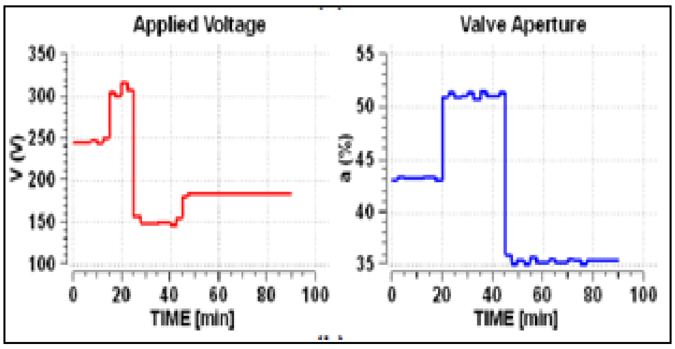
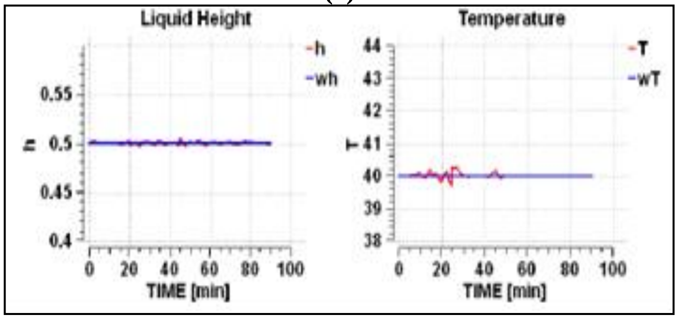


Figure 4: Comparison of the CPU time for the numerical sensitivity methods.
Source: Authors, (2018).

The results shows that the staggered direct method is the most efficient to calculate the sensitivities in terms of time, therefore this method was used to calculate the gradient in the optimization. The optimizations were carried out for an increasing number of decision variables with the aim of to refine the solution.The outcomes for the manipulated variables and for the controlled variables, obtained with the largest number of decision variables tested (24 for *V* and 32 α), are presented in Figure 5, 6 and 7 for: analytical sensitivities, numerical sensitivities with the staggered direct method and finite differences respectively.

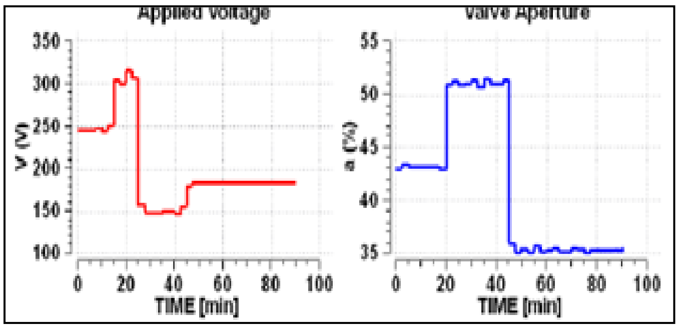


(a)

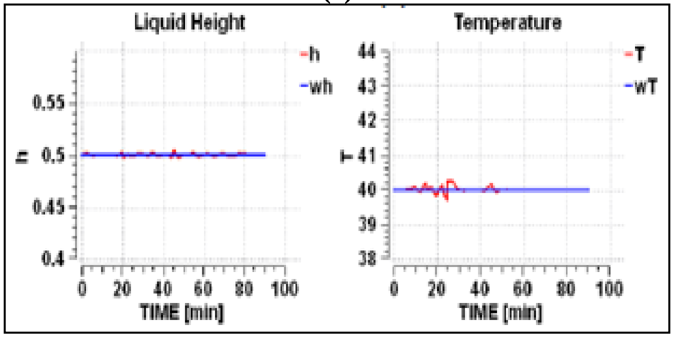


(b)

Figure 5: (a) Manipulated and (b) controlled variables.
Source: Authors, (2018).

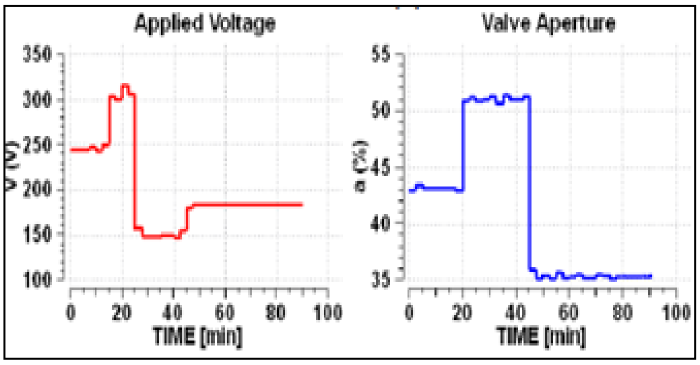


(a)

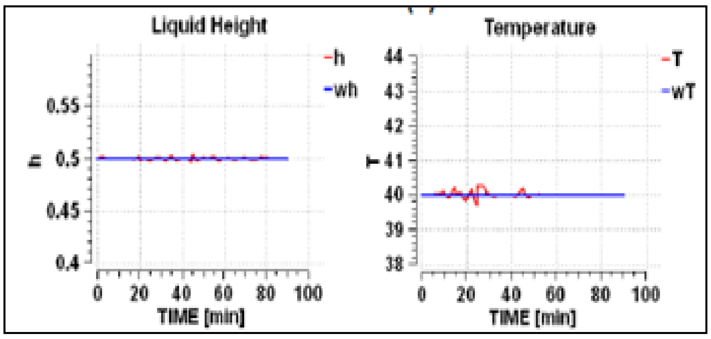


(b)

Figure 6: (a) Manipulated and (b) controlled variables,
Source: Authors, (2018).



(a)



(b)

Figure 7: (a) Manipulated and (b) controlled variables.
Source: Authors, (2018).

The optimization results shows that for the three methods tested in this example, the same trajectories in the manipulated variables are obtained. This trajectories fulfill the KKT conditions for the reformulated NLP problem of eq. 9, therefore we can say that they are locally optimal.

About the controlled variables it can be seen that those values are near to the set points, so the disturbance in the temperature and caudal of the influente is rejected.

About the value of the objective function and the time to perform the optimizations as a function of the decision variables (N), Table I summarizes this.

Table 1: CPU Time and objective function value.

	CPU Time [s]			Objective function		
N	Analytic	Numeri-cal	F.D.	Analy-tic	Nume-rical	F.D.
7	1.746	1.944	2.596	1.937	1.937	1.937
14	5.309	6.288	6.851	0.892	0.892	0.892
28	23.132	24.327	16.050	0.002	0.002	0.002
56	93.530	84.351	32.684	0.003	0.003	0.003

Source: Authors, (2018).

It can be observed that for the three methods tested in this example the results are the same. About CPU time, finite difference seems to be the most efficient method to run the optimization for a large number of decision variables.

II.2. SIMPLIFIED MODEL OF A HYDRODESULFURATION UNIT

The second example is a simplified model of a hydrodesulfuration plant (HDS) represented in Figure 8. The aim of this process is to reduce the súlfur content in the fuel in order to accomplish with the environmental policies.

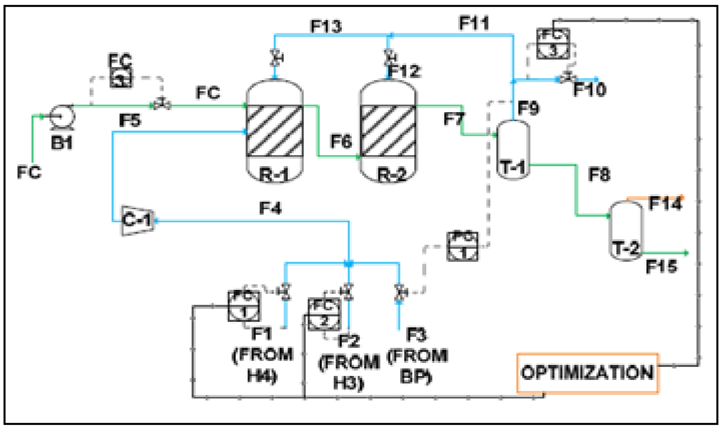


Figure 8: Diagram of a HDS plant.
Source: Authors, (2018).

To produce the desulfuration the rich – sulfur hydrocarbon (FC) is mixed with the hydrogen in a catalyzed reactor (R-1 and R-2). The hydrogen comes from different sources (F_1, F_2 and F_3) with different purities (x_1, x_2 and x_3, ordered from the greater to the lower value). The quantity of hydrogen fed is given by the flow of the hydrocarbon and its sulfur concentration.

The excess of hydrogen is recovered in a separation process (T-1). A fraction of this stream must be purged (F10) in order to ensure a purity of hydrogen inside the reactor greater than a lower bound compatible with the catalyst.

The hydrogen stream that comes into the reactor (F5) also must have a purity greater than a lower bound because the compressor specifications (C-1).

In practice the changes in the hydrocarbon load are scheduled, therefore a dynamic optimization can be formulated in order to obtain an optimal policy of the hydrogen production ensuring that the purities remains above their lower bounds.

Having this in to account and other assumptions: isobaric and isothermal operation, CSTR reactor, perfect separation in T-1, first order dynamics consumption and zero orther kinetics; a dynamic optimization can be formulated when a step change in the hydrocarbon is realized (eq. 7).

$$
\begin{aligned}
&\min_{F_1,F_2,F_{10}} J(t_f) = Cost(t_f)\\
&s.t.:\\
&F_1 + F_2 + F_3 = F_5\\
&x_1F_1 + x_2F_2 + x_3F_3 = x_5F_5\\
&\frac{PV}{ZRT}\dot{x}_{H2} = F_5x_5 - F_X - F_{10}x_{H2},\ \ x_{H2}(t_0) = x_{H20}\\
&F_5 = F_X + F_{10}\\
&\tau\dot{F}_X + F_X = F_{X0}\rho_1, \qquad F_X(t_0) = F_{X0}\\
&C\dot{o}st = C_{H4}F_1x_1 + C_{H3}F_2x_2\\
&x_5^{LO} \le x_5, \qquad x_{H2}^{LO} \le x_{H2}, \qquad F_1, F_2, F_{10} \in \mathcal{F}
\end{aligned}
\tag{7}
$$

Being: x_i and F_i the purity and the molar flow of the stream i, C_j the cost associated to produce hydrogen in the source j, ρ_1 the ratio between hydrogen required and the hydrocarbon load, F_x the hydrocarbon load, P and T the pressure and temperature of the system and Z and R parameters of the gas equation of state.

The dynamic optimization was solved using piece – wise linear parametrization of the manipulated variables, with equally spaced intervals. The simulated time was 10 hours. The path constraints where handled as a soft constraints using a function that penalizes the quadratic violation of its value. This way of handling the constraints modifies the objective function like eq. 8 shows:

$$
\begin{aligned}
C\dot{o}st = C_{H4}F_1x_1 + C_{H3}F_2x_2&\\
+\alpha_{x_5}\max\left(0, x_5^{LO} - x_5\right)^2&\\
+\alpha_{x_{H2}}\max\left(0, x_{H2}^{LO} - x_{H2}\right)^2&
\end{aligned}
\tag{8}
$$

Similar to the previous example, to know which method must be used in the numerical sensitivities, simulations were performed for different number of decision variables. These results are presented in Figure 9.

It can be noted that the staggered direct method is again the most effective way to obtain the numerical sensitivities using DASPK, therefore this method was used in the optimization.

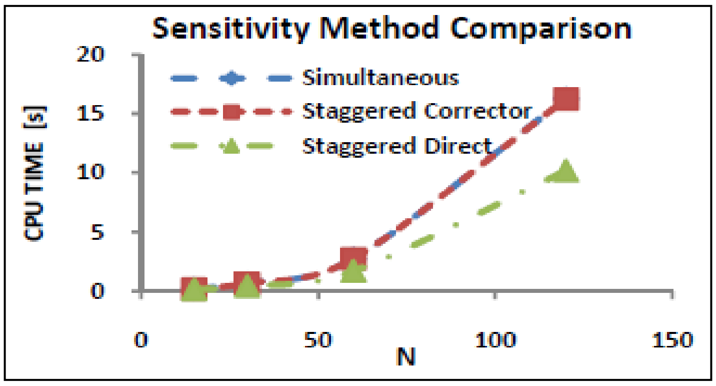


Figure 9: Comparison of the CPU time for the numerical sensitivity methods.
Source: Authors, (2018).

To study the effect of the number of decision variables several optimizations were performed for na increasing number of decision variables in order of refine the solution. The results obtained with the largest number of decision variables tested (120) are presented in Figure 10, 11 and 12 for the optimizations realized by means of: analytical sensitivities, numerical sensitivities with the staggered direct method and finite differences respectively.

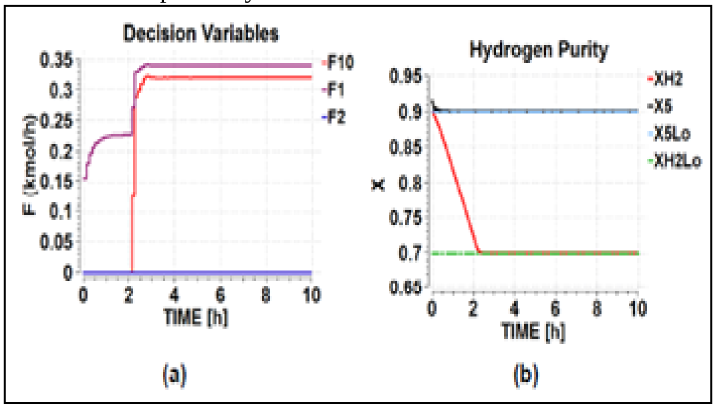


Figure 10: (a) Manipulated and (b) constrained variables.
Source: Authors, (2018).

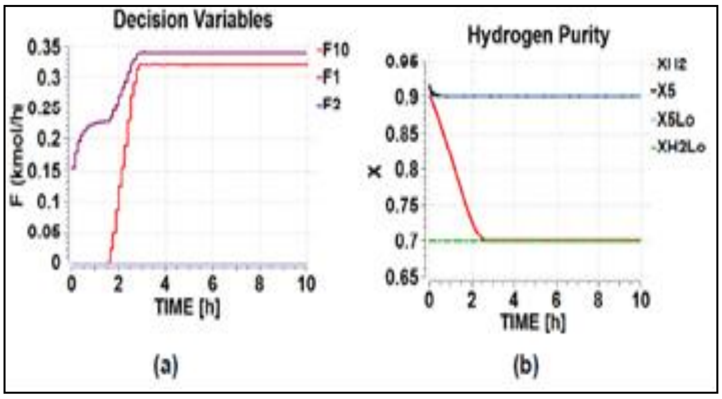


Figure 11: (a) Manipulated and (b) constrained variables.
Source: Authors, (2018).

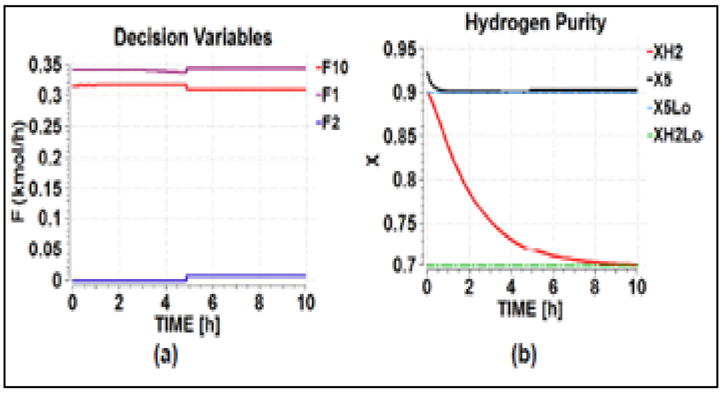


Figure 12: (a) Manipulated and (b) constrained variables.
Source: Authors, (2018).

From the previous Figs. it can be noted that for the three optimizations the hydrogen purities x_{H2} and x_5 are greater or equal to their lower bounds: 0.7 and 0.9 respectively, so the solutions presented are feasible.

About the optimal trajectory for this problem a physical guess can be postulated: because the initial value of the purities are greater than their lower bounds it's logical to expect that the decision variables bring this variables to their limits. There are two ways to do this: closing the purge stream (F10) in order to accumulate impurities and using the minimal quantity of high purity hydrogen (F1). Once that the purity constraint x_{H2} is active, the purge stream must be opened with the aim of keeping this value constant. This behavior can be observed in the outcomes obtained with the optimizations using analytical and numerical sensitivities.

The optimality in both methods is fulfilled (according to the SQP optimizator, KKT conditions were satisfied). On the other hand the results of the optimization using finite differences give trajectories of the decision variables that only ensure feasible solutions keeping the purity inside the reactor greater than its lower bound spending more hydrogen than the necessary. With this method the maximum number of outer iterations of the SQP method were reached (100), with no improvements in the objective function.

The summary of the CPU time and the value of the objective function for all the optimizations realized are presented in Table II as a function of the number of decision variables **N**.

Table II: CPU Time and objective function value.

	CPU Time [s]			Objective function		
N	Analytic	Nume-rical	F.D.	Analy-tic	Nume-rical	F.D.
15	6.465	5.634	0.820	0.275	0.275	0.297
30	26.080	55.742	0.918	0.274	0.274	0.298
60	155.683	232.157	8.529	0.273	0.273	0.300
120	1005.6	1486.9	406.6	0.273	0.273	0.302

Source: Authors, (2018).

Table II shows that for all the optimizations performed, the use of numerical sensitivities and analytical sensitivities has not difference in the optimal solution, instead of finite differences where sub–optimal results were obtained for all the optimizations performed.

About CPU time, it seems that finite diferences is the fastest way to obtain the gradient information, nevertheless, the fact that only feasible solutions were obtained indicates that these optimizations finished earlier because no improvements in the objective functions were observed.

II.3. OPTIMAL TRAJECTORY OF BATCH BIOREACTOR

The system is a fed - batch bioreactor with inhibition and biomass constraint [6], represented in Figure 13. In this system the substrate (**S**) can be fed during all the batch time. The substrate can be transformed by the microorganisms in two products: biomass (**X**) which gives the idea of the growing of the microorganisms and in a desired product designated with letter **P**. These reactions are summarized in eq. 9.

$$S \xrightarrow{\mu} X, \qquad S \xrightarrow{v} P \tag{9}$$

The velocity of these reactions (μ and *v*), defined as the change in the respective concentrations in time, depend on the concentration of the substrate: if S is more concentrated there is more reactant available to produce biomass and the desired product, but there are critical concentrations where the reactions are inhibited decreasing the conversion rate. Having this in to account, a dynamic optimization can be formulated in order to calculate the optimal trajectory of the substrate feeding (**u**) that maximizes the production of P at the final time, subject to: molar balances, equations of velocity of reaction (monod with inhibition kinetics [12]) and path constraints in the biomass concentration, as eq. 10 shows.

$$\begin{aligned}
&\min_{u(t)} J = -P(t_f)\\
&s.t.:\\
&\dot{X} = \mu X - \frac{u}{V}X, \qquad X(0) = X_0\\
&\dot{S} = -\frac{\mu X}{Y_X} - \frac{vX}{Y_P} + \frac{u}{V}(S_{in} - S), \qquad S(0) = S_0\\
&\dot{P} = vX - \frac{u}{V}P, \qquad P(0) = P_0\\
&\dot{V} = u, \qquad V(0) = V_0\\
&\mu = \frac{\mu_m S}{K_m + S + \frac{S^2}{K_i}}\\
&v = \frac{v_m S}{S + K_0}\\
&X(t) \le X^{UP}, \qquad u(t) \in \mathcal{U}
\end{aligned} \tag{10}$$

Being V the reaction volume μ_m, K_m, K_i, V_m and K_0 kinetics parameters.

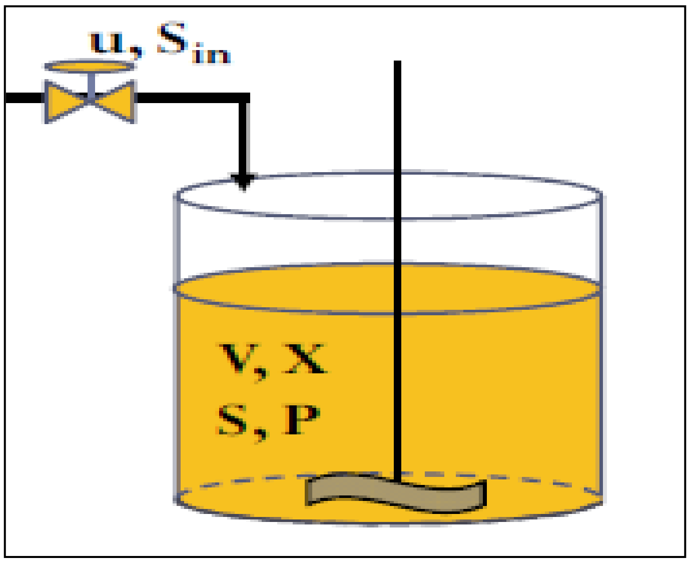


Figure 13: Diagram of a Batch Bioreactor.
Source: Authors, (2018).

The dynamic optimization was solved using piece – wise linear discretization of the decision variable with equally spaced intervals. To handle the path constraint a hard representation was used with a “max” function that register the time where X has the maximum value and is replaced in the constraint. With this representation there is only one constraint to evaluate; unlike other methods where a discretization in time is performed and then the constraint is evaluated in every discrete time increasing the number of constraints.

As in the previous examples, numerical simulations of the model was solved with DASPK to test which of the available methods is the most eficiente to calculate the numerical sensitivities. The results of these simulations in terms of the CPU time are presented in Figure 14 as a function of the number of decision variables.

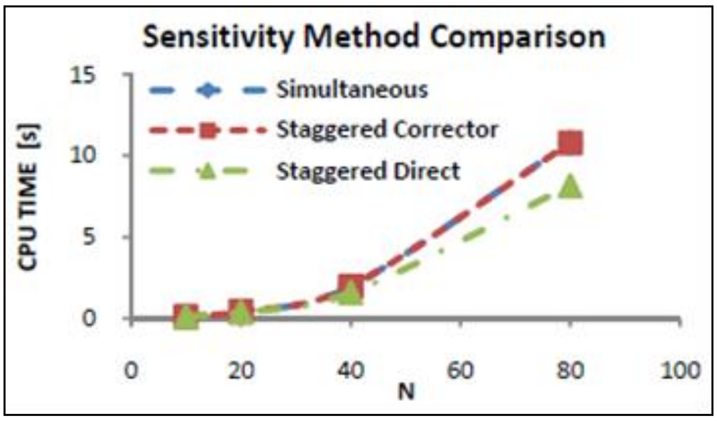
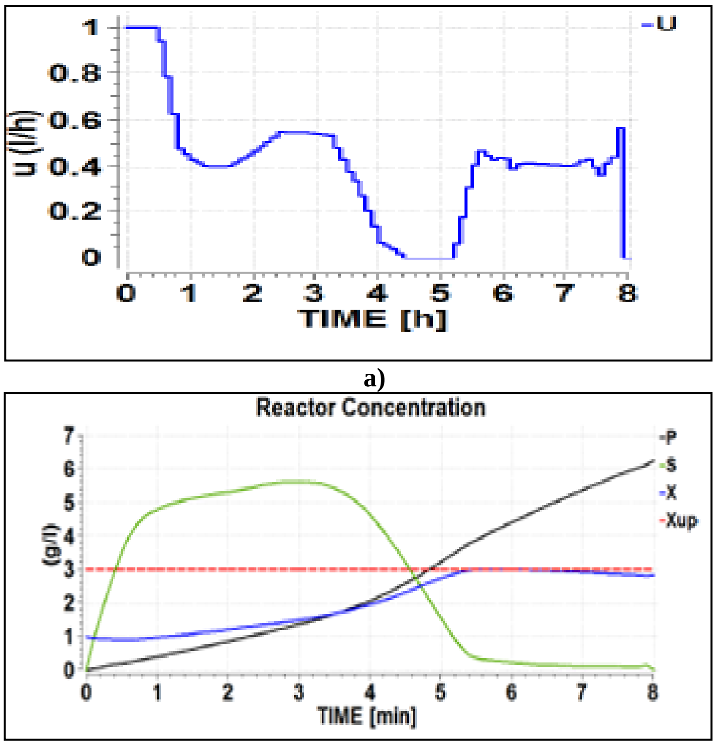


Figure 14: Comparison of the CPU time for the numerical sensitivity methods.
Source: Authors, (2018).

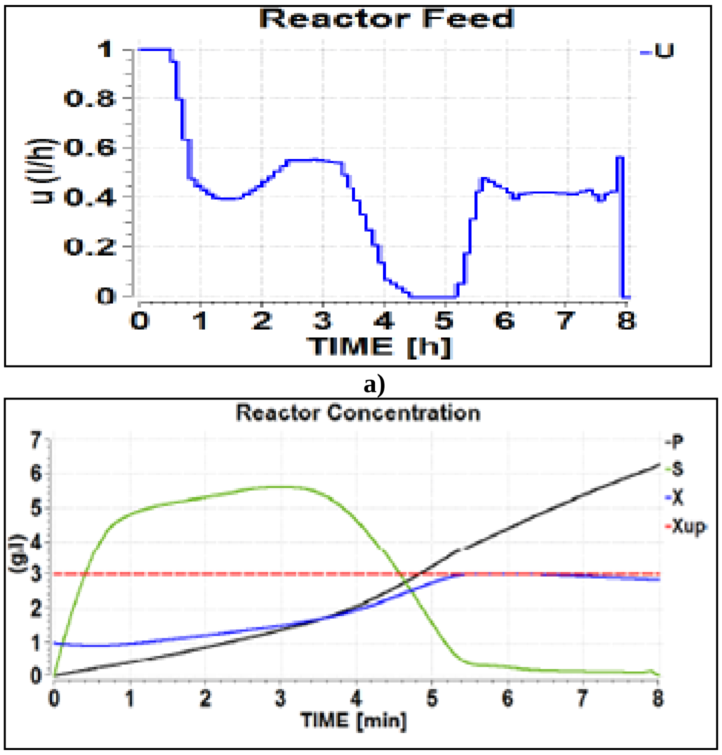
As the previous examples, the most effective method to calculate the numerical sensitivities is the staggered direct one; therefore this method was used in the optimization with numerical sensitivities.

The dynamic optimization was solved for a fixed final time of 8 hours, for an increasing number of decision variables with the aim of to refine the optimal solution. The outcomes of the optimizations solved with the largest number of decision variables tested (80), are presented in Figure 15, 16 and 17 for the optimizations solved with: analytical sensitivities, numerical sensitivities with the staggered direct method and finite differences respectively.



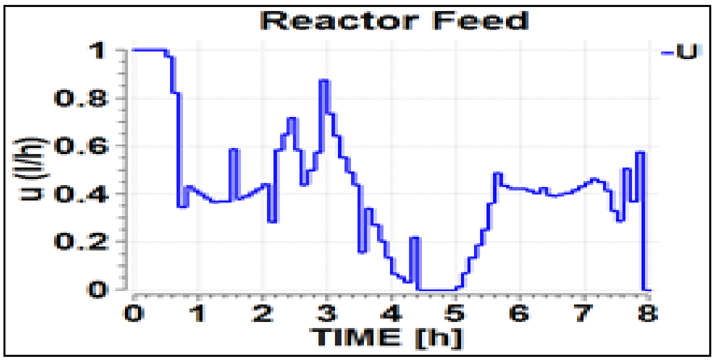
b)

Figure 15: (a) Manipulated and (b) constrained variables.
Source: Authors, (2018).

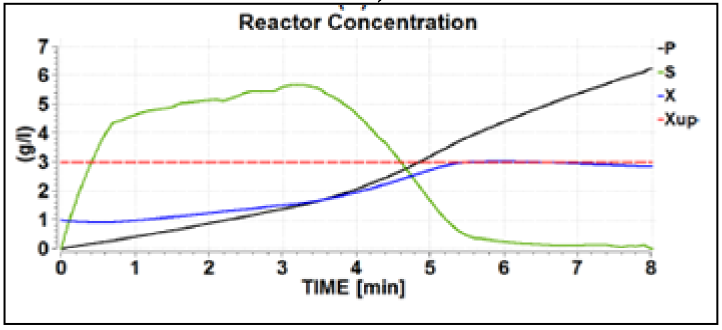


b)

Figure 16: (a) Manipulated and (b) constrained variables.
Source: Authors, (2018).



a)



b)

Figure 17: (a) Manipulated and (b) constrained variables.
Source: Authors, 2018.

The optimal trajectory for this problem, can be found in the work of Kadam and co-workers [13]. In the cited work, the authors distinguish six arcs in the optimal trajectory:

- Time 0 to 0.8h: $\mathbf{u=u^{UP}}$.
- Time 0.8h to 3.8h: sensitivity seeking arc.
- Time 3.8h to 5.4: $\mathbf{u=u^{LO}}$.
- Time 5.4 to 6.6h: sensitivity arc.
- Time 6.6h to 7.9h: **u** remains constant in 0.42 (g/l), keeping the biomass constraint active.
- Time 7.9 to 8h: $\mathbf{u=u^{LO}}$.

The six arcs of the optimal trajectory mentioned before can be easily distinguished in the outcomes of the optimization using analytical and numerical sensitivities. In both cases the optimizator indicates that KKT conditions where fulfilled. On the other hand, the results obtained using finite differences as a gradient estimator shows some variation with respect to the optimal arcs mentioned, this plus the fact that the optimizator gave a warning because the constraints were not satisfied for the required tolerances (1e-4) and no improvements were observed in the merit function, indicates that the solution can be considered as infeasible.

The summary of the objective function value and the CPU time for all the optimizations realized is presented in Table III as a function of the decision variables **N**.

Table III: CPU Time and objective function value.

	CPU Time [s]			Objective function		
N	Analytic	Numeri-cal	F.D.	Analy-tic	Nume-rical	F.D.
10	8.631	2.059	1.457	-6.212	-6.212	-6.210
20	34.714	5.455	9.196	-6.220	-6.220	-6.216
40	212.763	28.755	8.893	-6.227	-6.227	-6.222
80	1169.6	154.02	29.826	-6.232	-6.233	-6.227

Source: Authors, (2018).

About the value of the objective function, it can be noted that the optimizations with analytic sensitivities, numerical sensitivities and finite diferences have very similar solutions, nonetheless the constraints was not satisfied in the finite diferences method for the required tolerances.

About the CPU time, the situation is similar to the previous example for the finite differences method, however the numerical sensitivities gives better results than analytic sensitivities.

III. CONCLUSIONS

From the previous examples tested, local optimal solutions were obtained when the gradient was calculated using the sensitivities obtained by analytical and numerical ways, unlike the calculus of the gradient using finite differences where only in the first example optimality was reached This situation is expectable having in to account that the Nag instructions says that it's imperative to provide gradiente information to the optimizator to ensure optimality [11]. The difference between the methods that ensures optimality is the previous information available, because in the analytical sensitivities it's necessary to calculate by hand the partial derivatives of the state variables with respect to the decision variables before running the optimization, which can be very intractable for large scale systems, in contrast to the numerical sensitivities were no further information is required to estimate the gradient. Hence, from the point of view of optimality the numerical sensitivities calculus with DASPK is a good alternative to be implemented in a simulation software such as EcosimPro™ to solve a dynamics optimization problem. The drawback of the numerical sensitivities implementation is the time used to optimize, which makes this option not very attractive to be used in real time applications. This can be explained for the increase in the simulation time when the number of sensitivity variables is increased (see Fig.4, 9 and 14) because a greater linear system must be solved at each time step and also because the Jacobian has more terms that must be evaluated with finite differences. To overcome this problem, the developers of DASPK indicates that its necessary to use an automatic differentiation application to calculate the Jacobian [8].

Therefore as a future work, the implementation of a numerical differentiation software [14] like TAPENADE [15] must be done, with the aim to improve the CPU time and become DASPK a real option for dynamic optimization in EcosimPro.

IV ACKNOWLEDGMENTS

Erasmus Mundus External Cooperation Window Lot 17 grants (EMECW Lot 17).

V REFERENCES

[1] R. W. H. Sargent, "**Optimal control**," *Journal of Computational and Applied Mathematics*, vol. 124, pp. 361-371, 2000.

[2] A. E. Bryson. **Dynamic optimization.** Menlo Park, CA: Addison Wesley Longman, 1999.

[3] L. T. Biegler, "**Solution of dynamic optimization problems by successive quadratic programming and orthogonal collocation**," *Computers & Chemical Engineering*, vol. 8, pp. 243-247, 1984.

[4] V. S. Vassiliadis, *et al.*, "**Solution of a Class of Multistage Dynamic Optimization Problems. 1. Problems without Path Constraints**," *Industrial & Engineering Chemistry Research,* vol. 33, pp. 2111- 2122, 1994.

[5] H. G. Bock and K. Plitt, "**A multiple shooting algorithm for direct solution of optimal control problems**," in *9th IFAC World Congress*, Budapest, 1984.

[6] B. Srinivasan, *et al.*, "**Dynamic optimization of batch processes: I. Characterization of the nominal solution**," *Computers & Chemical Engineering,* vol. 27, pp. 1-26, 2003.

[7] T. Maly and L. R. Petzold, "**Numerical methods and software for sensitivity analysis of differential-algebraic systems**," *Applied Numerical Mathematics,* vol. 20, pp. 57-79, 1996.

[8] S. Li and L. R. Petzold, "**Design of New DASPK for Sensitivity Analysis** " 1999.

[9] K. E. Brenan, *et al.*, **Numerical solution of initial-value problems in differential-algebraic equations**. Philadelphia: Society for Industrial and Applied Mathematics, 1996.

[10] E. A. Internacional, "**EcosimoPro Modelling and Simulation Software - EL Modelling Language Version 4.6**," ed: Empresarios Agrupados Internacional, 2009.

[11] G. **Numerical Algorithms,** ***NAG C library manual*** *: mark 5*. Oxford, U.K.: Numerical Algorithms Group Ltd., 1998.

[12] J. E. Bailey and D. F. Ollis, **Biochemical engineering fundamentals**, Second ed. New York: McGraw-Hill, 1986.

[13] J. V. Kadam, *et al.*, "**Dynamic optimization in the presence of uncertainty: From off-line nominal solution to measurement-based implementation**," *Journal of Process Control,* vol. 17, pp. 389-398, 2007.

[14] L. B. Rall and G. F. Corliss, "**An introduction to automatic differentiation,"** ***Computational Differentiation***, pp. 1-16, 1996.

[15] V. Pascual and L. Hascoët, "**TAPENADE for C**," in *Advances in Automatic Differentiation*. vol. 64, C. H. Bischof, *et al.*, Eds., ed: Springer Berlin Heidelberg, 2008, pp. 199-209.