18th European Symposium on Computer Aided Process Engineering – ESCAPE 18
Bertrand Braunschweig and Xavier Joulia (Editors)

A real-time optimisation engine for nonlinear model-based optimising control of large-scale systems

Pablo A. Rolandi[a], José A. Romagnoli[b]

[a]*Process Systems Enterprise, 107a Hammersmith Bridge Road, London W6 9DA, UK*
[b]*Lousiana State University, South Stadium Road, Baton Rouge, LA, 70803, USA*

Abstract

Next-generation model-based advanced process control technologies should be centred on an architecture that allows the choice of models, solutions methods, control settings and optimisation strategies seamlessly. In this work a model-centric platform for advanced process control-and-optimisation of industrial manufacturing systems is presented, which aims at providing a framework for realising the aforementioned vision. This paper discusses the architecture of the dynamic real-time-optimisation (DRTO) platform and focuses on the formulation of on-line industrial optimisation-and-control problems from an operator viewpoint and its subsequent interpretation into a mathematical formalism. The optimal operation of an industrial continuous pulping system is examined as a case-study.

1. Introduction

For decades, industrial Model Predictive Control (MPC) technology has been based on linear empirical models obtained by identification from input-output process data. Typically, a discrete-time formulation is adopted, and the control problem is posed as an unconstrained optimisation problem with a quadratic-cost objective function. Even though applications following this successful technological paradigm will continue to deliver gains in conventional APC markets such as refining and petrochemicals, there is an increasing interest in nonlinear model-based control-and-optimisation, which is expected to meet higher requirements on productivity and quality control due to the intrinsic nonlinear nature of industrial manufacturing processes. In the past, however, large-scale mechanistic models 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; Leineweber et al., 2003).

We envision that *flexibility* and *interoperability* may be the key technological breakthrough of the next generation of model-based APC systems. For example, such an APC engine would allow embedding linear models as easily as linearised or nonlinear ones. Similarly, this APC engine would support (semi-)empirical models derived from identification- or reduction-based techniques, as well as fundamental mechanistic models derived from first principles. At the same time, the APC system would allow unconstrained, quadratic cost (MPC-like) optimisation-problem formulations or general constrained (RTO-like) ones. Finally, this next-generation APC engine would support discrete- and continuous-time formulations interchangeably (typical of MPC and RTO formalisms, respectively). Of course, one would not expect the form of the optimal control problem to depend on the characteristics of the APC application and, therefore, a set of mechanisms to formulate (and subsequently interpret) this control problem should be provided to the users.

In summary, we argue that next-generation APC systems should be founded on a domain framework and software platform that allows the interchange of models,

solutions methods and control/optimisation settings/strategies, seamlessly. One could anticipate that such a framework/platform would enable a transparent comparison of the benefits and drawbacks of controller-design choices made at each of these levels.
In this paper, leveraging on the current functionalities of modern advanced process modelling (APM) systems, a model-centric platform for advanced process control-and-optimization of industrial processing systems is presented. It provides the appropriate framework through which the aforementioned research issues can be investigated and addressed in a thorough and systematic way. Specifically, we devote our attention to the architecture of the DRTO platform and we focus on the formulation of (on-line) industrial optimization-and-control problems from an operator/process engineer viewpoint and its subsequent interpretation into a mathematical formalism.

2. Real-Time Optimisation Engine

2.1. Control Architecture
The kernel of the DRTO engine attempts to solve a moving finite horizon open-loop optimal control problem. Feedback is introduced at the end of the control window and the procedure is repeated according to a moving-horizon strategy. The formulation of the overall algorithm has been presented elsewhere (Rolandi & Romagnoli, 2005).
The conventional segregation of the control hierarchy into i) plant-wide steady-state optimisation, ii) unit-wide steady-state/dynamic optimisation, iii) dynamic multivariable constrained control and iv) regulatory control is a de-facto standard for industrial control systems. In this work, rigorous on-line real-time dynamic optimisation directly provides the set-points of the regulatory control system, condensing the two intermediate optimisation/control layers of a conventional hierarchical control structure into a single, consistent model-centric application.

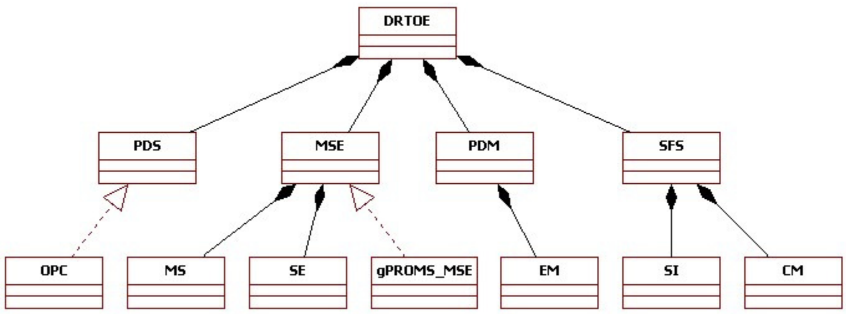

Figure 1. UML diagram of the dynamic RTO engine (DRTOE) software architecture.

2.2. Software Architecture
As shown by the UML diagram in Fig. 1, the dynamic RTO engine (DRTOE) software application is constructed by composition of several software objects: the process-data server (PDS), the modelling-and-solution engine (MSE), the problem-definition manager (PDM) and solution-feasibility supervisor (SFS).
The PDS is a client-side wrapper for an OPC (Historian) Data Access server or similar set of objects, interfaces and methods for connectivity to real-time and historical data from PLCs, DCSs, and other control devices and automation systems. In brief, the PDS software object provides the DRTO engine its on-line connectivity capabilities.

2.2.1. Modelling-and-Solution Engine

The Modelling-and-Solution Engine (MSE) is the component that provides both model-server services and numerical solution capabilities in a form that is consistent with the standard CAPE-OPEN interfaces, so that the implementation of the process model and solution methods is delegated to dedicated software objects. This way, the MSE provides a high-level abstraction of the modelling and solution functionalities demanded by the DRTO kernel.

In this work, gPROMS has been the MSE of choice: gPROMS is a powerful declarative, equation-oriented modelling system that supports the definition of hybrid continuous/discrete (HCD) integro-partial-differential-algebraic systems (IPDAEs) of arbitrary complexity. gPROMS' internal model representation is exposed to model-clients through the standard CAPE-OPEN interfaces via an ESO/STN formalism. In addition, not only is gPROMS an state-of-the-art model-server (MS) but also an advanced, robust and efficient solution engine (SE), since model-based activities executed via the gPROMS Server use gPROMS' numerical solution algorithms. These include direct sparse linear algebra routines (MA28/MA48), a sparse (Quasi-)Newton nonlinear solver (with proprietary block-decomposition and preset-propagation algorithms for increased robustness and speed), BDF/IRK implicit integrators (with sensitivity evaluations on request, via an augmented-system approach), a SQP-NLP solver, as well as single- and multiple-shooting (SS/MS) dynamic optimisation (DO) solvers (via a sequential-solution approach). Overall, gPROMS provides all modelling-and-solution services requested by a general-purpose application like the DRTO engine.

2.2.2. Event Manager

The framework upon which the DRTO engine was conceived is centred on the vision of translating any possible process control problem (e.g. normal/abnormal operation, disturbance rejection, production/grade transitions, etc) into an equivalent dynamic optimisation problem formulation. In (semi-)batch and continuous manufacturing processes, the nature of the control problem changes inevitably, due to the interaction with the surroundings and decision-makers (e.g. operators/control system). Indeed, in industrial processing plants, control variables may be "lost" due to hardware or software signal failure or unavailable due to direct intervention from the operator or the supervisory control system. Similarly, specifications of controlled variables may be modified to better represent operational requirements and safety limits. At the same time, an ill-conditioned system may arise as a consequence of poor definition of the control-problem and/or abnormal process performance, and an adequate modification of the input/output configuration could help recovering from this adverse situation. In summary, the structure of an industrial control problem will change dynamically.

Following the initiative adopted in our previous work (Rolandi & Romagnoli, 2005), the DRTO engine handles the interaction with external agents (surroundings/decision-makers) according to the following use case: a) at any given point in time, the operator or the control system adds to a list one or more events which encapsulate a desired feature of the control problem according to event-specific semantics; b) at any point in time, the DRTO kernel interprets the list of events into a dynamic optimisation problem (DOP) that represents the control problem the operator/control system expect to be resolved. Next, let us examine the nature of the events currently supported.

Our experience shows that a dozen or so different types of events are needed to define the formulation of typical industrial control problems. The "ObjectiveFunctionChange" event-type changes the identity of the objective function (the key performance indicator from the perspective of process performance) as well as the nature of the extremisation problem (minimisation/maximisation). As their names indicate, the event-types

"Prediction-HorizonChange", "ControlHorizonChange", "ControlWindowChange" and "Number-OfControlIntervalsChange" trigger modifications to the length/number of control intervals and length of the control/prediction horizons. In the case of adopting a sequential solution strategy of the underlying dynamic optimisation problem (gPROMS' approach), these changes are closely related to the control vector parameterisation (CVP) formulation which affects the multi-stage finite-dimensional approximation of the original infinite-dimensional problem.

The "ControlChange" event-type is used to specify the identity, bounds and initial guess of optimisation (decision) variables, as well as their explicit form of parameterisation (piecewise-constant and piecewise-linear are supported). "EndPointConstraintChange" and "InteriorPointConstraintChange" event-types are used to set end-point and interior-point constraints on state/output variables (Vassiliadis et al, 1994a).

The subset of events above is known as elementary events because they have a direct mapping into basic features of the dynamic optimisation problem formulation. In the case of gPROMS, this means that these events have a direct representation into the gPROMS optimisation-entity language. In our original manuscript (Rolandi & Romagnoli, 2005) we suggested the notion of composite events, that is, high-level, non-trivial even types that can be re-interpreted in terms of lower-level elementary events. In this work we introduce two composite events: "PathConstraintChange" and "ZoneConstraint-Change". These event-types, which denote path- and bounded-region-constraints which are very common in most engineering control/optimisation problem of interest, are reformulated internally by the DRTO engine as equivalent interior- and end-point constraints (see Vassiliadis et al, 1994b). The introduction of these composite events greatly simplifies the definition of industrial control-and-optimisation problems.

Another novelty of this work is the addition of event-types controlling the scaling of constraints and objective function (scaling of decision variables is also performed automatically by gPROMS), as well as other numerical aspects of the optimisation problem. The events-types, "VariableEnforcementChange", "PathEnforcementChange" and "ZoneEnforcementChange" are used for this purpose: while the former determines the degree of accuracy/enforcement of elementary end-point and interior point constraints, the second and third types apply to the elementary constraints that result from the interpretation of composite-constraint events. Finally, the user is able to control other algorithmic features such as the optimisation tolerance and event-time tolerance through their corresponding event-types.

As indicated by Fig. 1, the Event Manager (EM) component of the DRTO kernel is responsible for listening-to, parsing and validating individual user-posted events. It is expected that in the future this component will make use of validating XML schema and will be driven by user-interaction with a GUI. At the moment, events are edited and posted as structured plain-text lists.

2.2.3. Problem Definition Manager

Multivariable constrained control problems are difficult to pose. Furthermore, they must be interpreted into an equivalent dynamic optimisation-problem formulation and, in turn, mapped as input arguments to low-level numerical routines or language constructs of high-level modelling systems (e.g. gPROMS optimisation-entity language). To the best of our knowledge, this is the only system that adopts this overall philosophy.

The Problem Definition Manager (PDM) is a central element of the DRTO engine. The goal of the PDM component is to interpret user-posted events into an optimal-control-problem formulation, and resolve any conflicting elementary/composite events during the interpretation process. The PDM operates in two different modes. If new events are posted, or existing events become active (their corresponding event times are smaller than the actual time), the PDM constructs an entirely new problem-definition from scratch. If no events become active at the next control window, the existing problem formulation is recalculated according to a simple moving-horizon update. In general, the mapping of single elementary events into corresponding high-level problem formulation (and in the case of gPROMS corresponding declarative language constructs) is neither trivial nor convoluted provided that the semantics of the individual event-types are unambiguous. Interpreting composite events as well as the (rather common) case of multiple elementary events for a single process variable is more involved.

2.2.4. Solution and Feasibility Supervisor

Since the problem resulting from the PDM is a constrained dynamic optimisation problem, the existence of control-specification related constraints create a serious problem for on-line real-time applications: existence of a feasible solution. Given the fact that infeasibilities will occur, the proposed framework would have little success in an industrial setting if it was not able to recover from infeasibilities gracefully. Following the initiative we took in our original investigation, in this work we adopt the Solution-Feasibility Supervisor (SFS) as the dedicated component of the DRTO engine for monitoring the solution progress and handling infeasibilities. The SFS is further composed by a Solution Interpreter (SI) and a Constraint Manager (CM), as described below.

The Solution Interpreter monitors the progress of the open-loop optimisation computation with the goal of avoiding solution failures and slow convergence due to constraint infeasibility/inconsistency. In essence, it logs the evolution of the numerical solution (magnitude of control variables, constraints violations and corresponding Lagrange multipliers) as well as the computation statistics (number of NLP iterations, line searches and corresponding times, etc) for performance monitoring and forecasting. Upon termination of the execution these logs are saved into a file as records of the open-loop optimisation computation.

At the moment of writing, the Constraint Manager has yet to be implemented because a number of technical issues (mostly timely access to solution and computation statistics in between gradient evaluations and line searches) impede the realisation of some key features of this component. In the future, it is expected that the CM will support two infeasibility recovery mechanisms: i) the less rigorous (and more straightforward to implement) constraint-ranking-and-elimination approach common of third-generation industrial MPC technology (Qin & Badgwell, 2003); and ii) the more rigorous infeasible-constraint identification-and-relaxation strategy. In the first approach, a sub-set of low-priority constraints are dropped upon infeasibility; in the second strategy, a minimal subset of inconsistent constraints is first identified as problematic and then their enforcement is relaxed progressively until feasibility is achieved.

3. Case-study

The case study presented in this section is based on an industrial continuous cooking digester (and its auxiliary units) of a state-of-the-art Pulp and Paper mill. The model of such process system has been implemented in gPROMS resulting in a medium-to-large-scale system of approximately 14,000 algebraic and 1,000 differential equations (DAEs) and 100 state transition networks (STNs). In this case study we use the DRTO engine to control a production-rate transition from 600.0 ad.ton/day to 650.0 ad.ton/day. From an operations-and-control perspective, the goal is to maximise the pulp-yield (Y) at a given production target (P) while maintaining the deviation of pulp-selectivity from its quality-control target below a given threshold. The process is initially at steady-state and control actions start taking place two hours before the scheduled production-rate change. However, two hours after the transition a trip in the paper machine forces a production slow-down to the original rate of 600.0 ad.ton/day, which is enforced only two hours later. The DRTO engine drives the operations of the continuous cooking digester via the set-point of three controllers: the chip meter (CM) speed (feed rate of wood chips), and the temperature of the lower (LH) and wash (WH) circulation heaters (indirect bulk heating). Two interior-point and end-point constraints are imposed on the trajectories and final magnitude of blow-line kappa number and brownstock pulp production rate. The DRTO engine's prediction and control horizon are set to 7 and 5 hr, respectively, and the control window is 1 hr. Figures 1 and 2 show the trajectories of key process variables for this case-study. The DRTO engine successfully finds a set-point trajectory for the regulatory control layer that keeps all controller variables within their designated operative bounds and also minimises the consumption of raw-materials (maximises the yield) at a given target production rate.

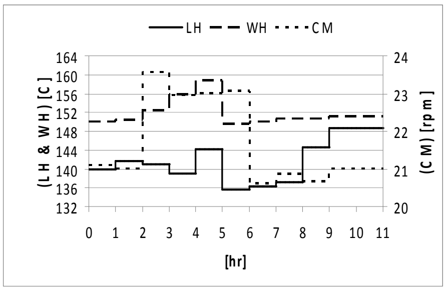

Figure 1: Manipulated variables (MVs).

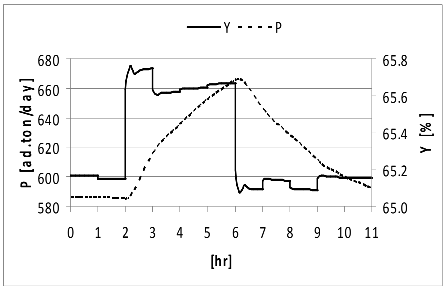

Figure 2: Controlled variables (CVs).

4. Conclusions and Future Work

In this manuscript we discussed the vision of a domain framework and software platform that would enable the adoption of different model forms, solutions methods, and control/optimisation settings/strategies transparently and effectively in an industrial setting. The DRTO engine presented here is a proof-of-concept of this vision which enables industrialists to engage in a comparison of the pros and cons of controller-design choices, including assessing optimality performance, feasibility-recovery schemes and computational load. Future work will involve the implementation of the Constraint Manager and address the subject of plant-model mismatch.

5. Acknowledgements

PA Rolandi acknowledges the funding of the Marie Curie Research Training Network under the programme PROMATCH, contract number MRTN-CT-2004-512441.

References

Vassiliadis, V.S., Sargent, R.W.H. and Pantelides, C.C. (1994a). Industrial & Engineering Chemistry Research, 33, 2111-2122.

Vassiliadis, V.S., Sargent, R.W.H. and Pantelides, C.C. (1994b). Industrial & Engineering Chemistry Research, 33, 2123-2133.

Wisnewski, P.A. and Doyle, F.J. (2001). IEEE Trans Control Sys Tech, 9, 435-444.

Leineweber, D.B., Bauer, I., Bock, H.G., Schloder, J.P. (2003). Comp & Chem Eng, 27, 157-166.

Rolandi P.A., Romagnoli, J.A. (2005). ESCAPE-15, Barcelona, Spain.

Qin, S.J., Badgwell, T.A. (2003). Control Engineering Practice, 11, 733–764.