Proceedings of the 9th International Symposium on
Dynamics and Control of Process Systems (DYCOPS 2010),
Leuven, Belgium, July 5-7, 2010
Mayuresh Kothare, Moses Tade, Alain Vande Wouwer, Ilse Smets (Eds.)

A Performance Study of a Novel Dynamic Real-Time Optimisation Engine Applied to an Industrial Continuous Pulping System

Pablo A Rolandi*. José A Romagnoli**

** Process Systems Enterprise Ltd, London, UK*
(e-mail: p.rolandi@psenterprise.com).
***Luisiana State University, Luisiana, USA*
(e-mail: jose@lsu.edu).

Abstract: In this work we present a flexible real-time dynamic optimisation engine that successfully decouples the controller design and innovation space into three orthogonal axes given by the model formulation, the problem formulation and the solution methods. A simulation of an industrial continuous pulping system is used to run several performance studies with an emphasis on the effect of different model formulations in various production transition scenarios.

Keywords: nonlinear control, optimal control, model-based control, real-time optimisation

1. INTRODUCTION

Model Predictive Control (MPC) and Real Time Optimisation (RTO) are two techniques for Advanced Process Control (APC) that use a process model for optimisation and control of industrial processes. Traditionally, research in these two technologies has focussed on exploring different model formulation alternatives (Henson, 1998). Within this design space, one research axis represents the division between linear and nonlinear models, while another axis provides the division between empirical and first-principles models.

Driven by the need to reduce the computational time of large-scale model-based optimisation techniques in an on-line, real-time environment, research in the academic community has recently shifted towards efficient numerical methods and advanced solution strategies (Zavala et al, 2008). Interestingly, a research topic that has received little attention is the formulation of control and/or optimisation problems by operators, and subsequent interpretation of this formulation into a mathematical representation to be provided to and solved by a numerical solution algorithm (Rolandi & Romagnoli, 2008).

In this work, we change the perspective upon which research and development on APC techniques such as MPC and RTO is approached. Instead of adopting the conventional focus, we visualise the space of APC innovation as being divided along three main axes: the model formulation, the problem formulation, and the solution methods.

This idea has been motivated by a vision where flexibility and interoperability are the key technological differentiators of the next generation of model-based APC platforms. For example, such APC engine would allow embedding linear models as easily as nonlinear models. Similarly, the APC engine would support empirical/semi-empirical models derived from identification- or reduction-based techniques, as well as rigorous mechanistic models derived from first principles. Concurrently, the APC system would allow unconstrained, quadratic cost problem formulations or general constrained control problems. Finally, this next-generation APC engine would support discrete- and continuous-time formulations interchangeably, and would be integrated into the MPC multivariable control and/or RTO economic optimisation layers of the APC hierarchy.

In this work, we present a novel DRTO software platform that successfully decouples the three main axes of innovation on APC. This DRTO engine is used on a number of case studies to reveal the effect of the model formulation, the problem formulation and the solution methods. The performance of the DRTO engine is analysed on a simulated industrial process system.

2. DRTO ENGINE

The DRTO engine presented in this work is based on an architecture presented elsewhere (Rolandi & Romagnoli, 2008). In this work, we describe the elements of this architecture in some detail.

2.1 Modelling Engine

The modelling engine (ME) encapsulates all model-formulation services needed by the DRTO kernel. Naturally, the choice of ME imposes restrictions on the type of models that can be embedded in the DRTO engine

The advanced process modelling package gPROMS, from Process Systems Enterprise Ltd, has been chosen as the ME for the current implementation. gPROMS is an equation-oriented modelling system that supports the description of hybrid continuous/discrete (HCD) integro-partial-differential-algebraic systems (IPDAEs) of arbitrary complexity.

2.2 Models

Implicit differential-algebraic systems (DAE) are the most common class of models arising from first-principles

modelling. In general, these models are highly nonlinear due to phenomena such as kinetics, transport and equilibrium relationships. Linear models are a common class of models used in linear constrained/unconstrained MPC. These models are obtained by empirical identification or they result from linearising nonlinear models around one or more nominal operating points. In the former case, this result in a linear time-invariant (LTI) model; in the latter case, the model is linear time-variant (LTV). gPROMS has a hybrid structural-and-numerical linearisation algorithm which allows the transformation of implicit nonlinear models into explicit, linear state-space form.

2.3 Solution Engine

gPROMS is both a *model server* and a *solution engine* (SE). All model-based activities executed via the gPROMS API have access to the built-in numerical solution algorithms.

2.4 Solution Methods

As a SE, gPROMS provides a number of solver components for the solution of common numerical problems arising in simulations and optimisations. These are direct sparse linear algebra routines (MA28/MA48), a sparse (Quasi-)Newton nonlinear solver (SPARSE), a nonlinear solver with a proprietary block-decomposition algorithm (BDNLSOL), BDF (DASOLV) and IRK (SRADAU) implicit integrators (with sensitivity evaluations on request, via an augmented-system approach), a SQP-NLP solver (SRQPD), as well as single-shooting (SS) (CV_SS) and multiple-shooting (MS) (CVP_MS) dynamic optimisation (DO) solvers adopting a sequential-solution approach (also known as control vector parameterisation, or CVP).

2.5 Problem Definition Manager

Changes in process conditions that take place at different intrinsic time scales and changes to operation specifications imposed by operators and production managers are the norm in industrial process systems. As a result, the control-and-optimisation problem specification needs to change constantly to reflect these circumstances. The DRTO engine adopts an event-based mechanism to deal with these situations. In previous studies (Rolandi & Romagnoli, 2008), we have found that approximately a dozen different types of control events are needed to define typical industrial control/optimisation problems in most common situations.

The translation of these operator-posted control/optimisation events into a high-level dynamic optimisation problem is executed by the Problem Definition Manager software component (PDM), which is also responsible for implementing this formulation in the formalism requested by the modelling and solution engine.

Scaling of decision variables is performed automatically by gPROMS (and most optimisation routines). Scaling of constraints, which is not normally built-in in optimisation modules, is performed by the PDM using the same event-based mechanism. This transformation allows fine control of the enforcement of constraints which is needed to improve the robustness and efficiency of industrial control problems.

2.6 Solution and Feasibility Supervisor

The outcome of the PDM is, in general, a constrained dynamic optimisation problem, and the existence of a feasible solution is a key issue to be addressed by any industrial control application. The Solution-Feasibility Supervisor (SFS) is a dedicated component of the DRTO engine for monitoring the solution and handling of infeasibilities. The SFS is composed by a Solution Interpreter (SI) and a Constraint Manager (CM).

The CM has been built-in with two independent infeasibility-recovery strategies: i) relaxation or elimination and ii) ranking or identification. These individual strategies can be combined to create four different constraint-management policies.

The most common recovery mechanism in commercial MPC packages is constraint ranking and elimination (Qin & Badgwell, 2003). Here, one or more constraints are associated to a ranking level and removed from the control problem formulation upon infeasibility when their priority is below a cut-off level. Constraint identification uses information on the constraints' bounds and values (or their Lagrange multipliers, if available) to detect the subset that is active at any given iteration. Constraints that are violated are either eliminated or relaxed. The relaxation takes place by applying factors which are provided in the form of user events.

3. CASE-STUDY AND ANALYSIS OF RESULTS

3.1Industrial process

The case study presented in this paper is based on an industrial continuous pulping system of a state-of-the-art pulp and paper mill. The key unit in this process is a vertical tubular reactor with several fluid-phase extraction and introduction points along the length of the vessel. The reactor is a solid-liquid heterogeneous system with inter- and intra-particular transport phenomena, where the column of wood chips moves continuously down and the fluid phase moves in co-current or counter-current flow at different heights. Besides the tubular reactor, there is auxiliary heating and heat-recovery equipment consisting of nine unit operations (condensers, heaters, pre-heaters and a kettle-reboiler), and seven transportation and handling devices (the so-called feed line). There are around 50 PID control loops.

From a process perspective, the goal is to maximise the pulp yield (Y) at a given production target (P) while maintaining the deviation of pulp selectivity (S) from its quality-control target below a given threshold. Selectivity is measured in terms of the "kappa number" (#K) and it is an indication of the proportion of non-cellulosic components present in the wood pulp.

3.2 First principles model

Large-scale mechanistic models (that is, models of the order of 10,000 variables or more) have seldom been used in advanced model-based control systems, with only a few examples resulting from academic studies rather than industrial applications (e.g. Wisnewski & Doyle, 2001). A mechanistic model of the continuous pulping system described above has been implemented in gPROMS, resulting in a system of approximately 14,000 algebraic, 1,000 differential equations, and 300 degrees-of-freedom.

This model is used as the virtual plant for all closed-loop simulations. Depending on the model configuration, either the full nonlinear model is embedded in the optimiser or a reduced-order linear model. The latter is obtained by exact linearisation and minimal realisation of the first-principles' nonlinear model at the nominal steady-state starting point. Linearisation requires full knowledge of the operating point, that is, forcing input trajectories and state variable values of the system. In order to satisfy this requirement as well as the need for state feedback a perfect observer is implemented.

While a perfect observer is not feasible in an industrial setting, this choice allows us to focus on algorithmic and performance aspects and avoid any issues introduced by the ubiquitous plant/model mismatch that would be present in any real industrial application.

3.3 Case-studies set up

In this virtual-plant simulation, the process is initially at steady-state and control actions start taking place three hours before the scheduled production-rate change. The configuration of the control problem is such that the manipulated variables (MVs) optimised by DRTO are the setpoints of three key PID controllers: the chip meter (CM) speed (feed rate of wood chips), and the temperature of the lower (LH) and wash (WH) circulation heaters (which perform indirect heating of the reactor). Two interior-point and two end-point constraints are imposed on the trajectories and final values of selectivity and pulp production rate, and one path constraint is introduced for the deviation of selectivity with respect to its control target. The configuration of the time parameterisation is such that the prediction and control horizons are set to 7 and 6 hr, respectively, and the control window is 1 hr; these settings are representative of industrial practice.

In this work, real-time dynamic optimisation provides the setpoints of key loops of the regulatory control system, condensing the two intermediate optimisation/control layers of a conventional hierarchical control structure (i.e. MPC and RTO) into a single, consistent model-based application (DRTO).

3.4 Case-study 1

In this case study the main target is to control a production-rate transition from 600.0 tpd to 650.0 tpd. Figures 1 to 3 show the trajectories of key process variables for this case-study, for the linear (L) and nonlinear (NL) embedded model.

Note that the progression of control moves/manipulated variables (MVs) is similar for both controllers. However, while the NL-controller relies more heavily in the setpoint of the LH PID loop (also higher temperatures most of the time – peak temperature of ~156C), the L-controller uses more heavily the setpoint of the WH PID loop (also at higher temperatures –peak temperature of ~154C).

Having a look at the trajectories of controlled variables (CVs), the selectivity remains within the control limits at all times for the NL-controller (these limits were defined in the control problem formulation to be between kappa number 89 and 91). Overall, the NL-controller and the L-controller exhibit similar control performance.

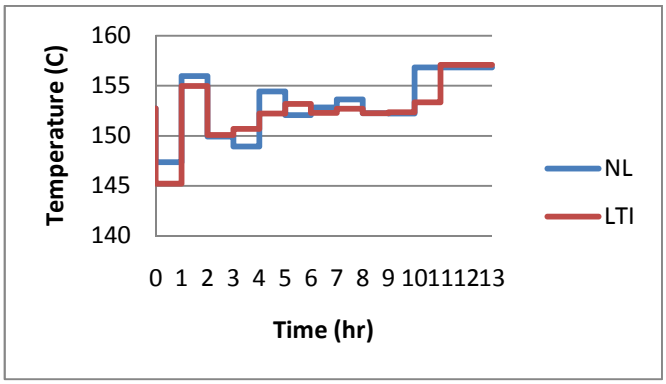

Figure 1: Lower heater (LH), MV.

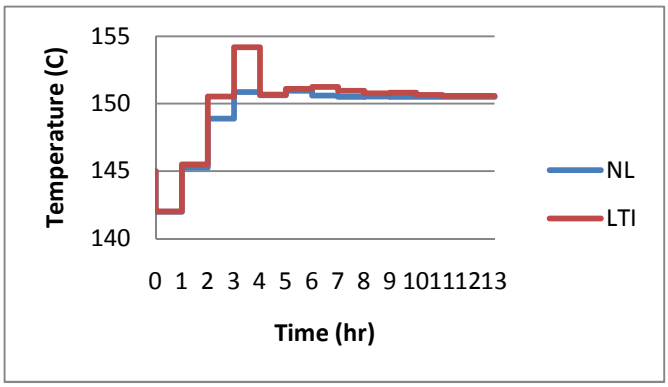

Figure 2: Wash heater (WH), MV.

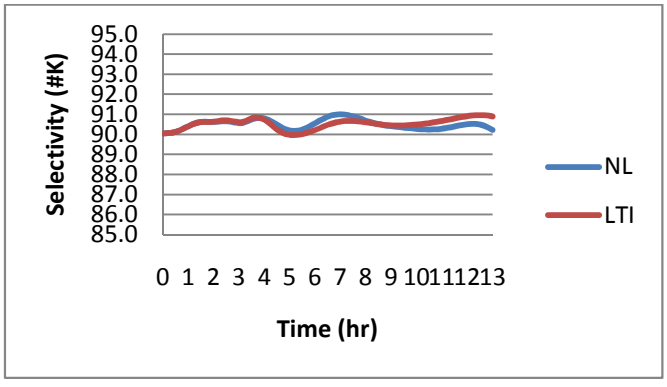

Figure 3a: Selectivity (S), CV (as seen by the operators).

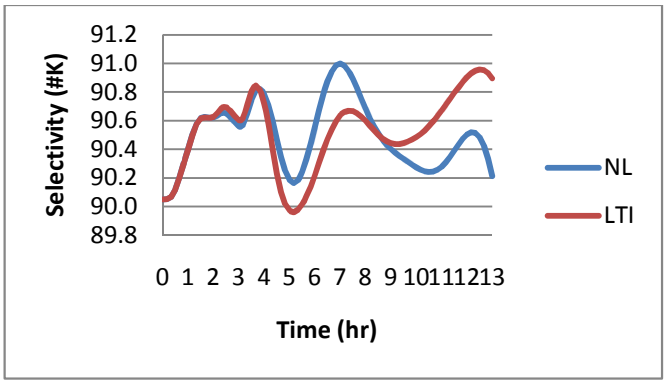

Figure 3b: Selectivity (S), CV (zoomed-in).

It is interesting to note that the solution of the control problem is feasible for both controllers; this occurs in the open-loop computation, as well as in closed-loop, when the control moves are applied to the plant.

The virtual-plant simulation can be used to examine optimality of the controllers. The mean integral value of the objective function for the NL-controller corresponds to a value of 64.98% yield; this compares with 64.95% for the L-controller. This shows that the NL-controller is optimal (locally, global optimality is not guaranteed), and the difference of performance amounts to US$50k/year. This small difference in profit does not necessarily justify the possible additional investment of a NL-controller.

The good performance of the L-controller is due to fact that the plant is reasonably linear in this limited range of operation and an accurate model of the plant has been obtained by exact linearisation of the first principles' nonlinear model. Also, the prefect observer periodically corrects the effects of plant/model mismatch by resetting the linear model to the true operating point.

3.5 Case-study 2

In this case study the main target is to control a production-rate transition from 600.0 tpd to 700.0 tpd, which is a more significant change to the operating level of the system than that examined in case study 1. Figures 4 to 6 show the trajectories of key process variables for this case-study, for the linear (L) and nonlinear (NL) embedded model.

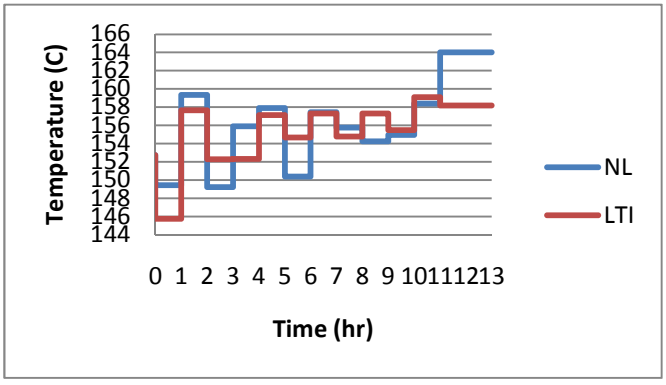

Figure 4: Lower heater (LH), MV.

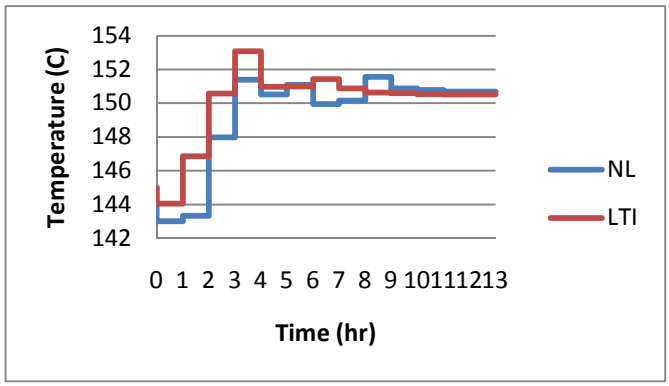

Figure 5: Wash heater (WH), MV.

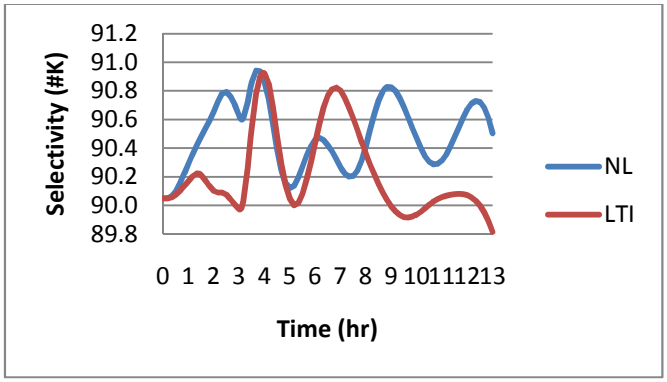

Figure 6: Selectivity (S), CV

By inspecting the control moves we can see that the L-controller relies more heavily on changes in the WH temperature setpoint as well as higher temperatures. The moves of the LH for both controllers show signs of slight oscillations which could be interpreted as aggressive control.

The trajectories of the output/controlled variable selectivity also show the aforementioned signs of oscillations. The dynamic responses differ qualitatively, but remain within the acceptable limits of industrial operation.

In terms of optimality and feasibility, the NL-controller fails to converge on the 3rd optimisation cycle. Interestingly, this does not cause a true infeasibility because the infeasible response is further down the prediction horizon. During the 4th /cycle, the NL-controller is able to recover from the episode alone if no infeasibility recovery scheme is applied.

3.6 Case-study 3

In the previous case-study we encountered the case were the NL-controller had an infeasibility on the 3rd cycle. In this case study we examine the performance of the two infeasibility recovery alternatives discussed in this work. The trajectories for identification-and-relaxation (IaR(1)) and ranking-and-elimination (RaE(3)) are shown in Figure 7.

It is clear from this case-study that the differences between the two schemes are minimal within the realisation horizon (the sequence of successive controls that have actually been applied into the plant). It can also be shown that both schemes outperform the base-case (without recovery scheme) since higher average yields are obtained without violating operating constraints.

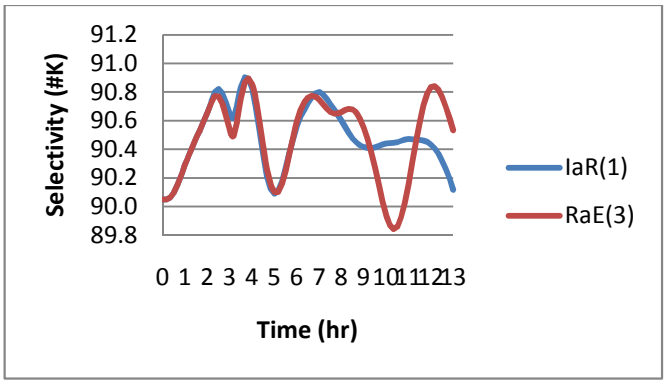

Figure 7: Selectivity (S), CV

While these results do not allow us to draw conclusions about the relative merits of both schemes, we argue that, in general, identification is superior to ranking as a constraint-differentiation scheme. Since elimination is a special case of relaxation, we expect that identification-and-relaxation would normally outperform ranking-and-elimination.

3.7 Case-study 4

This case study is a variation of case study 1. Again, a production-rate transition from 600.0 tpd to 650.0 tpd is scheduled, this time, 2 hr in advance. However, 2 hr after the transition a failure in downstream processing equipment triggers a production slow-down to the original rate of 600.0 tpd, which is enforced 2 hr later. This is a common yet challenging operating scenario in integrated pulp and paper mills which have low inventories by design.

The selectivity responses are quite different throughout the simulation horizon, While the NL-controller manages to boost selectivity from the very beginning, the L-controller gives rise to a sharp drop in selectivity and suffers from this performance degradation (lower objective function values) until it reaches similar levels than the NL-controller does around the 4th cycle.

The NL-controller reacts quickly to the change in the operational envelope and achieves such increase of selectivity by means of a sharp drop in the temperature setpoint of the WH as it can be seen on Figure 10, a claim which can be justified from a first-principles perspective.

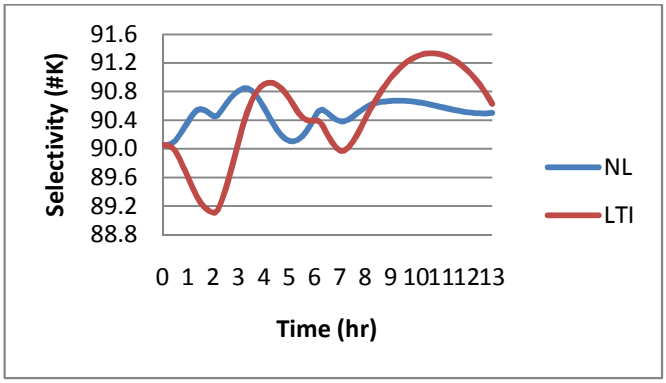

Figure 9: Selectivity (S), CV; tight tolerance.

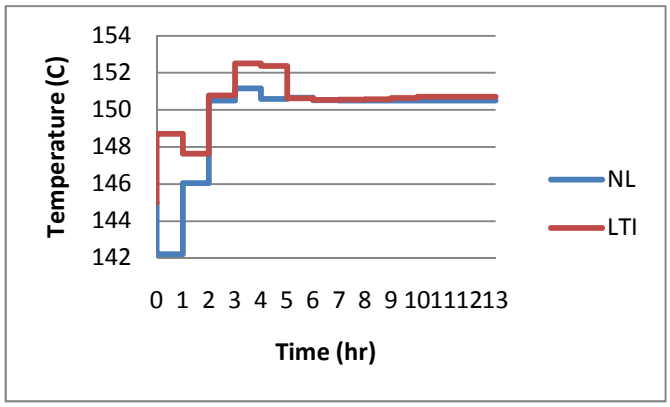

Figure 10: Lower heater (LH), MV; tight tolerance.

3.8 Case-study 5

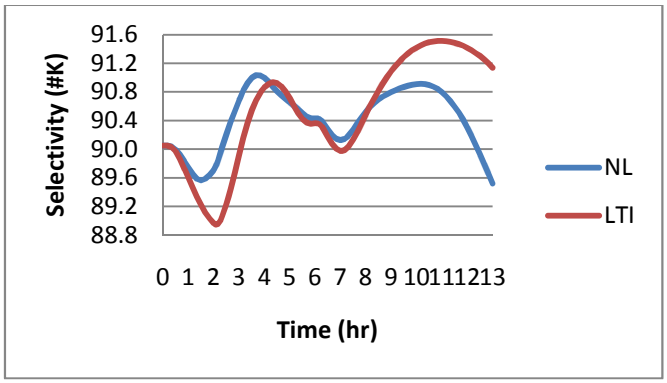

Figure 11: Selectivity (S), CV; loose tolerance.

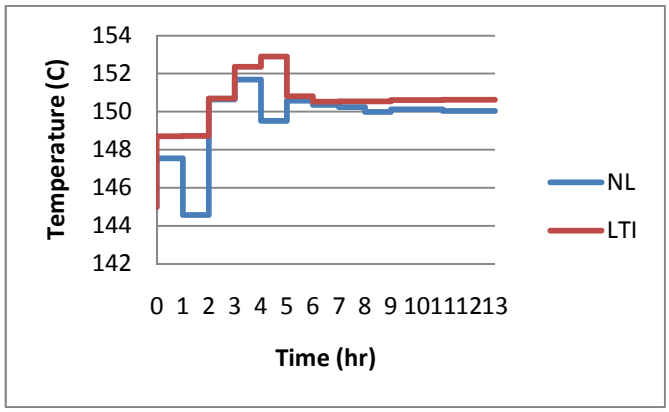

Figure 12: Lower heater (LH), MV; loose tolerance.

This case study is a variation of case study 4. The only difference resides in the optimisation tolerance setting, which has been relaxed from 1E-4 to 1E-3. In other words, Figures 9/10 and 11/12 show the responses for tight and loose tolerances, respectively, each one for a nonlinear and linear controller model. This case study is used to highlight the effect of problem configuration in the result of rigorous constrained optimisation activities.

Note that the results of the L-controller at a loose tolerance do not differ much from the performance at tight tolerance, although there are some minor differences and performance is generally better in the latter case.

Again, the most difference is seen in the early cycles of the transition. At a loose tolerance (Fig 11), the NL-controller behaves quite similarly to the L-controller and also suffers from a relatively sharp drop in selectivity, although the NL-controller still performs better overall.

3.9 Computational performance discussion

The following tables compile the key computational performance indicators for a representative case study. For each cycle (#C), we present the overall computational time (TC), the number of major nonlinear iteration (#I), the number of minor iterations or line searches (#LS), the unit cost of computation per major iteration (UC) and the ratio between sensitivity evaluations (gradient computations) and the total computation time (RSE).

Table 1. computational statistics; NL-controller

#C	TC	#I	#LS	UC	RSE
1	28.9	6	6	4.8	75
2	14.4	3	3	4.8	79
3	14.2	3	3	4.7	79
4	8.0	2	2	4.0	84
5	7.6	2	2	3.8	84
6	3.0	1	1	3.0	-
7	7.4	2	2	3.7	83
Mean	*11.9*	*2.7*	*2.7*	*4.4*	*81*

Table 2. computational statistics; L-controller

#C	Time	#I	#LS	UC	RSE
1	2.1	2	2	1.0	82
2	3.3	3	3	1.1	81
3	1.9	2	2	1.0	85
4	1.7	2	2	0.9	85
5	3.6	3	3	1.2	79
6	4.0	3	3	1.3	80
7	3.6	3	3	1.2	80
Mean	*2.9*	*2.6*	*2.6*	*1.1*	*82*

For both controllers, the determining cost is that of sensitivity calculations of the augmented system (79-85%). This fact is well known in the literature, and has been subject to the research of parallel computation of sensitivities (Keeping & Pantelides, 1998) which can reduce the computational cost significantly.

On average, the unit cost (per major/minor iteration) of a NL-controller is 4 times higher than that of a L-controller. The L-controller takes between 2 to 3 major iterations to converge. The NL-controller, on the contrary, takes anything between 1 to 6 iterations, at an average of 3 iterations approximately. The most computationally demanding cycle is the first, since the solution is started from an arbitrary set of initial guesses for the decision variables. Even in the most expensive iteration (NL-controller, 1st), the system can solve the control problem in real-time (at a speed up factor of approximately 2); on average the speed factor can be as high as 6 (NL-controller) and 20 (L-controller). Such excess computational time is a strong incentive for applying real-time dynamic optimisation, as well as for the development of advanced solution techniques, such as multi-scenario computations (Biegler, 2009).

4. CONCLUSIONS AND FUTURE WORK

In this manuscript we presented a framework and software platform that enables the adoption of different model types, solutions methods, and control-and-optimisation strategies transparently and effectively. We examined the performance of a novel DRTO engine through a number of industrial case studies using linear and nonlinear models, different problem formulations and different configuration settings. Both model-based controllers performed well and were able to drive the plant through production transitions satisfactorily. While the nonlinear model was optimal, the linear model resulted in much faster computational times. The two constraint identification and relaxation mechanisms for recovery of infeasibilities performed well. Future work will involve refinements to this framework as well as the development of observers for realistic closed-loop feedback with plant/model mismatch.

5. ACKNOWLEDGEMENTS

The authors wish to acknowledge the contribution of Dr Joseph Zeaiter to this work, who enthusiastically shared his vast experience in industrial APC.

REFERENCES

Biegler, L.T. (2009). Computers & Chemical Engineering, 27, 1-6.
Henson, M.A. (1998). Computers & Chemical Engineering, 23, 187-202.
Keeping, B.R. and Pantelides, C.C. (1998). Mathematics and Computers in Simulation, 44, 545-558.
Rolandi P.A. and Romagnoli, J.A. (2008). ESCAPE-18, Lyon, France.
Qin, S.J., Badgwell, T.A. (2003). Control Engineering Practice, 11, 733–764.
Wisnewski, P.A. and Doyle, F.J. (2001). IEEE Trans Control Sys Tech, 9, 435-444.
Zavala, V.M., Laird, C.D. and Biegler, L.T. (2008). Int. J. Robust Nonlinear Control; 18:800–815.