

Hard constraints in control and state variables of multivariable nonlinear processes resolved by elementary nonlinear decoupling†

JENS G. BALCHEN‡ and BJARNE SANDRIB‡

Keywords: *Multivariable control, nonlinear control, nonlinear decoupling, hard constraints*

A model based control strategy for nonlinear multivariable processes having hard constraints on both control and state variables, is derived based upon the concept of Elementary Nonlinear Decoupling (END). In END a *property* space is defined which has the same dimension as the control space and which permits *inverse* control. An optimization procedure is used to design a property transformation that gives the best solution in terms of performance, stability, robustness, etc. This control strategy is shown to be convenient when control and state variables are subject to hard constraints. When constraint limits are reached, new property variables are introduced which approximate the variables to be constrained and thereby achieving fast and accurate control on constraint boundaries.

Introduction

When designing a control strategy for a nonlinear multivariable system it is essential that the problem of possible hard constraints in control and state variables can be handled. In most procedures this problem is solved by employing an optimization technique where the consequences of hard constraints are weighed against measures of performance when the optimal control actions are derived (Polak 1971, Mayne and Polak 1993).

In this contribution another approach is taken. The unconstrained END algorithm (Balchen 1991, Balchen and Sandrib 1995) is applied as long as none of the control and state variables (u and x respectively) have reached their constraints. When, say, one of the state variables (x_i) reaches a constraint (high or low) a new END strategy is introduced which replaces the state variables (x_i) with appropriate property variables (\bar{z}_j). Thereby an END control scheme is developed which keeps the control and state variables within their boundaries until the system again returns to the unconstrained operating conditions. Obviously if any of the control and state variables stay permanently at their constraints, the system is not properly designed and must be reevaluated from the design phase.

Short review of the END algorithm

Consider a general nonlinear multivariable process described by

$$\dot{x} = f(x, u, v) \quad (1)$$

where $x \in R^n$: state vector, $u \in R^r$: control vector, $v \in R^s$: disturbance vector, $f \in R^n$: rate

Received 20 September 1997.

†Reprinted from Preprints IFAC Symposium DYC'D'95, Helsingør, Denmark, with permission from Elsevier Science Ltd., The Boulevard, Langford Lane, Kidlington OX5 1GB, UK.

‡The Norwegian Institute of Technology, Department of Engineering Cybernetics, 7034 Trondheim, Norway.

function vector. In addition to the model (1), a model must be defined describing the measurements that are extracted from the process forming the basis for estimating the process states

$$y = h(x) + w \quad (2)$$

where $y \in R^m$: measurement vector, $h \in R^m$: measurement functions, $w \in R^m$: measurement uncertainty (noise).

For the system modeled above a property vector (z) is defined which describes the quantities (properties) one wants to control to certain values (setpoints). The number of such properties is related to the number of degrees of freedom of control namely r . A mathematical model of the properties is

$$z = d(x) \quad (3)$$

where $z \in R^m$: property vector, $d \in R^m$: property function vector.

Often the property transformation is linear so that

$$z = Dx \quad (4)$$

Nonlinear decoupling algorithms have the purpose of finding the control vector (u) which drives the system in such a way that $\dot{z} = \dot{z}_d$ is a specified property rate of change trajectory. One solution to this problem is known as input-output linearization employing a differential geometry method to make the system invertible (Isidori 1989). In the END algorithm the invertibility problem is resolved by designing the $d(\cdot)$ or D to achieve invertibility, proper dynamic behavior of the inner system and robustness.

In the following (4) will be assumed describing the property transformation. Then from (1) we have

$$\dot{z} = \dot{z}_d = Df(x, u, v) \quad (5)$$

which can be solved by an iterative equation solver of the form

$$\dot{u} = K_u(\cdot)(\dot{z}_d - Df(\cdot)) \quad (6)$$

The matrix $K_u(\cdot)$ of (6) is there to assure convergence of (6) and can be found in many ways. One way is to specify a linearized version of (6) to have constant eigenvalues (Λ) such that

$$\delta \dot{u} = -K_u(\cdot)D \frac{\partial f(\cdot)}{\partial u} \delta u = \Lambda \delta u \quad (7)$$

leading to

$$K_u(\cdot) = -\Lambda \left(D \frac{\partial f(\cdot)}{\partial u} \right)^{-1} \quad (8)$$

A block diagram illustrating the END algorithm is shown in Fig. 1. It includes a nonlinear estimator which has an updating matrix denoted $K_x(\cdot)$ of unspecified form since a number of algorithms may be used. It is seen that an estimate of \dot{z} is derived from the process model in the estimator. The feedback loop containing $K_u(\cdot)$ generates the proper control action (u) which satisfies $\dot{z} = \dot{z}_d$. It is also noted that in Fig. 1 the integrator generating u from \dot{u} is equipped with an *antiwindup* facility (*AW*) to prevent this integrator from overcharging when the control variables go into saturation. Furthermore it is noted that the vector \bar{v} which represents the measured disturbances, gives a direct feedforward into the model. If all dominant disturbances are measured

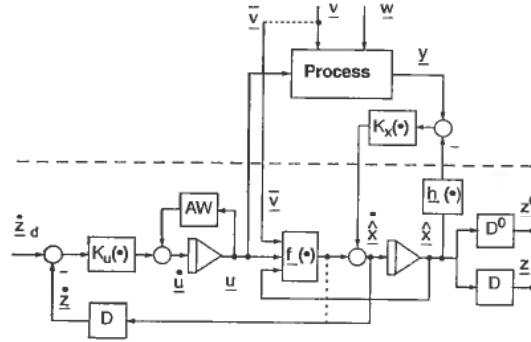


Figure 1. Blockdiagram of END algorithm.

and the model is nearly perfect, the updating signal from the estimator gain matrix ($K_x(\cdot)$) will be small and white. The fewer disturbances that are measured, the more power will be contained in this updating signal.

Assuming for the sake of simplicity that (1) is replaced by its linearized version

$$\delta\dot{x} = A\delta x + B\delta u + C\delta v \quad (9)$$

and assuming that the eigenvalues Λ of the equation solver are very large, it can be shown that the resulting system after the application of END will be described by

$$\dot{x} = (I - B(DB)^{-1}D)(Ax + Cv) + B(DB)^{-1}\dot{z}_d \quad (10)$$

It is obvious from both (8) and (10) that for a solution to exist the matrix

$$D \frac{\delta f(\cdot)}{\delta u} = DB$$

must be nonsingular. The least requirement for this to be true is that $\dim z = \dim u$.

Furthermore it must be required that the dynamic system described by the state equation of (10) has acceptable behavior, i.e. eigenvalues properly located in the left half of the complex plane.

The dynamics of this system is determined by the eigenvalues of the matrix

$$(I - B(DB)^{-1}D)A \quad (11)$$

From (10) the main system transfer matrix is given by

$$x(s) = H(s)\dot{z}_d(s) \quad (12)$$

where

$$H(s) = (sI - (I - B(DB)^{-1}D)A)^{-1}B(DB)^{-1} \quad (13)$$

The transfer matrix of (13) can be characterized by the eigenvalues of (11) or when converted into the frequency response matrix $H(j\omega)$ by the maximal and minimal singular values $\bar{\sigma}_H(j\omega)$ and $\underline{\sigma}_H(j\omega)$. Thereby different methods for designing the dynamic behavior can be derived by choosing the elements of the D matrix so that the system becomes invertible with acceptable dynamic properties.

When choosing the properties to be controlled without concern for invertibility one could arrive at the transformation $z = D^0x$ expressing the *desirable* but *nonrealizable* property vector. The reason for the nonrealizability is that the matrix D^0B most often will be singular or the resulting dynamic properties of (10) are not acceptable.

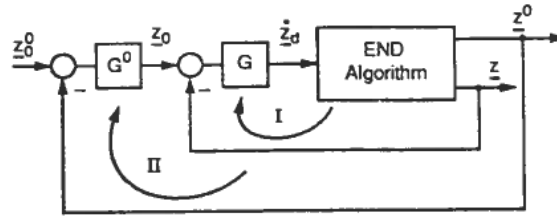


Figure 2. The feedback control around the END algorithm.

An alternative way of designing D is derived by defining the difference

$$\Delta z = z - z^0 = (D - D^0)x = \Delta D x \quad (14)$$

and from that an objective functional

$$\begin{aligned} J &= \int_0^{\infty} (\Delta z^T Q \Delta z + u^T P u) dt \\ &= \int_0^{\infty} (x^T \Delta D^T Q \Delta D x + u^T P u) dt \end{aligned} \quad (15)$$

where the matrices Q and P are chosen to give proper weight on the different quantities. An optimization procedure based upon gradient calculations is developed to find $\min_{\Delta D} J$ where ΔD is represented by its elements $\Delta D = \{\Delta d_{ij}\}$ converted into a vector. By minimizing (15) the dynamic behavior of (10) is included.

The consequence of the End algorithm of Fig. 1 is that the nonlinear multivariable system is replaced by a set of r integrators from \dot{z}_d to z . This very simple process can be controlled by diagonal feedback as shown in Fig. 2. The inner loop (I) contains the control matrix G and has a reference vector z_0 . When there is no saturation in any of the control variables, this loop will have nearly perfect response with very high bandwidth. The outer feedback loop (II) contains a control matrix G^0 , a reference vector z_0^0 and an output z^0 . This outer loop will be slightly degraded relative to loop (I). In some cases the outer loop (II) can be omitted yielding a higher bandwidth performance.

The end algorithm with constrained state variables

The problem arising when control variables (u) are constrained (saturated) has been dealt with in Balchen and Sandrib (1994). In short the solution is that for each combination of saturating control variables $u^{ij\dots}$, where the superscripts i and j indicate which control variables are saturating, specific property transformations $D^{ij\dots}$ are defined expressing the priorities to be employed when the process is to be controlled with fewer degrees of freedom.

Assuming for the sake of simplicity that the control saturation is taken care of by this method, a similar principle is employed to handle *state variable* constraints.

In addition to (1) and (2) the state constrained case has the extra condition

$$c(x) \leq 0 \quad (16)$$

The simplest realization of (16) is

$$x_{imin} \leq x_i \leq x_{imax} \quad (17)$$

where x_i is one of the components of x .

The proposed strategy is to replace the constraints in x by representative constraints in z and labeling these \tilde{z} . In doing this it is observed that the number of state constraints that can be handled at any time is equal to or less than the number of unconstrained (unsaturated) control variables (degrees of freedom). In other words, if there are only two unconstrained control variables available, only two state variables can be controlled to stay on the constraints (maximal or minimal). Thus all the other state variables have to stay inside acceptable operating ranges. The way to achieve this is to design the D matrix with this in mind.

The property transformation to be applied when state variables i and j have reached their constraint values is thus

$$\tilde{z} = \tilde{D}^j x \quad (18)$$

(18) indicates that two of the property variables have to be allotted to the constrained states x_i and x_j . Thus, as an example, if the number of control variables is $r = 3$ and the number of states is $n = 8$ one could have a case with

$$D^0 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (19)$$

indicating that x_1, x_2 and x_3 are to be controlled. The designed property transformation for this case will be of the form

$$D = \begin{bmatrix} 1 & d_{12} & d_{13} & d_{14} & d_{15} & d_{16} & d_{17} & d_{18} \\ d_{21} & 1 & d_{23} & d_{24} & d_{25} & d_{26} & d_{27} & d_{28} \\ d_{31} & d_{32} & 1 & d_{34} & d_{35} & d_{36} & d_{37} & d_{38} \end{bmatrix} \quad (20)$$

Assuming that states x_4 and x_6 are reaching their constraint values simultaneously, the appropriate \tilde{D} matrix could be

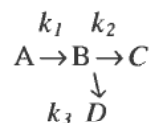
$$\tilde{D}^{46} = \begin{bmatrix} 1 & \tilde{d}_{12} & \tilde{d}_{13} & \tilde{d}_{14} & \tilde{d}_{15} & \tilde{d}_{16} & \tilde{d}_{17} & \tilde{d}_{18} \\ \tilde{d}_{21} & \tilde{d}_{22} & \tilde{d}_{23} & 1 & \tilde{d}_{25} & \tilde{d}_{26} & \tilde{d}_{27} & \tilde{d}_{28} \\ \tilde{d}_{31} & \tilde{d}_{32} & \tilde{d}_{33} & \tilde{d}_{34} & \tilde{d}_{35} & 1 & \tilde{d}_{37} & \tilde{d}_{38} \end{bmatrix} \quad (21)$$

This \tilde{D}^{46} matrix indicates that the original emphasis on x_1 has been kept while x_2 and x_3 have been deemphasized in order to be able to handle the constraints in x_4 and x_6 . The elements \tilde{d}_{ij} introduced in \tilde{D}^{46} are mostly very small. Some of them will probably be so small that they can be neglected and made equal to zero. When none of the state variables reach their constraint values, the property transformation will return to that of (20) which has been optimized for the unconstrained case. The design of the matrix $\tilde{D}^{ij\dots}$ is done offline.

Further discussion of the algorithm is conveniently done in relation to a concrete example.

End control of a stirred tank reactor with two consecutive reactions

Consider a chemical reactor with two consecutive and parallel reactions



where k_i are the reaction rate constants for the three reactions which have reaction rates

$$r_B = k_1 c_A, r_C = k_2 c_B \text{ and } r_D = k_3 c_B$$

The reaction rate constants are temperature dependent

$$k_i = k_{i0} \exp\left(-\frac{E_i}{R\theta}\right) \quad (22)$$

where k_{i0} , E_i and R are constants and θ : reactor temperature.

Four differential equations describe the mass balances in terms of the concentrations ($c_A \dots c_D$) of each component:

$$\dot{c}_A = (-c_A + c_{A0})\frac{q}{V} - r_B \quad (23)$$

$$\dot{c}_B = (-c_B + c_{B0})\frac{q}{V} + r_B - r_C - r_D \quad (24)$$

$$\dot{c}_C = (-c_C + c_{C0})\frac{q}{V} + r_C \quad (25)$$

$$\dot{c}_D = (-c_D + c_{D0})\frac{q}{V} + r_D \quad (26)$$

The differential equation describing the energy balance is

$$\dot{\theta} = (-\theta + \theta_0)\frac{q}{V} + \frac{1}{\rho c V}(-\Delta H_1 V r_B - \Delta H_2 V r_C - \Delta H_3 V r_D + Q) \quad (27)$$

where ρ , c , V , ΔH_i are constants, q : feed flow, Q : heat flow.

In summary the state, control and disturbance vectors are:

$$x = [c_A, c_B, c_C, c_D, \theta]^T \quad (28)$$

$$u = [q, Q]^T \quad (29)$$

$$v = [c_{A0}, c_{B0}, c_{C0}, c_{D0}, \theta_0]^T \quad (30)$$

The following numerical values of model parameters are used:

$$k_{10} = 1.0 \times 10^{15}, k_{20} = 4.5 \times 10^{15}, k_{30} = 1.0 \times 10^{14}, E_1 = E_2 = E_3 = 1.3 \times 10^5, R = 8.3, \\ \rho = 1, c = 4.2 \times 10^3, V = 10^3, \Delta H_1 = -5 \times 10^4, \Delta H_2 = -5.5 \times 10^4, \Delta H_3 = -5 \times 10^3 \\ u_{nom} = [0.513, 70.188 \times 10^3]^T, v_{nom} = [1, 0, 0, 0, 350]^T.$$

Since the process has two degrees of freedom, we define a desirable property transformation

$$D^0 = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (31)$$

which means that we want to control the concentration of the desirable product B and the temperature θ . Differentiation of (23)–(26) yields

$$B = \frac{\partial f(\cdot)}{\partial u} = \begin{bmatrix} 1/V(-x_1 + v_1), & 0 \\ 1/V(-x_2 + v_2), & 0 \\ 1/V(-x_3 + v_3), & 0 \\ 1/V(-x_4 + v_4), & 0 \\ 1/V(-x_5 + v_5), & 1/\rho c V \end{bmatrix}$$

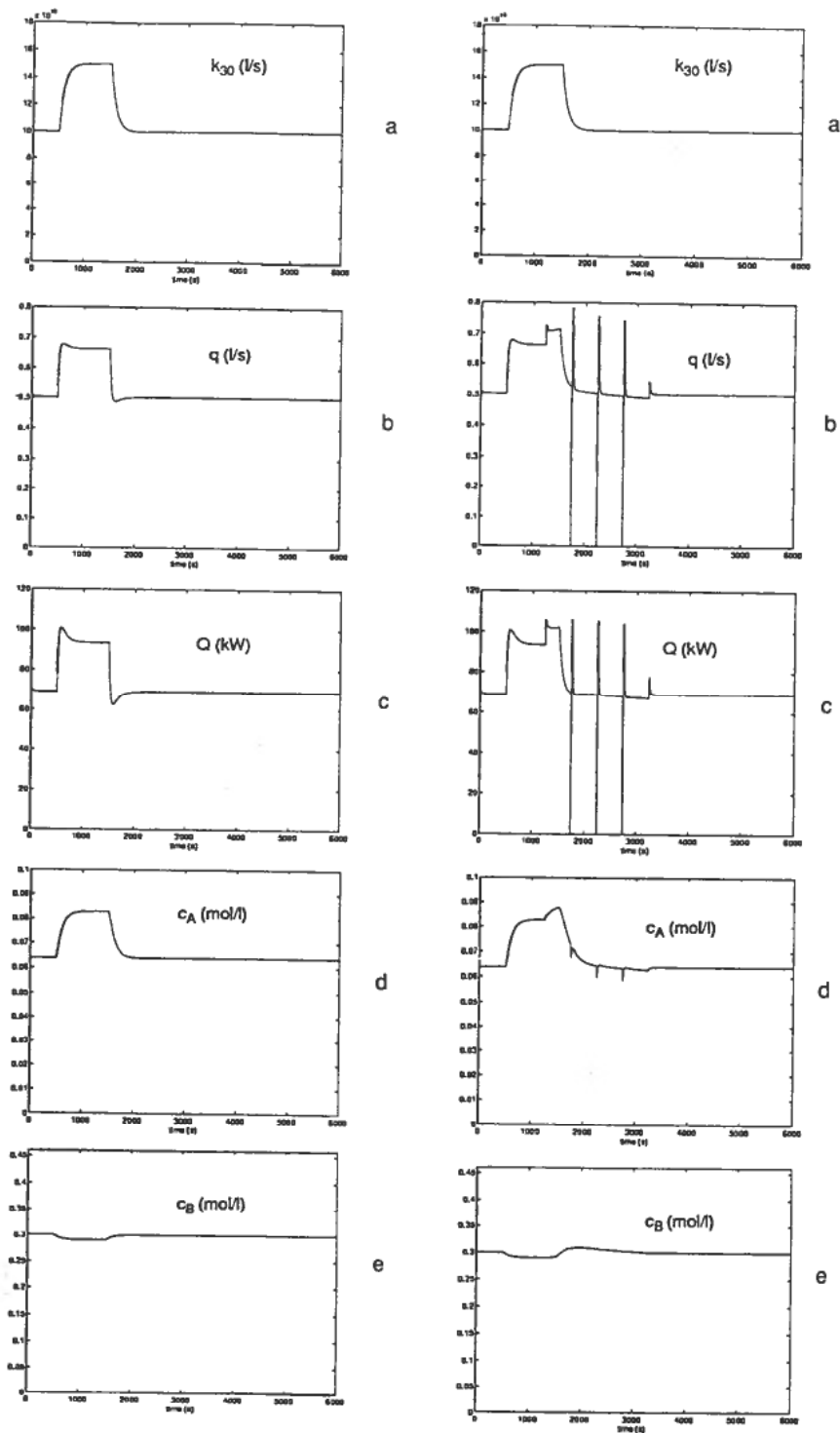


Figure 3a-e. Responses of unconstrained END controlled system when k_{30} has 50% change.

Figure 4a-e. Responses of state constrained END controlled system when k_{30} has 50% change.

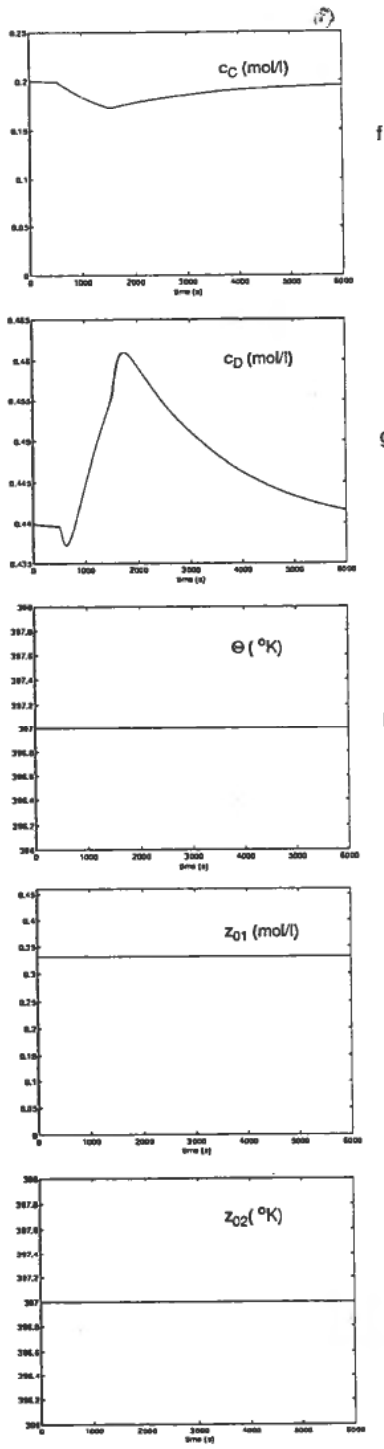


Figure 3f-j. Responses of unconstrained END controlled system when k_{30} has 50% change.

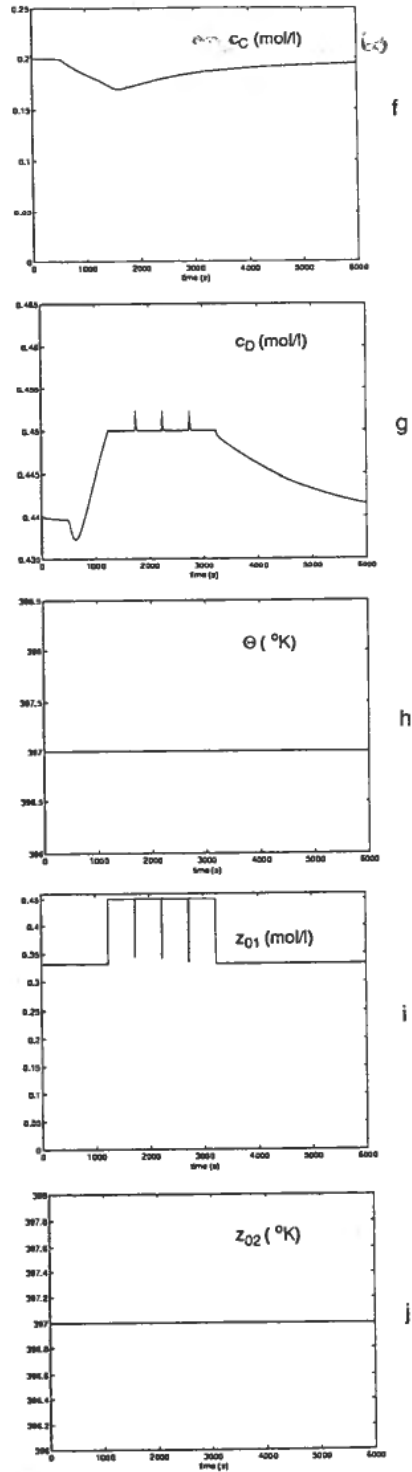


Figure 4f-j. Responses of state constrained END controlled system when k_{30} has 50% change.

Even though the matrix

$$\left(D^0 \frac{\partial f(\cdot)}{\partial u} \right)$$

is nonsingular another matrix D must be designed because the dynamics of the inner system (10) becomes unacceptable.

An adequate matrix D which secures stability of the inner system will be

$$D = \begin{bmatrix} d_{11} & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (32)$$

where $d_{11} = 0.5$. An optimization of the other elements of D could have given some improvement of the performance, but that is not important here. In addition to the END algorithm of Fig. 1 the system is controlled by an inner loop

$$\text{(loop I as shown in Fig. 2 with } G = \begin{bmatrix} 0.05 & 0 \\ 0 & 0.05 \end{bmatrix} \text{)}.$$

The responses of this system are shown in Fig. 3a–j where one of the system parameters k_{30} is changed from 1.0×10^{14} to 1.5×10^{14} and back again.

As can be seen from Fig. 3g the concentration c_D is first increasing and then decreasing again. If the component D is an undesirable byproduct and we want c_D to be less than 0.45 we may design a new property transformation.

$$\tilde{D}^A = \begin{bmatrix} \tilde{d}_{11} & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (33)$$

which replaces that of (32) when $c_D > 0.45$. At selected intervals short tests are made to see if the system can be returned to the unconstrained transformation (32). In Fig. 4g that happens after the fourth test when c_D again becomes less than 0.45. The results of this strategy are shown in Fig. 4. It is clearly seen that c_D is controlled on its constraint with only small overshoots.

Conclusion

Elementary Nonlinear Decoupling (END) can conveniently be modified to handle hard constraints in states by replacing these states by new properties to be controlled.

ACKNOWLEDGEMENTS

This research has been supported by SINTEF (JGB) and STATOIL (BS).

REFERENCES

- BALCHEN, J. G. (1993). Design of the property transformation in elementary nonlinear decoupling of multivariable processes. *Modeling, Identification and Control*, **14**, 219–227.
- BALCHEN, J. G. and SANDRIB, B. (1994). Input saturation in nonlinear multivariable processes resolved by nonlinear decoupling. *3rd IEEE Conference on Control Applications*, August 24–26, 1994, Glasgow, Scotland, UK.
- BALCHEN, J. G. and SANDRIB, B. (1995). Elementary non-linear decoupling control of composition in binary distillation columns. *J. Proc. Cont.*, **5**, No. 4, pp. 241–247 (Elsevier Science Ltd).
- ISIDORI, A. (1989). *Nonlinear Control Systems* (2nd ed.) (Springer Verlag, Berlin).
- MAYNE, D. Q. and POLAK, E. (1993). Optimization based design and control. *12th IFAC World Congress*, Sydney, Australia, pp. 129–138.
- POLAK, E. (1971). *Computational Methods in Optimization* (Academic Press, New York).