NUMERICAL STATIC STATE FEEDBACK LAWS FOR CLOSED-LOOP SINGULAR OPTIMAL CONTROL

Stefan C. de Graaf, Johannes D. Stigter, Gerrit van Straten

Wageningen University,Department of Agrotechnology and Food Sciences
P.O. Box 17, 6700 AA Wageningen, The Netherlands
Fax: +31 317 484957, Email: Stefan.degraaf@wur.nl

Abstract: Singular and non-singular control trajectories of agricultural and (bio) chemical processes may need to be recalculated from time to time for use in closed-loop optimal control, because of unforeseen changes in state values and noise. This is time consuming. As an alternative, in this paper, numerical, nonlinear, static state feedback laws are developed for optimal control on the singular arc that can be applied in closed-loop without the need for iteration. The efficacy of these laws is demonstrated in an example.

Keywords: closed loop, optimal control, nonlinear systems, feedback control.

1. INTRODUCTION

Singular and non-singular control trajectories of agricultural, and (bio) chemical (semi-)batch processes may need to be recalculated in closed-loop optimal control, because of unforeseen changes in state values and noise. This can be achieved by using an iterative optimisation algorithm (Bequette, 1991). Iteration, however, is time consuming. It is, therefore, attractive to look for non-iterative algorithms.

An interesting non-iterative closed-loop optimisation procedure was developed by Palanki, *et al.* (1993) and Rahman and Palanki (1996). They proposed to first determine in open loop the sequence of singular and non-singular intervals and accompanying switching times. Next, they develop symbolic static state feedback laws for the singular trajectories, while during the non-singular trajectories the minimum or maximum control values are used.

Rahman and Palanki recommend using symbolic manipulation software such as MAPLE or MATHEMATICA for the development of static state feedback laws, because of the need to compute a large number of Lie-derivatives. However, in case of complex systems, symbolic manipulation leads to expressions that are difficult to handle. Numerical derivation of optimal static state feedback laws is therefore more convenient.

Magana Jimenez (Magana Jimenez, 2002) developed a MATLAB5.3-ADIFOR2.0-FORTRAN^{compaq}6.0-CONTROL (MAFC) software package that is able to synthesize numerical static state feedback laws for nonlinear systems. In the synthesis of these laws automatic differentiation is incorporated to compute the necessary Lie-derivatives numerically.

In order to be able to use this software package for optimal control, the optimisation problem needs to be cast in a form of a control-affine non-linear system. This paper shows how this can be done, and subsequently describes how numerical static state feedback laws for singular optimal control trajectories can be obtained using MAFC. The novelty of this approach lies in developing state feedback laws for (singular) optimal control by using a software package that is designed for the synthesis of numerical static state feedback laws for non-linear systems. It makes it possible to implement closed-loop optimal control for complex agricultural, and (bio) chemical systems with singular trajectories, which is the main motivation for this research.

The outline of this paper is as follows. First the optimal control problem is cast in the form of a nonlinear system. Next, the static state feedback laws as synthesized by the MAFC software package are presented. Finally, an example is presented that demonstrates that singular trajectories generated by a numerical static state feedback law are comparable to singular trajectories generated open-loop by a gradient method. It also will demonstrate that a numerical static state feedback law is able to adjust singular optimal control trajectories in response to unforeseen changes in state values.

2. OPTIMAL CONTROL PROBLEM

The state-equation of the optimisation problem considered here is:

$$\underline{\dot{x}} = \underline{f}(\underline{x}) + \underline{g}_1(\underline{x})u_1 + \ldots + \underline{g}_m(\underline{x})u_m \qquad x(t_0) = x_0 \quad (1)$$
$$x \in \Re^n, \quad u \in \Re^m$$

The performance equation that has to be minimised is:

$$J = \phi\left(\underline{x}(t_f)\right) \qquad \underline{x}(t_f) = \underline{x}_f \quad (2)$$

In these equations $\underline{x}(t)$ is the system state vector and $u_1(t),\ldots, u_m(t)$ are control inputs. The functions $\underline{f}(\underline{x})$, and $\underline{g}_1(\underline{x}),\ldots,\underline{g}_m(\underline{x})$ are smooth system state vector functions and $\phi\left(\underline{x}(t_f)\right)$ is the final weighting function.

Note that the optimal control problem is control-affine and written in the *Mayer*-formulation (Bryson, 1999; Stengel, 1994), which means that the problem is an end-point optimal control problem. These are not restrictive assumptions, because any non-linear optimisation problem can be cast in the form of equations 1 and 2 by introducing an additional state differential equation for the running costs function $L(x,u)$ of the performance equation:

$$J = \phi\left(\underline{x}_f\right) + \int_{t_0}^{t_f} L(\underline{x},\underline{u})\,dt \quad (3)$$

and by introducing additional state-differential equations for inputs that make the optimisation problem not control-affine, for example:

$$\begin{bmatrix} \underline{\dot{x}} \\ \dot{u} \end{bmatrix} = \begin{bmatrix} \underline{f}(\underline{x}) \\ 0 \end{bmatrix} + \begin{bmatrix} \underline{g}(\underline{x},u) \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} v \qquad \begin{matrix} x(t_0) = x_0 \\ u(t_0) = u_0 \end{matrix} \quad (4)$$

In these equations u is the additional control state vector, $v(t)$ is the new control and $f(\underline{x})$ and $g(\underline{x},u)$ are adjusted smooth system state vector functions.

3. NON-LINEAR SYSTEM REPRESENTING THE OPTIMAL CONTROL PROBLEM

The control-affine optimisation problem, consisting of equations 1 and 2 needs to be (re)formulated as a non-linear system of the form

$$\underline{\dot{\tilde{x}}} = \underline{\tilde{f}}(\underline{\tilde{x}}) + \underline{\tilde{g}}_1(\underline{\tilde{x}})u_1 + \ldots. + \underline{\tilde{g}}_m(\underline{\tilde{x}})u_m \qquad \underline{\tilde{x}}(t_0) = \underline{\tilde{x}}_0 \quad (5)$$
$$\underline{y} = \underline{h}(\underline{\tilde{x}}) \quad (6)$$

in order to be able to synthesize static state feedback laws by the MAFC-software package. This is done by writing a Hamiltonian system for the optimal control problem (Schaft, 1984):

$$\underline{\tilde{x}} = \begin{bmatrix} \underline{x} \\ \underline{\lambda} \end{bmatrix} \quad (7) \qquad \underline{\tilde{f}}(\underline{\tilde{x}}) = \begin{bmatrix} \underline{f}(\underline{x}) \\ -\left[\dfrac{\partial \underline{f}(\underline{x})}{\partial \underline{x}}\right]^T \underline{\lambda} \end{bmatrix} \quad (8)$$

$$\underline{\tilde{g}}(\underline{\tilde{x}}) = \begin{bmatrix} \underline{g}_1(\underline{x}) & \cdots & \underline{g}_m(\underline{x}) \\ -\left[\dfrac{\partial \underline{g}_1(\underline{x})}{\partial \underline{x}}\right]^T \underline{\lambda} & \cdots & -\left[\dfrac{\partial \underline{g}_m(\underline{x})}{\partial \underline{x}}\right]^T \underline{\lambda} \end{bmatrix} \quad (9)$$

$$h(\underline{\tilde{x}}) = \begin{bmatrix} \underline{\lambda}^T \underline{g}_1(\underline{x}) \\ \vdots \\ \underline{\lambda}^T \underline{g}_m(\underline{x}) \end{bmatrix} \quad (10)$$

In this system, $\underline{\tilde{x}}$ is a new system state vector consisting of the system state vector $\underline{x}(t)$ and a vector of *Lagrange* multipliers (costates) $\underline{\lambda}$ with an equal length as the length of the system state vector $\underline{x}(t)$. The vector $\underline{y}$ is the output vector and $\underline{\tilde{f}}(\underline{\tilde{x}})$, $\underline{\tilde{g}}_1(\underline{\tilde{x}}),\ldots.,\underline{\tilde{g}}_m(\underline{\tilde{x}})$ and $\underline{h}(\underline{\tilde{x}})$ are smooth system state vector functions.

Controlling the outputs h of this Hamiltonian system with initial values for $\underline{x}_0$ and $\underline{\lambda}_0$ such that the outputs are set to zero is equivalent to fulfilling the necessary optimality conditions of the original optimal control problem:

$$H(\underline{x},\underline{\lambda},\underline{u}) = \underline{\lambda}^T \left(\underline{f}(\underline{x}) + \underline{g}_1(\underline{x})u_1 + \ldots + \underline{g}_m(\underline{x})u_m\right) \quad (11)$$

$$\underline{\dot{x}} = \frac{\partial H}{\partial \underline{\lambda}} \qquad \underline{x}(t_0) = \underline{x}_0 \quad (12)$$

$$\underline{\dot{\lambda}} = -\frac{\partial H}{\partial \underline{x}} \qquad \underline{\lambda}(t_f) = \frac{\partial \phi}{\partial \underline{x}}(t_f) \quad (13)$$

$$\underline{0} = \frac{\partial H}{\partial \underline{u}} \qquad \forall t \in [0,t_f] \quad (14)$$

The equivalence holds if the initial values for $\underline{\lambda}_0$ are selected in such a way that the optimality condition for $\underline{\lambda}(t_f)$ is also met. Values for $\underline{\lambda}_0$ are obtained by solving the optimal control problem (equations 1 and 2) open-loop as will be discussed in section 5.

4. NUMERICAL STATIC STATE FEEDBACK LAWS SYNTHESIZED BY MAFC

The MAFC software package, developed by Magana (2002) is able to synthesize numerical static state feedback laws, $\underline{u}$ for nonlinear Hamiltonian system presented by equations 5 to 10, and thus enables the synthesis of static state feedback laws that perform singular optimal control. In these laws the vectors $\underline{k}(\underline{\tilde{x}})$ and $\underline{l}(\underline{\tilde{x}})$ are smooth vector functions and the vector $\underline{y}_{sp}$ is the vector of output setpoints that acts as the new vector of inputs:

$$\underline{u} = -\underline{k}(\underline{\tilde{x}}) + \underline{l}(\underline{\tilde{x}})\underline{y}^{sp} \tag{15}$$

The vector functions $\underline{k}(\underline{\tilde{x}})$ and $\underline{l}(\underline{\tilde{x}})$ are such that the closed loop system is decoupled, which means that individual input-output channels are separated (Isidori, 1989; Nijmeijer and Schaft, 1990). This property is inherent to the theory of the synthesis of static state feedback laws. Each output is also forced to follow an r-th order linear exponential trajectory if this output is not at its setpoint value. This trajectory is defined by (Kravaris, et al., 1997; Magana Jimenez, 2002):

$$\begin{gathered} y_1 + \varepsilon_{1,1}\frac{dy_1}{dt} + \dots + \varepsilon_{1,r_1}\frac{d^{r_1}y_1}{dt^{r_1}} = y_1^{sp} \\ \vdots \\ y_m + \varepsilon_{m,1}\frac{dy_m}{dt} + \dots + \varepsilon_{m,r_m}\frac{d^{r_m}y_m}{dt^{r_m}} = y_m^{sp} \end{gathered} \tag{16}$$

where $\varepsilon_{1,1},\dots,\varepsilon_{r_m,r_m}$ are pre-selected constant tuning parameters, assigning specific eigenvalues to the output dynamics. The parameters $r_1,\dots,r_m$ are relative degrees, i.e. the smallest integers such that the r^{th} derivative of the output y with respect to t depends explicitly on the input $\underline{u}$.

The vector functions $\underline{l}(\underline{\tilde{x}})$ and $\underline{k}(\underline{\tilde{x}})$ are defined by (Magana Jimenez, 2002):

$$\underline{l}(\underline{\tilde{x}}) = C^{-1}\begin{bmatrix} \varepsilon_1^{r_1} & .. & 0 \\ : & .. & : \\ 0 & .. & \varepsilon_m^{r_m} \end{bmatrix}^{-1} \tag{17}$$

$$\underline{k}(\underline{\tilde{x}}) = C^{-1}\begin{bmatrix} \varepsilon_1^{r_1} & .. & 0 \\ : & .. & : \\ 0 & .. & \varepsilon_m^{rm} \end{bmatrix}^{-1}\begin{bmatrix} \sum_{j=0}^{r_1}\binom{r_1}{j}\varepsilon_1^{r_1-j}L_{\underline{\tilde{f}}}^{r_1-j}h_1(\underline{\tilde{x}}) \\ \sum_{j=0}^{r_2}\binom{r_2}{j}\varepsilon_2^{r_2-j}L_{\underline{\tilde{f}}}^{r_2-j}h_2(\underline{\tilde{x}}) \\ : \\ \sum_{j=0}^{r_m}\binom{r_m}{j}\varepsilon_m^{r_m-j}L_{\underline{\tilde{f}}}^{r_1-j}h_m(\underline{\tilde{x}}) \end{bmatrix} \tag{18}$$

with $\binom{r}{j} = \dfrac{r!}{(j!)((r-j)!)}$ and $0! = 1$

In these equations and $L_{\underline{\tilde{f}}}^{r_1-1},\dots,L_{\underline{\tilde{f}}}^{r_m-1}$ are Lie-derivative operators (Isidori, 1989; Nijmeijer and Schaft, 1990) and C is the following matrix:

$$C = \begin{bmatrix} L_{\underline{\tilde{g}}_1}L_{\underline{\tilde{f}}}^{r_1-1}h_1 & .. & L_{\underline{\tilde{g}}_m}L_{\underline{\tilde{f}}}^{r_1-1}h_1 \\ L_{\underline{\tilde{g}}_1}L_{\underline{\tilde{f}}}^{r_2-1}h_2 & .. & L_{\underline{\tilde{g}}_m}L_{\underline{\tilde{f}}}^{r_2-1}h_2 \\ .. & .. & .. \\ L_{\underline{\tilde{g}}_1}L_{\underline{\tilde{f}}}^{r_m-1}h_m & .. & L_{\underline{\tilde{g}}_m}L_{\underline{\tilde{f}}}^{r_m-1}h_m \end{bmatrix} \tag{19}$$

Static state feedback laws cannot be obtained if the C-matrix is singular. Matrix singularity will occur if, among other things there is no r^{th} time-derivative of the output y that explicitly depends on the input $\underline{u}$.

Note that Palanki, *et al.* (1993) develop symbolic static state feedback laws by eliminating the costates, thereby eliminating possible unstable costate-dynamics. These laws consist of Lie-brackets instead of Lie-derivatives. In contrast, MAFC calculates numerical laws using Lie-derivatives.

More details on static state feedback laws can be found in Magana (2002), Isidori (1989) and Nijmeijer and Van der Schaft (1990).

5. SYNTHESIS AND APPLICATION GUIDE OF THE STATIC STATE FEEDBACK LAWS

Following Palanki, *et al.* (1993) and Rahman and Palanki (1996) static state feedback laws are developed and applied in closed-loop optimal control according to the following procedure:

1. Calculate open-loop optimal state, costate, input and output trajectories for the optimal control problem, using optimisation methods such as gradient methods presented by Bryson (1999) and De Graaf (2004).
2. Determine which optimal trajectory intervals are singular and monitor the switching times that mark each beginning and end of these intervals. Also determine the state and costate values at the

each switching time that marks the beginning of an interval.

3. If needed, make the optimal control problem control-affine. Then (re)formulate the control-affine optimal control problem in a Hamiltonian nonlinear system according to equations 5 to 10.
4. Calculate relative degrees of this nonlinear system at each switching time that marks the beginning of a singular optimal trajectory interval, using MAFC. Check whether the relative degrees change or whether the C-matrix is singular somewhere in this singular optimal trajectory interval. If this happens then stop, as static state feedback laws cannot be obtained in this case.
5. Calculate the static state feedback laws, using MAFC.
6. While on-line, observe the states, compute the costates by simulating the process on-line with equations 5 and 6, and apply in closed-loop the static state feedback laws for singular intervals and maximum or minimum input values for the non-singular intervals. Switch from singular tot non-singular intervals or vice versa at the switching times determined in open loop.

6. VALIDATION OF STATIC STATE FEEDBACK LAWS

An example taken from Srinivasan *et al.* (2000) is used for two validation objectives. The first objective is to demonstrate that singular trajectories generated by a numerical static state feedback law are comparable to singular trajectories generated open-loop by a gradient method presented by De Graaf (2004). The second objective is to demonstrate that a numerical static state feedback law is able to adjust singular optimal control trajectories in response to unforeseen changes in state values.

The synthesis and application guide, presented in section 5 was used to synthesize and apply the static state feedback law.

Srinivasan *et al.* (2000) calculated optimal trajectories for one input of a non-linear system consisting of two simultaneous chemical reactions taking place in a jacket batch reactor. The optimal control problem is described by:

$$f(\underline{x}) = \begin{bmatrix} -p_1 x_1 x_2 \\ -p_1 x_1 x_2 - p_2 x_2^2 \\ 0 \\ -p_1 x_1 x_2 x_3 \\ p_4 x_2^2 \end{bmatrix} \quad (20)$$

$$g(\underline{x}) = \begin{bmatrix} -\frac{x_1}{x_3} & -\frac{p_3 - x_2}{x_3} & 1 & 0 & -\frac{x_5}{x_3} \end{bmatrix}^T \quad (21)$$

The performance equation that has to be minimized is:

$$J = x_4\left(t_f\right) \quad (22)$$

The initial state values are in table 1. The parameter values are in table 2. The input u and the final state values of x_2 and x_5 are constrained. These constraints are in table 3.

Table 1: Initial state values

x_1	0.72	x_4	0
x_2	0.05	x_5	0
x_3	1		

Table 2: Parameter values of the system

p_1	0.053	p_3	5
p_2	0.256	p_4	0.128

Table 3: Constraints on inpus u and final values of states x_2 and x_5

	lower bound	upper bound		lower bound	upper bound
$u(t)$	0	$1.0\cdot10^{-3}$			
$x_2(t_f)$		0.025	$x_5(t_f)$		0.15

Step 1: Optimal trajectories of the states, costates, input and output were calculated using the ACW-gradient-gradient algorithm (de Graaf, 2004). These trajectories are plotted with dotted lines (----) in figures 1 to 4 and figures 5 to 8.

Step 2: These figures show that the optimal control trajectory consist of a singular optimal control trajectory in the time interval 25 tot 200 minutes, because outside this time-interval the input is on its upper or lower bound. A static state feedback law may define this singular optimal control trajectory. The state and costate values at 25 minutes are in table 4.

Table 4: State and costate values at 25 minutes

x_1	0.6422	λ_1	-0.5882
x_2	0.0772	λ_2	0.0012
x_3	1.0244	λ_3	-0.3619
x_4	-0.0622	λ_4	1.0000
x_5	0.0149	λ_5	0.7478

Step 3: There was no need to make the optimal control problem control-affine because the problem is already control-affine. The problem was reformulated as a Hamiltonian non-linear system according to equations 5 to 10. This system is not shown here.

Step 4: The relative degree of this system at 25 minutes is equal to 2 and the C-matrix is non-singular in the time interval 25 tot 200 minutes. This means that a static state feedback law based on a relative degree of 2 is applicable for this singular optimal control trajectory interval.

Step 5 and 6: The static state feedback law was calculated and applied to calculate singular trajectories of the states, costates, input and output in three simulated closed-loop optimal control experiments:

a. To meet the first validation objective, an experiment was carried out in which state trajectories were simulated and offered as artificial observed state trajectories to the static state feedback law for the closed-loop calculation of state, costate, input and output trajectories. These trajectories are plotted with solid lines (—) in figures 1 to 4.
b. For the second validation objective experiment a was repeated but in this experiment the state trajectories were perturbed deliberately by a 20%-increase of the third state at 100 minutes, thereby simulating an unforeseen change in a state value. Closed-loop calculated state, costate, input and output trajectories of this experiment are plotted with solid lines (—) in figures 5 to 8.
c. To further substantiate the second validation objective, experiment b was repeated, but now the singular optimal control trajectory calculated in open loop was applied instead of controlling the system by the static state feedback law. These trajectories are plotted with solid lines marked with dots (—•—) in figures 5 to 8.

7. RESULTS

Figures 1 to 4 show a good resemblance between the state, costate, output and input trajectories calculated in open loop and those obtained with the static state feedback law.

Figures 5 to 8 show that in experiment b the static state feedback law changes the singular optimal control trajectory when there is a 20%-increase of the third state at t=100, which leads to changes in state and costate trajectories. The output, being *dH/du*, correctly returns to its setpoint value zero (Figure 7). The application of the open-loop calculated optimal control trajectory in experiment c leads to an output deviation from its setpoint after 100 minutes, which means that the state, costate and input trajectories are not optimal. This is confirmed by the final value of the goal function, which is the fourth state here. Its value is higher than the one in experiment b.

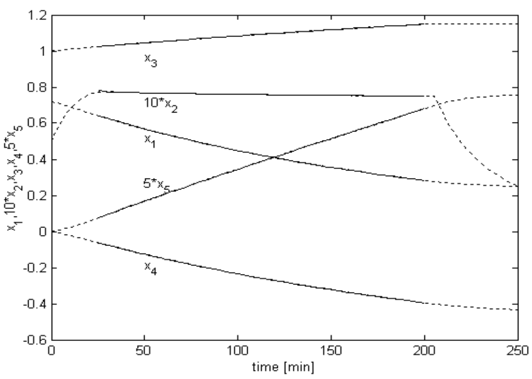

Fig. 1 Optimal state trajectories calculated open-loop (---) and closed-loop in experiment a (—).

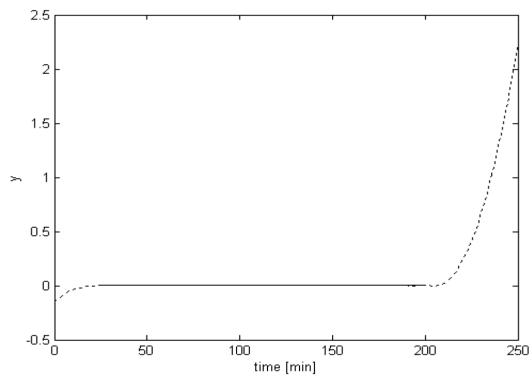

Fig. 3 Optimal output trajectories calculated open-loop (---) and closed-loop in experiment a (—).

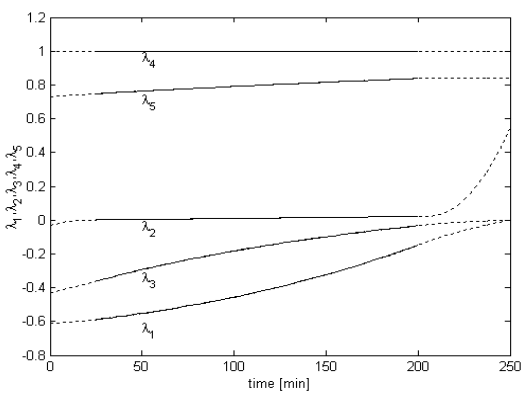

Fig. 2 Optimal costate trajectories calculated open-loop (---) and closed-loop in experiment a (—).

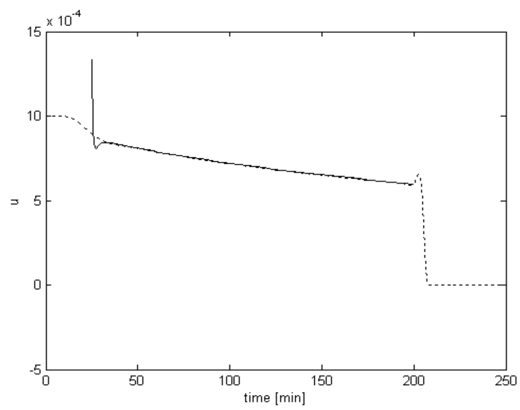

Fig. 4 Optimal control trajectories calculated open-loop (---) and closed-loop in experiment a (—).

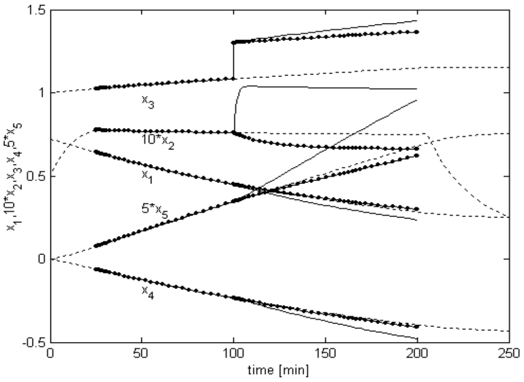

Fig. 5 Optimal state trajectories calculated open-loop (---) and closed-loop in experiments b (—) and c (—●—).

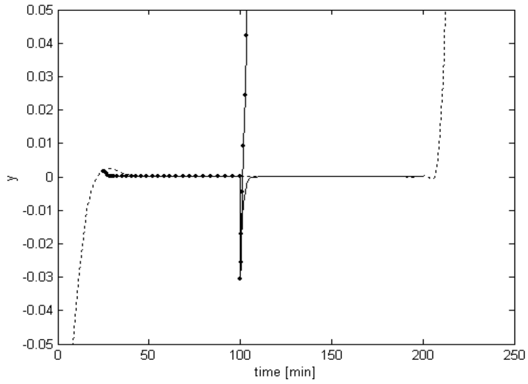

Fig. 7 Optimal output trajectories calculated open-loop (---) and closed-loop in experiments b (—) and c (—●—).

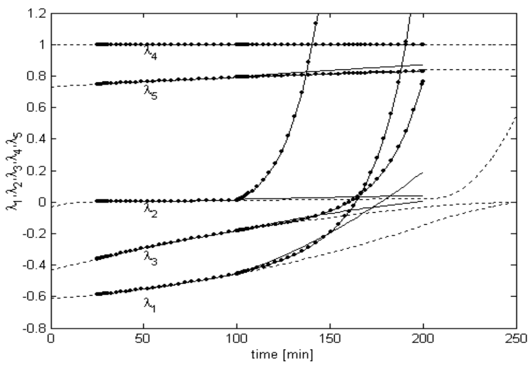

Fig. 6 Optimal costate trajectories calculated open-loop (---) and closed-loop in experiments b (—) and c (—●—).

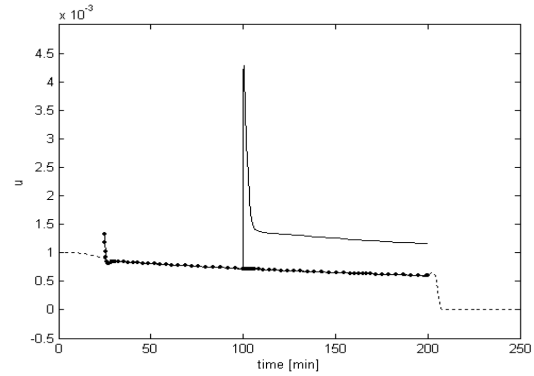

Fig. 8 Optimal control trajectories calculated open-loop (---) and closed-loop in experiments b (—) and c (—●—).

8. CONCLUSION

A numerical development of static state feedback laws for singular optimal control trajectories using MAFC is presented in this paper. An example demonstrates that singular trajectories generated by a numerical static state feedback law are comparable to singular trajectories generated open-loop by a gradient method. It also demonstrates that a numerical static state feedback law is able to adjust the singular optimal control trajectories in response to unforeseen changes in state values.

REFERENCES

Bequette, B. W. (1991). Nonlinear control of chemical processes: a review. *Industrial and engineering chemistry research,* **30,** 1391-1413.

Bryson, A. E. (1999). *Dynamic optimization*, Addison-Wesley Longman, Inc., Menlo Park.

de Graaf, S. C. (2004). Test of ACW-gradient optimisation algorithm in computation of an optimal control policy for achieving acceptable nitrate concentration of greenhouse lettuce. *Mathematics and computers in simulation,* 117-126.

Isidori, A. (1989). *Nonlinear control systems; an introduction*, Springer-Verlag, Berlin.

Kravaris, C., et al. (1997). *Nonlinear model-based control of nonminimum-phase processes*, Kluwer academic publishers, Dordrecht.

Magana Jimenez, Q. (2002). *Nonlinear control via automatic differentiation*, Case western reserve univeristy, Cleveland.

Nijmeijer, H. and v. d. A. J. Schaft (1990). *Nonlinear dynamical control systems*, Springer-Verlag New York Inc., New York.

Palanki, S., C. Kravaris and H. Y. Wang (1993). Synthesis of state feedback laws for end-point optimization in batch processes. *Chemical Engineering Science,* **48,** 135-152.

Rahman, S. and S. Palanki (1996). On-line optimization of batch processes in the presence of measurable disturbances. *American Institute of chemical engineering journal,* **42,** 2869-2882.

Schaft, v. d. A. J. (1984). *System theoretic description of physical systems*, Centre for mathematics and computer science, Amsterdam.

Srinivasan, B., S. Palanki and D. Bonvin (2000). *A tutorial on the optimization of batch processes: I. Characterization of the optimal solution*, BatchPro, Gregersen, L., Danmarks Tekniske Universitet,

Stengel, R. F. (1994). *Optimal control and estimation*, Dover publications, Inc., New York.