INCLUSION OF ACTUATOR SATURATION AS COMPLEMENTARITY CONSTRAINTS IN INTEGRATED DESIGN AND CONTROL

Rhoda Baker and Christopher L. E. Swartz

Department of Chemical Engineering, McMaster University, Hamilton, Ontario, L8S 4L7, Canada

Abstract: The design of a process can significantly affect its ability to be satisfactorily controlled, which motivates the need to design the process and its associated control system simultaneously. However, since all physical systems have constraints, it is necessary to include these within the model as well. A bilinear formulation for rigorous saturation handling is able to correctly describe the logical conditions which accompany saturation of the manipulated variable and is suitable for incorporation within a simultaneous optimization framework. This paper explores a number of issues. First, a bilinear formulation is used to examine the effect of including anti-reset windup within an integrated design. Second, the performance of a solver specifically tailored to solve optimization problems with complementarity constraints is compared to other nonlinear solvers for the solution of a simple nonlinear case study. Finally, its performance on a more complex nonlinear CSTR case study is discussed.

Keywords: saturation, anti-reset windup, complementarity, integrated design

1. INTRODUCTION

It has long been recognized that the design of a process can significantly affect its inherent ability to be controlled using feedback. This has motivated the need for techniques which allow the control system to be designed in conjunction with the process design. One means of achieving this is through an optimization-based approach, which allows the simultaneous consideration of both the steady state economics as well as the dynamic operation of the process under feedback control.

Applying such optimization techniques requires a mathematical model of the process as well as its associated control system for inclusion within an optimization problem. However, all physical systems are subject to constraints and, in order to model processes correctly, these constraints should be included within the model formulation as well. These include, in particular, constraints on manipulated inputs where a valve, for example, cannot be more than 100% open. However, when creating a model of a process for the purpose of optimization based integrated control and design, merely including these limits as upper or lower bounds may be insufficient to correctly describe the behavior of the actual system.

In particular, when dealing with a control system, a disturbance of sufficiently large magnitude may in reality cause the manipulated variable to saturate, whereas the above constraint specification would merely allow the modelled manipulated variable to touch the bound and then move away as illustrated in Figure 1. This is because input saturation essentially corresponds to a set of logical conditions, and requires special formulation in order to be included within a conventional optimization framework. A bilinear or mixed-integer formulation for rigorous saturation handling can be used to correctly describe this phenomenon (Young *et al.*, 2004).

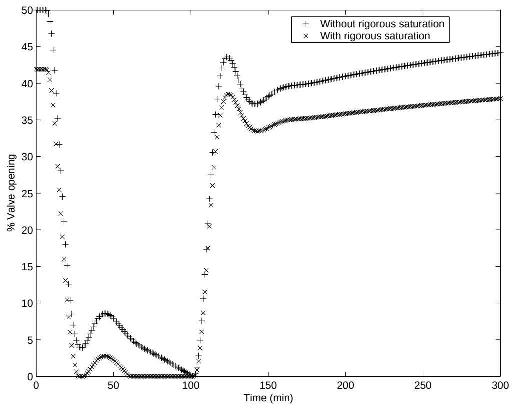

Fig. 1. The effect of including rigorous saturation handling in an integrated design and control optimization framework.

Briefly, in the case of the bilinear formulation, the inclusion of the following constraints in an integrated control and design problem will correctly account for the difference between the actuator, u_a, and the controller output generated by the control law, u_c, by means of the slack variables S_L and S_U at time step k. The complementary terms (1b) and (1c) ensure that a discrepancy occurs only when the actuator is at either the lower or upper bound, u_L and u_U respectively, and that the rest of the time u_a is equal to u_c.

$$u_c(k) = u_a(k) - S_L(k) + S_U(k) \tag{1a}$$
$$0 = S_L(k)(u_a(k) - u_L) \tag{1b}$$
$$0 = S_U(k)(u_a(k) - u_U) \tag{1c}$$
$$u_L \leq u_a(k) \leq u_U \tag{1d}$$
$$0 \leq S_L(k) \tag{1e}$$
$$0 \leq S_U(k) \tag{1f}$$

Recent work shows that failure to handle the actuator constraints rigorously within an integrated design and control optimization framework can lead to a suboptimal design (Baker, 2000).

On a related note, depending on the implementation of the controller, saturation of the manipulated variable can lead to the phenomenon of reset windup. Reset windup occurs in controllers with reset or integral action when the integral mode continues to integrate the error when it does not affect the output. In the case of input saturation, this occurs because the manipulated variable is unable to respond to changes in the controller output signal because it is at a bound. However, reset windup can also occur in processes with selective control systems, which have a manipulated variable that can be controlled by one of many controllers at different times.

There exist a number of approaches for dealing with reset windup. Simply limiting the controller output signal does not eliminate reset windup because there is still a discrepancy between the calculated controller output and the controller output as seen by the process. One approach is to freeze the integral term if windup is detected, and reactivate it once the cause for windup has been removed. Alternatively, if the digital control algorithm is implemented using the velocity form of the controller, then if the calculated value of the input change is added to the actual value of the manipulated variable, no windup will occur (Khandheria and Luyben, 1976).

In this paper we specifically focus on the effects of including anti-reset windup measures within an integrated control and design problem in processes that experience input saturation.

As mentioned earlier, previous work had considered two formulations for modelling input saturation within an optimization framework; namely the bilinear and mixed-integer formulations. When the bilinear problem was solved using a standard nonlinear solver it would occasionally find a suboptimal solution, or sometimes not find a solution at all depending on the initial starting guess.

The inclusion of the complementarity constraints of the bilinear formulation within the optimization framework leads to a problem within the class of mathematical programs with equilibrium constraints (MPECs). The general form of the MPEC with complementarity constraints is:

$$\begin{array}{ll} \min\limits_{x,y,z} & f(x,y,z) \\ \text{s.t.} & g(x,y,z) \geq 0 \\ & c(x,y,z) = 0 \\ & 0 \leq F(x,y,z) \perp y \geq 0 \end{array} \tag{2}$$

where $x \in \Re^n$, $y \in \Re^m$, $z \in \Re^l$, $f : \Re^{n+m+l} \rightarrow \Re$ is the objective function, and the constraints are $g : \Re^{n+m+l} \rightarrow \Re^p$, $c : \Re^{n+m+l} \rightarrow \Re^q$ and $F : \Re^{n+m+l} \rightarrow \Re^m$. The last constraint can be interpreted as an "either/or" condition which can be rewritten as the following set of complementarity constraints:

$$\left.\begin{array}{r} F_i(x,y,z)\, y_i = 0 \\ F_i(x,y,z), y_i \geq 0 \end{array}\right\} \quad \forall \quad i = 1, ..., m \tag{3}$$

Using simple examples, it was shown that these problems may be nonconvex or have feasible solution sets that may be discontinuous or non-closed (Luo *et al.*, 1997), which might explain some of the symptoms described previously. Unfortunately, this class of problems has only recently come under scrutiny and solvers for MPECs are consequently not widely available. Examples of algorithms and software developed to specifically solve MPECs are the Penalty Interior Point Algorithm (Luo *et al.*, 1997), and IPOPT, which is a general interior point based NLP solver that has recently been modified to solve MPECs (Raghunathan and Biegler, 2003). There is also work which provides encouraging results indicating that certain general NLP solvers are able to perform well in comparison to specialized algorithms when solving MPECs (Fletcher and Leyffer, 2002).

The approach IPOPT uses in its MPEC solving mode differs from the approach a conventional NLP solver would take in that it handles the complementarity constraints differently from the manner in which it deals with conventional equality and inequality constraints. In particular, the algorithm relaxes the complementarity constraints by adding a nonnegative slack variable which is then also included in the log barrier function and the entire problem solved using an interior point method (Raghunathan and Biegler, 2003).

In this study, the IPOPT algorithm is applied to the optimization-based integrated design and control of three mixing tanks in series as well as a stirred tank reactor, both with rigorous input saturation handling provided by including the bilinear formulation for saturation handling. There is also a brief comparison of the performance of the IPOPT solver against general nonlinear solvers for a problem of this type.

2. CASE STUDY I

The system under consideration is a set of three mixing tanks in series depicted in Figure 2 (Marlin, 2000). The controlled variable is the percentage of component A exiting the final tank, while the manipulated variable is the percentage valve opening for the flow of stream A. The disturbance entering the system is a change in the concentration of A in stream B. The aim of the control system is to maintain the concentration of A in the final tank as close to the set point as possible in the face of disturbances and set point changes and to this end, the objective used in this case is the integral absolute error.

The transfer functions relating the concentration of A exiting the final tank to the valve position and to the disturbance are respectively:

$$G_p(s) = \frac{0.039}{(5.0s+1)^3}$$

$$G_d(s) = \frac{1.0}{(5.0s+1)^3}.$$

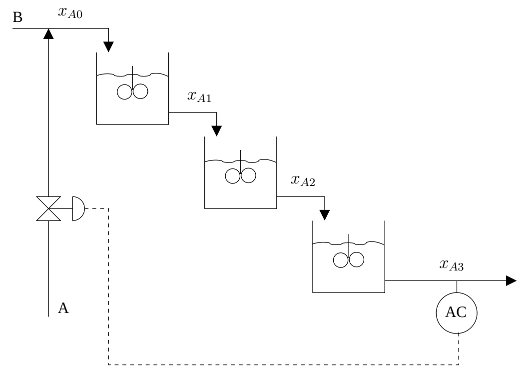

Fig. 2. Schematic of 3 mixing tanks in series.

The disturbance being rejected in this case study is a pulse of 4% in the concentration of A in feed stream B of duration $t = 5$ min to $t = 100$ min. The size of the disturbance increase is such that the actuator has to saturate in order to reject the disturbance. The set point for the outlet concentration of A, and therefore the initial percentage valve opening, as well as the controller reset time are fixed.

Under investigation is the effect that the inclusion of anti-reset windup has on the objective function and on the design of the controller.

The process was modelled in AMPL by means of a finite impulse response model with the saturation logic handled by including the set of equations (1).

In the first example the controller parameters are fixed, and since there are no other design degrees of freedom, the optimization problem reduces essentially to a simulation. This will demonstrate that the bilinear formulation is able to model mathematically the logical conditions associated with input saturation. It will also demonstrate the effect of including anti-reset windup within an optimization-based integrated control and design problem.

The results for the two cases with and without anti-reset windup are presented in Table 1. The resulting trajectories are displayed in Figure 3.

Table 1. Results of scenarios with and without anti-reset windup.

	Objective
Without anti-reset windup	369.68
With anti-reset windup	233.78

As expected, the example without anti-reset windup in the controller has a larger integral absolute error. This demonstrates worse control performance in comparison to the case with anti-reset windup for these particular parameters.

In the second example, the velocity form of the digital controller will be used for both scenarios:

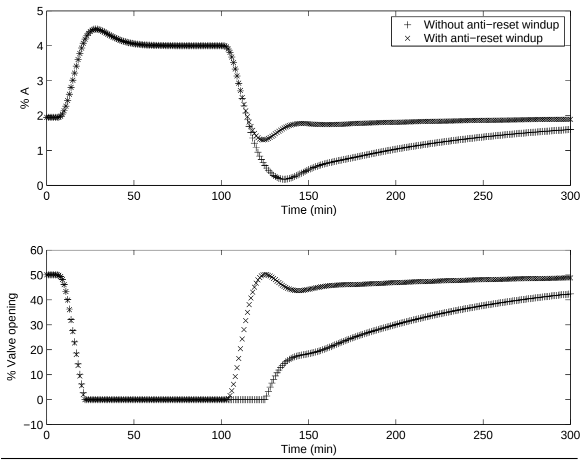

Fig. 3. Results of scenarios with and without anti-reset windup, control parameters fixed.

$$\Delta u_c\left(i\right) = K_c\left[e\left(i\right) - e\left(i-1\right) + \frac{\Delta t}{\tau_I}e\left(i\right)\right] \quad (4)$$

The key difference between the two scenarios is that, in the first, the change in the controller output is calculated based on the previous predicted controller position, which may differ from the actual value entering the process.

$$\Delta u_c\left(i\right) = u_c\left(i\right) - u_c\left(i-1\right) \quad (5)$$

In the second case, reset windup is handled correctly and the true actuator value is used instead:

$$\Delta u_c\left(i\right) = u_c\left(i\right) - u_a\left(i-1\right) \quad (6)$$

In this example, the gain is allowed to vary and we compare the results of a controller with and without correct anti-reset windup. It is required that the optimization scheme choose the best gain to reject this particular disturbance and so the controller gain will be included as a design variable within this problem.

The results of the two formulations are reported in Table 2. The case without correct anti-reset windup has a larger integral absolute error in comparison to the case where the correct actuator position was fed back to the controller. It can also be seen in Figure 4 that reset windup occurs in the case without anti-reset windup and results in poor control performance, since there is a delay before the manipulated variable reacts to the termination of the disturbance pulse at t = 100 min. The optimal gain obtained for equation (5) is also smaller than that for equation (6) because a larger gain would exacerbate the effect of the reset windup. It can also be seen that at the end of the time horizon the controlled variable trajectory has still not achieved steady state at the set point in the case of equation (5), because the controller gain is less aggressive.

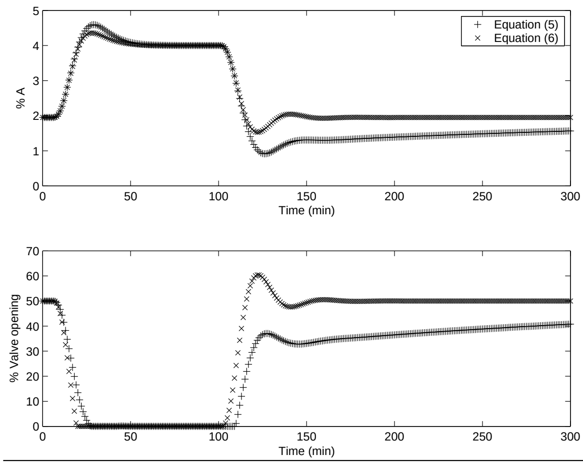

Fig. 4. Results using equations (5) and (6) with the gain allowed to vary.

Table 2. Results using equations (5) and (6) with the gain allowed to vary.

	Objective	Kc
Equation 5	310.25	17.29
Equation 6	206.99	25.69

3. COMPARISON OF IPOPT AND NONLINEAR SOLVERS

As mentioned previously, IPOPT/MPEC uses a different approach to solving problems with complementarity constraints than the conventional general nonlinear solver. In this section, the reliability of the IPOPT/MPEC solver is compared to IPOPT as a nonlinear solver (IPOPT/NLP), MINOS and CONOPT2 running under GAMS. The case study in question is the three mixing tanks in series example from the previous section with anti-reset windup provided by equation (6).

The nonlinearity of the case study arises from two sources: the complementarity constraints of the bilinear formulation for rigorous saturation handling and the fact that the controller gain is included as a decision variable.

The methodology of the comparison is as follows. The results for the case study of the previous example are taken as the base case. From the base case, the values of all the variables are extracted. Uniformly distributed random factors are then added to these nominal values and the resulting values are provided as an initial guesses to each of the solvers. Thus each solver uses the same initial guess. Each random factor is chosen from a “ball” of radius α around the nominal value of the variable. This was then repeated ten times for each particular value of α.

The IPOPT solver was compiled with the MA27, MA28, and the MA47 routines from the Harwell Science Library. The performance of the augmented Lagrangian line search was found to be superior to that of the default filter method, and therefore used in all the studies. All of the solvers were limited to a maximum of 1000 iterations.

It should be noted that since the case study under investigation is a nonlinear problem for which the convexity has not yet been characterized, it is possible that the resulting solution is not the global optimum. For the sake of notational brevity however, we will refer to the lowest known feasible solution as the "optimal" solution.

The solvers were compared based on whether they found the optimal solution, converged to a point that was not the optimal value, or failed and were unable to to converge to a solution within a given number of iterations. This was done for three different values of α (0.5, 1 and 10) and the combined results are reported in Table 3.

Table 3. Summary of results of comparitive study

	Failure	Suboptimal	Optimum
IPOPT/MPEC	6.7%	0%	93.3%
IPOPT/NLP	100%	0%	0 %
MINOS	90%	10 %	0 %
CONOPT2	6.7%	0%	93.3%

The IPOPT/NLP solver performed the least satisfactorily, although it is possible that a different set of algorithm parameters might have resulted in better performance. MINOS also performs fairly poorly. IPOPT/MPEC and CONOPT2 were the only solvers that found the known optimum. For some initial guesses IPOPT would result in line search errors terminating the solver prematurely. At other times the objective function would come fairly close to the known optimum, but the solver would require over a 1000 iterations. Once again, it should be borne in mind that a different set of algorithmic parameters might have yielded different results.

4. CASE STUDY II

The following case study is an example of the integrated design and control of a single continuous stirred tank reactor. The first-principles model described in (Schweiger and Floudas, 1998) was used.

Briefly, the reaction taking place is a first order, exothermic irreversible reaction ($A \rightarrow B$). The controlled variable is the temperature in the tank, while the flow rate of water through the cooling jacket is the manipulated variable. The disturbance is a step increase of 18.3 ^{o}C in the temperature of the feed to the tank. The design variables of this example are: the height and diameter of the tank; the nominal operating temperature in the CSTR and the reactor jacket; the nominal concentration of reactant A exiting the tank; and the gain and reset time of the associated controller. The disturbance for this CSTR example is such that the system exhibits saturation behavior when the input saturation model formulations are included in the integrated design and control optimization framework, and the lower and upper limits of the cooling water flow rate were set as 0 and 12.88 m^3/h respectively. The objective function for the problem is the total cost of the process operating over 4 years, which includes the capital cost of the CSTR as well as the utility cost or cost of operation over the period. The cost of off-spec product is included as part of the objective function by means of a penalty on the integral square error of the concentration of reactant A in the tank. Furthermore, the anti-reset windup formulation described in equation (6) is used.

Table 4. Results of CSTR integrated control and design case study

	No saturation	Saturation
Total Cost, $	572.7×10^3	502×10^3
Capital Cost, $	559.5×10^3	488.7×10^3
Diameter, m	8.57	7.97
Height, m	4.29	3.98
Controller gain	-68.74	-100
Reset time	2	1.83

Previous work attempted to solve this problem using the mixed-integer mathematical formulation of input saturation, but found that the time to solve the optimization problem with currently available MINLP solvers could become prohibitive as the number of variables increased (Baker, 2000). For this reason, the interior point based solver IPOPT/MPEC will be applied to the CSTR example instead to assess whether it is able to successfully solve the optimization problem within a reasonable amount of time.

IPOPT/MPEC was successfully able to solve the problem with results shown in Table 4 and Figure 5. It can also be seen that the case in which the system is allowed saturate has a lower cost than a comparative case in which the actuator behavior is assumed to be strictly linear.

Using results for the system without saturation handling as an initial guess, IPOPT/MPEC was able to solve the rigorous saturation problem in around 1 minute of CPU time, converging in 255 iterations.

5. CONCLUSION

This paper has discussed the use of complementarity constraints to describe actuator saturation effects in integrated design and control type problems. Neglecting to include anti-reset windup was shown to lead to a suboptimal design. In addition, a comparison of the IPOPT/MPEC

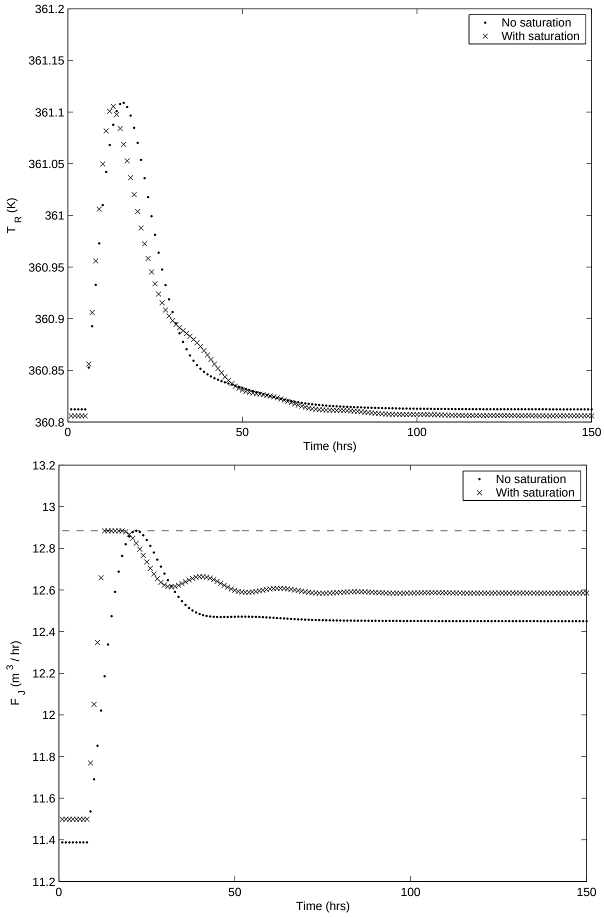

Fig. 5. Rigorous saturation handling for a CSTR.

solver against certain general nonlinear solvers on a problem with rigorous saturation handling was conducted, and showed IPOPT/MPEC and CONOPT2 to have superior performance. Finally, IPOPT/MPEC was successfully used to solve a large scale nonlinear integrated design and control problem with rigorous saturation handling.

REFERENCES

Baker, R. (2000). Inclusion of input saturation in the design of dynamically operable plants. Master's thesis. University of Cape Town.

Fletcher, R. and S. Leyffer (2002). Numerical experience with solving MPECs as NLPs. Technical report. Department of Mathematics, University fo Dundee, Dundee, UK.

Khandheria, J. and W.L. Luyben (1976). Experimental Evaluation of Digital Algorithms for Antireset Windup. *Industrial and Engineering Chemistry, Process Design and Development* **15**(2), 278–285.

Luo, Z.-Q., J.S. Pang and D. Ralph (1997). *Mathematical Programs with Equilibrium Constraints.* Cambridge University Press.

Marlin, T.E. (2000). *Process Control: Designing Processes and Control Systems for Dynamic Performance.* 2nd ed.. McGraw-Hill.

Raghunathan, A.U. and L.T. Biegler (2003). Mathematical Programs with Equlibrium Constraints (MPECs) in Process Engineering. *Computers and Chemical Engineering* **27**(10), 1381–1392.

Schweiger, C.A. and C.A. Floudas (1998). Interaction of design and control: Optimization with dynamic models. In: *Optimal Control: Theory, Algorithms, and Applications* (W.W. Hager and P.M. Pardalos, Eds.). pp. 388–435. Kluwer Academic Publishers.

Young, J.C.C., R. Baker and C.L.E. Swartz (2004). Input Saturation Effects in Optmizing Control - Inclusion within a Simultaneous Optimization Framework. *Computers and Chemical Engineering, in press.*