Emerging Technologies for Enterprise Optimization in the Process Industries

Rudolf Kulhavý*
Honeywell Laboratories
Pod vodárenskou věží
CZ-182 08 Prague, Czech Republic

Joseph Lu†
Honeywell Hi-Spec Solutions
16404 N. Black Canyon Highway
Phoenix, AZ 85023 U.S.A.

Tariq Samad‡
Honeywell Laboratories
3660 Technology Drive
Minneapolis, MN 55418 U.S.A.

Abstract

We discuss three emerging technologies for process enterprise optimization, using a different application domain as an example in each case. The first topic is cross-functional integration on a model predictive control (MPC) foundation, in which a coordination layer is added to dynamically integrate unit-level MPCs. Enterprise optimization for oil refining is used to illustrate the concept. We next discuss data-centric forecasting and optimization, providing some details for how high-dimensional problems can be addressed and outlining an application to a district heating network. The final topic is adaptive software agents and their use for developing bottom-up models of complex systems. With learning and adaptation algorithms, agents can generate optimized decision and control strategies. A tool for the deregulated electric power industry is described. We conclude by emphasizing the importance of seeking multiple approaches to complex problems, leveraging existing foundations and advances in information technology, and a multidisciplinary perspective.

Keywords

Model predictive control, Cross-functional integration, Oil refining, Data-centric modeling, District heating, Adaptive agents, Electric power

Introduction

The history of advances in control is the history of progress in the level of automation. From single-loop regulation to multivariable control to unit-level optimization, we have seen step changes in the efficiency, throughput, and autonomy of operation of process plants. The next step in this progression is the optimization of the enterprise—a process plant, a multifacility business, or even a cross-corporate industry sector.

Enterprise optimization is the object of significant research today. Given the diversity of potential applications and the relative newness of the topic, it is not surprising that no one approach has been accepted as a universal solution. Although it is conceivable that future research will identify a cross-sector solution, at this point it appears that multiple approaches are likely to be necessary to best cover the full spectrum of potential applications.

In this paper, we discuss three emerging technologies for process enterprise optimization: cross-functional integration on a model predictive control (MPC) foundation, data-centric modeling and optimization, and adaptive agents. The discussions are generic, but different process applications are used in each case for illustration: oil refining, district heating, and electric power.

Enterprise Optimization on an MPC Foundation[1]

MPC has proven to be the most viable multivariable control solution in the continuous process industries and it is becoming increasingly popular in the semibatch and batch industries as well. Most industrial MPC products also include a proprietary economic optimization algorithm that is essential for driving the process to deliver more profit. In terms of economic benefit, MPC is one of the most significant enabling technologies for the process industries in recent years, with reported payback times between 3 and 12 months (Hall and Verne, 1993, 1995; Smith, 1993; Sheehan and Reid, 1997; Verne and Escarcega, 1998).

Cross-Functional Integration as a New Trend

Traditionally, most MPC applications are used for stabilizing operations, reducing variability, improving product qualities, and optimizing unit production. In most cases, a divide-and-conquer approach to a complex plantwide production problem is adopted. In this approach, a large plant is divided into many process units, and MPC is then applied on appropriate units. The divide-and-conquer approach reduces the complexity of the plantwide problem, but each application can reach only its local optimum at best. In a complex plant, the composition of local optima can be significantly less than the potential global optimum. For example, the estimated latent benefit for a typical refinery is 2-10 times more than what the combination of MPC controllers can capture (Bodington, 1995).

One possible approach to plantwide control is employing a single controller that is responsible for the whole plant; however, this option is infeasible. To note just the most obvious issue, commissioning or maintenance would require the whole plant to be brought offline. An alternative and practical approach for delivering global optimization benefit is to add a coordination layer on top of all the MPC applications to achieve the global optimum. The coordination layer usually covers multiple functions of the plant, such as operations, production

*rudolf.kulhavy@honeywell.com
†joseph.lu@honeywell.com
‡tariq.samad@honeywell.com
[1]This section is adapted from Lu (1998).

scheduling and production planning.

Complexity of MPC Coordination. With the divide-and-conquer approach, transfer price is traditionally used for measuring the merit of an advanced control application from a plantwide view and over an appropriate time period. The transfer price of a product (or a feed) is an artificial price assigned to determine the benefit contribution of a process unit to the overall plant under an assumed level of global coordination. A common assumption in calculating transfer price is that the product produced in a unit will travel through the designed paths (or designed processes) with the designed fractions to reach the final designated markets. This assumption is not always valid due to a lack of dynamic coordination among the units.

This phenomenon is referred to as benefit erosion[2] (e.g., see Bodington, 1995) and is typically alleviated by manual coordination between different sections of the production processes. For example, after an advanced control application is implemented on a refinery's crude unit, the yield of the most valuable component often increases, whereas the yield of the least valuable component decreases. The scheduling group would detect the component inventory imbalances (which could cause tank level problems if not corrected in time) rippling through various parts of the refinery, and it would then coordinate the affected parts of the refinery to "digest" the imbalances. With feedback from scheduling and operations groups, the planning group would update its yield models to reflect the yield improvement and rerun the plantwide optimization to generate a new production plan and schedule.

The fundamental cause of benefit erosion, however, is a lack of global coordination or optimization. Generally, the more complex the production scheme, the greater the problem. Therefore, a complex plant, such as a refinery or a chemicals plant, presents a higher benefit potential for cross-functional integration. If a new dynamic or steady-state bottleneck is encountered, or if an infeasible production schedule results, the scheduling group and the planning group would have to work together with the operations group to devise a new solution. The final solution may take a few adjustments or iterations. For complex cases, this process can take a few weeks or even months.

Only when all operations and activities in the refinery are coordinated together will benefit erosion be minimized. Although the situation described in this refinery example may sound primitive, it is still one of the better cases. In reality, different parts of a refinery usually use different tools with different models on different platforms. Engineers and operators in different units look at the same problem with very different time horizons. All of these factors complicate plantwide coordination in practice.

Although the cross-functional integration approach requires existing infrastructure to be revamped, including the DCS hardware and supporting software systems, this is increasingly less of an issue. Hardware and infrastructure costs have dropped significantly in recent years, particularly when viewed as a percentage of the total advanced control project budget. Support and organizational "psychology" sometimes hinder progress, but the situation has improved as more and more refineries and chemical plants realize the benefits of advanced control.

Long- and Short-Term Goals. Cross-functional integration takes a holistic approach to the plantwide problem. The integration encompasses a large number of process units and operating activities. Practical considerations suggest that it proceed bottom up, integrating one layer at a time until the level of enterprise optimization is finally reached, as depicted in Figure 1.

In Figure 1, the various layers in the pyramid describe the plantwide automation solution structure and the decision-making hierarchy. Around the pyramid structure is the circle of supply chain, production planning and scheduling, process control, global optimization, and product distribution. As more layers are integrated into the cross-functional optimization, the integrated system will perform more tasks and make more decisions that are made heuristically and manually today. Long envisioned by many industrial researchers, such as Prett and Garcìa (1988), this concept is now being further developed with design details of hardware systems, software structures, network interfaces, and application structures.

As a long-term goal, enterprise optimization integrates all activities in the whole business process, from the supply chain to production, and further to the distribution channel. In addition, risk management can also be included as a key technical differentiator. Parallel to the structural development for such a general-purpose complex system, some proof-of-concept projects have been piloted, and experiments have been conducted on several different structures (Bain et al., 1993; del Toro, 1991; Watano et al., 1993). The multiple-layer structure described in Figure 1 is believed to provide the flexibility needed for implementing and operating such a system. Moreover, each layer can be built at an appropriate level of abstraction and over a suitable time horizon. The lower layers capture more detailed information of various local process units over a shorter time horizon, whereas the higher layers capture more of the business essence of the plant over a longer horizon.

As a short-term goal, cross-functional integration could include, in a refinery, for example, raw material allocation, inventory management, production management, unit process control, real-time optimization,

[2]The benefit loss when the assumed level of global coordination is not reached.

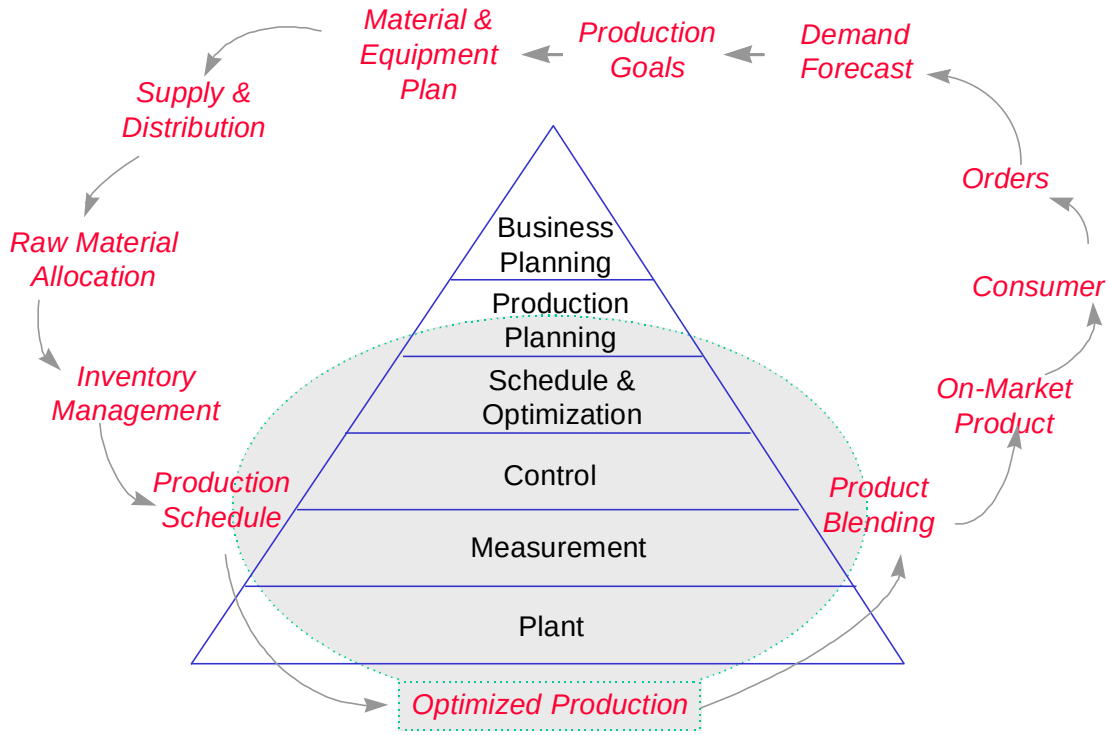

Figure 1: Enterprise optimization—refinery example.

product blending, production planning, and production scheduling. The instrumentation and real-time database are in the bottom layer. The regulatory control comes in the second layer. Following that are the MPC layer, the global/multiunit optimization layer, the production scheduling layer, and, last, the top production planning layer.

More relevant to the control community, this integration will have a large impact on almost all control, optimization, and scheduling technologies currently employed in the process industries. Advanced process control, primarily MPC and real-time optimization (RTO), will certainly be affected, particularly in terms of connectability, responsiveness, and compatibility.

MPC Considerations

This integration requires us to reassess the MPC and RTO designs in terms of their online connectability and their dynamic integration. There is a need for codesigning MPC and RTO, or at least designing one taking into account that it will be working with the other dynamically. Furthermore, from a control perspective, global coordination requires MPC applications to perform over a much wider operating region and more responsively. This poses new challenges to MPC technology, three of which are discussed below.

Nonlinear MPC. The majority of chemical process dynamics is, loosely speaking, slightly nonlinear and can be well modeled with a linearized model around a given operating point. With the application of CV/MV linearizing transformations, linear MPC can be extended to effectively handle a variety of simple nonlinear dynamics. Practical examples include pH control, valve position control, high-purity quality control, and differential pressure control. However, under cross-functional integration, a process will be required to promptly move its operating point over a significant span, or else to operate under very different conditions. Linear MPC may not be adequate in such cases.

Two examples can serve to illustrate. First, as the operating point of the process migrates, the dynamics of some processes may change dramatically, even to the extent that the gain will change sign. For example, some yields of the catalytic cracking unit in a refinery can change their signs when operated between undercracking and overcracking regions. Second, transition control presents a unique type of nonlinear control problem. During a transition, the process operating point typically moves much faster than usual, and the process typically responds more nonlinearly than usual. Examples include crude switch in a refinery and grade transition in many series production processes such as polymer and paper.

Both of these problems can be seen as instances of multizone control, where the process needs to be regulated

most of the time in each of the zones and to be occasionally maneuvered from one zone to another according to three performance criteria. The first criterion is for normal continuous operation in the first zone. The second criterion is for normal operation in the second zone. The third criterion is for special requirements during the transition or migration.

Although solving the multizone control problem alone has merit, solving it while coordinating with other parts of the plant will potentially provide much greater benefit. As enterprise optimization further advances, there is an increasing need for a nonlinear MPC tool that can solve multizone control problems. Likewise, there is an increasing need for MPC formulations, linear and nonlinear, that take into account the requirements of cross-functional optimization.

Model Structure. The MPC model structure is preferred to have a linear backbone or linear substructure. The linear model can be developed experimentally, and the nonlinear portion of the dynamics can be added as the need arises. This preference stems from the fact that, in most cases, one does not know a priori if a nonlinear model is necessary.

The model structure should also be scalable. Adding controlled variables (CVs) and manipulated variables (MVs) should not require the existing model to be reidentified or regenerated. The user should easily be able to eliminate any input-output pair of the model. The preferred solution should not require the system to be square or all the MVs to be available all the time.

The ability to share model information in the cross-functional integration scheme is important. The model obtained in an advanced control project is often the single most costly item in the implementation. Sharing model information with the other layers of cross-functional integration can show substantial monetary savings. This model sharing can be a two-way process: bottom up and top down.

The preferred model-sharing scheme is bottom up. It is much easier for engineers to find and correct modeling errors on a unit-by-unit basis in the control implementation than on a multiunit or plantwide basis under cross-functional integration. Bottom-up model sharing enhances usability, as engineers can verify their models as they commission the MPC applications unit by unit.

The solution technique needs to be algorithmically robust. If multiple solutions exist, it is preferable to stick to one branch of solutions throughout, or better yet, stick to the branch of solutions that requires only the minimum amount of control effort by some criterion.

Coordination Port. An essential requirement for cross-functional integration is an independent port for coordination. Simply sending a steady-state target as a set-point to an MPC controller is inadequate. The dynamic response required for coordination is often very different from that for set-point change, and the performance criterion can be completely different from set-point tracking or disturbance rejection. With a single port, it is difficult, if not impossible, to always satisfy both requirements.

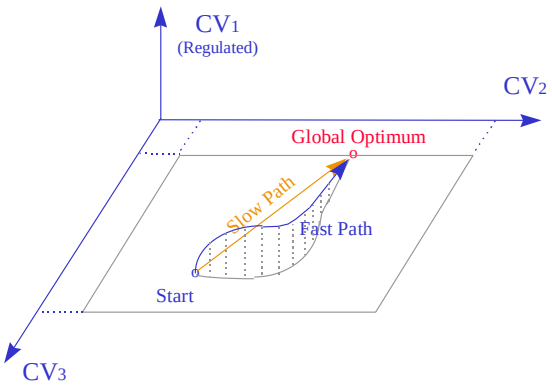

Figure 2: An example of different dynamic optimization paths.

A good coordination port implementation in MPC is one of the essential links in instituting cross-functional optimization. The nature of the problem is illustrated in Figure 2 as a simplified example of three CVs and two MVs. The first CV has a set-point, and the other two both have high and low bounds. (The MVs are not shown in the diagram.) The coordination or optimization target is solved by a global optimizer. The controller needs to drive the system to the target in an independent response for coordination. This is similar to designing a two-degree-of-freedom controller for the coordination port, except that the input direction and the response requirement may vary from one transition to another.

Furthermore, when the coordination port problem is not fully specified dynamically, as in many practical problems that we encounter, multiple solution paths to the destination exist, as depicted in Figure 2. Treating this as a traditional control problem by specifying a desired response trajectory is not always suitable for two reasons. First, the CV error violation is usually much less important in the transition than in normal operation. Second, except in trivial cases, one rarely knows a priori which transitional path would be financially optimal yet dynamically feasible.

An alternative solution is to solve for all equal solution paths in terms of the performance criteria (including financial terms) and choose the one that requires the minimum MV movement by some criterion. See Lu and Escarcega (1997) for one such solution to the coordination port implementation.

Summary

If MPC has been the main theme of the '80s and '90s in the process control industries, cross-functional integration and enterprise optimization will be the main theme of the next two decades. As we advance toward a higher level of computer-integrated manufacturing, the problem definition and the scope of classical model predictive control will be expanded. The cross-functional integration approach requires the dynamic coordination of multiple MPCs, not just in terms of steady-state operation, as in many steady-state RTO and composite LP approaches. One practical way of achieving dynamic coordination is by designing a coordination port in the MPC control strategies.

Although truly enterprise-scale applications along these lines are yet to be implemented, our initial projects in cross-functional integration are yielding exciting results. A cross-functional RMPCT[3] integration is discussed by Verne and Escarcega (1998), who report significant benefits. More recently, Nath et al. (1999) describe an application of Honeywell's Profit® Optimizer technology to an ethylene process at Petromont's Varennes olefins plant. In the Profit Optimizer approach, controllers are not operating in tandem, and the dynamic predictions of bridged disturbance variables are used by individual RMPCTs to ensure dynamic coordination/compensation among them. The controllers thereby work together to protect mutual CV constraints. The optimizer coordinates 10 controllers and other areas, covering the entire plant except for the debutanizer. During the acceptance test for the system, a sustained increase of over 10% in average production was achieved. This production level surpassed the previous all-time record for the plant by over 3.7% and was well above the expectation of a 2.7% increase.

Our experience to date in enterprisewide advanced control coordination has largely been limited to refining and petrochemical plants. For chemical plants, one of the key additional requirements is the integration of scheduling, product switchover, and other discrete-event aspects of plant operation within the formulation. The research of Morari and colleagues (Bemporad and Morari, 1999) on the optimization of hybrid dynamical systems is especially promising in this context.

Exploiting the Data-Centric Enterprise

From a technology that leverages the state-of-the-art in empirical-model-based control, we next turn to a more radical alternative to large-scale optimization. The key idea is that, where first principles or identified models cannot usefully be developed, we can consider historical data as a substitute. In other words, "the data is the model." This mantra was not especially useful even a few years ago, but now, with modern storage media available at affordable prices and computer performance increasing at a steady pace, entire process and business histories can be stored in single repositories and used online for enhanced forecasting, decision making, and optimization. This makes it possible to implement a data-centric version of intelligent behavior:

- Focus on what matters—by building a local model, on demand and for the immediate purpose.
- Learn from your errors—by consulting all relevant data in your repository.
- Improve best practices—by adapting proven strategies first.

The data-centric paradigm combines database queries for selecting data relevant to the case, fitting the data retrieved with a model of appropriate structure, and using the resulting model for forecasting or decision making.

When the size of the data repository becomes large enough, the architecture of the database (the data model) becomes crucial. To guarantee sufficiently fast retrieval of historical data, the queries need to be run against a specifically designed data warehouse rather than the operational database. Once the relevant data are retrieved, nonparametric statistical methods can be applied to build a local model fitting the data. Data-centric modeling can thus be seen as a synergistic merger of data warehousing and nonparametric statistics.

Data-centric models can form the basis for enterprise optimization. For example, the central problem of business optimization is matching demand with supply in situations when the supply or demand, or both, are uncertain. Suppose that a reward due to supply is defined as a response variable dependent on the decisions made and other conditions. Supply optimization then amounts to searching for maximum reward over the response surface.

In contrast to traditional response surface methods, data-centric optimization does not assume a global model of the response surface; rather it constructs a local model on demand—for each decision tested in optimization. Since the response is estimated through locally weighted regression (as discussed below), noise is automatically filtered. The uncertainty of the response estimate can also be respected in the optimization by replacing the estimated reward with the expected value of a utility function of the reward. By properly shaping the utility function, one can make decision making either risk-prone or risk-averse. As for the optimization algorithm itself, data-centric models lend themselves to any stochastic optimization method, including simulated annealing, genetic algorithms, and tabu search. (Response surface optimization is only one example of a data-centric

[3]RMPCT (Robust Multivariable Predictive Control Technology) is a Honeywell MPC product.

optimization scheme. Other schemes, such as optimization over multiunit systems and distributed optimization, are currently being investigated.)

We next contrast data-centric modeling with the more established global and local modeling approaches and provide some technical details. An overview of a reference application concludes this section.

Empirical Modeling

When exploring huge data sets, one must choose between trying to fit the complete behavior of the data and limiting model development to partial target-oriented descriptions.

The *global* approach generally calls for estimation of comprehensive models such as neural networks or other nonlinear parametric models (Sjöberg et al., 1995). The major advantage of global modeling is in splitting the model-building and model-exploitation phases. Once a model is fit to the data, model look-up is very fast. Global models also provide powerful data compression. After a model is built, the training data are not needed any further. The drawback is that the time necessary for estimation of unknown model parameters can be very long for huge data sets. Also, global models are increasingly sensitive to changes in the data behavior and may become obsolete unless they are periodically retuned.

The *local* approach makes use of the fact that often it is sufficient to limit the fit of the process behavior to a neighborhood of the current working point. Traditionally, local modeling has been identified with recent-data fitting. Linear regression (Box and Jenkins, 1970) and Kalman filtering (Kalman, 1960) with simple recursive formulae available for parameter/state estimation have become extremely popular tools. Their simplicity comes at some cost, however. Adaptation of local-in-time models is driven solely by the prediction error. When a previously encountered situation is encountered again, learning starts from scratch. Further, when a process is cycling through multiple operating modes, adaptation deteriorates model accuracy.

The data-centric approach is an alternative to both these prevailing paradigms. It extends *recent-data* fitting to *relevant-data* fitting (see Figure 3). This "shift of paradigm" combines the advantages of global and local modeling. Namely, it provides

- a global description of the data behavior,
- through a collection of simple local models built on demand,
- using all data relevant to the case.

The price we pay for this powerful mix is that all data need to be at our disposal at all times (i.e., no single compact model is returned as a result of modeling). The data-centric model can be built out of the original data stored in the database without any data compression.

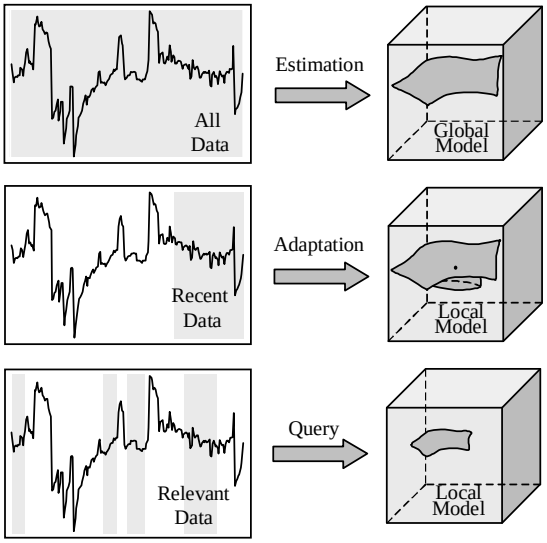

Figure 3: Global, local-in-time, and local-in-data modeling.

When applied for forecasting at the enterprise level, the data needs to be aggregated properly. The forecasting literature (see, e.g., West and Harrison, 1989) shows that forecasting from aggregated data yields more robust and more precise forecasts. The design of proper aggregation or more sophisticated preprocessing of data thus becomes a crucial part of data-centric modeling. Poor preprocessing strategies will need to be corrected before data-centric models can be effective.

Data-Centric Modeling with Locally Weighted Regression

To be more specific, we describe one possible implementation of data-centric modeling, namely, locally weighted regression with local variable bandwidth. The algorithm is by no means the only option and should be understood only as an illustration of the general idea.

It is difficult to trace the originator of local modeling. The concept appeared independently in various fields under names such as locally weighted smoothing (Cleveland, 1979), nonparametric regression (Härdle, 1990), local learning (Bottou and Vapnik, 1992), memory-based learning (Schaal and Atkeson, 1994), instance-based learning (Deng and Moore, 1994), and just-in-time estimation (Cybenko, 1996).

General Regression. Assume that a single dependent variable or response y depends on n independent or regressor variables $\varphi_1, \varphi_2, \ldots, \varphi_n$ through an unknown static nonlinear function $f(\cdot)$ with precision up to an unpredictable or stochastic component e

$$y = f(\cdot) + e.$$

The objective is to estimate response y_0 for any particular regressor φ_0.

Linearization in Parameters. Rather than trying to fit the above global model to all the data available, data-centric modeling suggests the application of a simpler model to data points $(y_1, \varphi_1), \ldots, (y_N, \varphi_N)$ selected so that $\varphi_1, \varphi_2, \ldots, \varphi_N$ are close to a specified regressor φ.

A typical example is a model linearized around a given column vector φ:

$$y = \theta'(\varphi)\varphi + e,$$

where θ denotes a column vector of regression coefficients and θ' its transposition.

Curse of Dimensionality. A word of caution applies here. As the dimension of φ increases, the data points become extremely sparsely distributed in the corresponding data space. To locate enough historical data points within a neighborhood of the query regressor φ_0, the neighborhood can become so large that the actual data behavior cannot be explained through a simplified model.

Consider, for instance, a 10-dimensional data cube built over 10-bit data; it contains $2^{100} \approx 10^{30}$ bins! Even a trillion (10^{12}) data points occupy an almost zero fraction of the data cube bins. Even with 5-bit (drastically aggregated) data, 99.9% of bins are still empty.

We have an obvious contradiction here. To justify a simple model, we must apply it in a relatively small neighborhood of the query point. To retrieve enough data points for a reliable statistical estimate, we must search within a large enough neighborhood. One must ask: Can data-centric modeling work at all?

Coping with Dimensionality. Luckily, real data behavior is rarely that extreme. First, the data is usually concentrated in several regions around typical operating conditions, which violates the assumption of uniform distribution assumed in mathematical paradoxes. Second, the regressor φ often lives in a subspace of lower dimension. That is, $\varphi = \varphi(x)$ where x is a vector of dimension smaller than the dimension of φ. Suppose, for instance, that y, x_1, and x_2 denote process yield, pressure, and temperature, respectively, and the model is a polynomial fit with

$$\varphi_i(x_1, x_2) = x_1^{m(i)} x_2^{n(i)}.$$

The dimension of regressor φ can easily be much larger than the dimension of x, but it is the dimension of x that matters here; it defines the dimension of a data space within which we search for "similar" data points.

Under the assumption $\varphi = \varphi(x)$, the model is linearized around the vector x:

$$y = \theta'(x)\varphi + e$$

and applied to data points $(y_1, x_1), \ldots, (y_N, x_N)$ selected so that $\|x_k - x_0\| \le d$ for $k = 1, 2, \ldots, N$, where $\|\Delta\|$ is an Euclidean norm of vector Δ, x_0 is a query vector, and d is a properly chosen upper bound on the distance.

Weighted Least Squares. The simplest statistical scheme for estimating the unknown parameters θ is based on minimizing the weighted sum of prediction errors squared:

$$\min_\theta \sum_{k=1}^{N} K(\|x_k - x_0\|)(y_k - \theta^T \varphi_k).$$

Each data point (y_k, x_k), $k = 1, 2, \ldots, N$ is assigned a weight inversely proportional to the Euclidean distance of x_k from x_0 through a kernel function $K(\cdot)$. Typical examples of kernel functions are the Gaussian kernel $K(\Delta) = \exp(-\Delta^2)$ or the Epanechnikov kernel $K(\Delta) = \max(1 - \Delta^2, 0)$.

The kernel function assigns zero or practically zero weight to data points (y, x) that appear too far from the query vector x_0. We can use this fact to accelerate database query by searching only for data points within the neighborhood of x_0 defined by $K(\|x_k - x_0\|) \le \varepsilon$, where ε is close to zero.

Performance Tuning. The performance of the above algorithm crucially depends on the definition of the Euclidean norm $\|\Delta\|$. In general, the norm is shaped by the "bandwidth" matrix S:

$$\|\Delta\|^2 = \Delta' S^{-1} \Delta.$$

Through S, it is possible to emphasize or suppress the importance of deviations from the query point x_0 in selected directions in the space of x-values. The matrix S depends on x_0 and can be determined, for example, by the nearest neighbor or cross-validation method (Hastie and Tibshirani, 1990).

Bayesian Prediction. The precision of prediction is an important issue with any statistical method, but in data-centric modeling it is even more pressing due to the relatively small number of data points used for local modeling. To quantify consistently the prediction uncertainty, one can adopt the Bayesian approach to estimation and prediction (Peterka, 1981). Compared with the least-squares method chosen above for simplicity, the Bayesian method calculates a complete probability distribution of the estimated parameters. The distribution is then used for calculating a probability distribution of the predicted variable(s). From the predictive distribution, any derived statistic such as variance or confidence interval can be computed. Due to the relative simplicity of local models, the Bayesian calculations, notorious for their computational complexity in many tasks, can be performed here analytically, without any approximation (for more details, see Kulhavý and Ivanova, 1999).

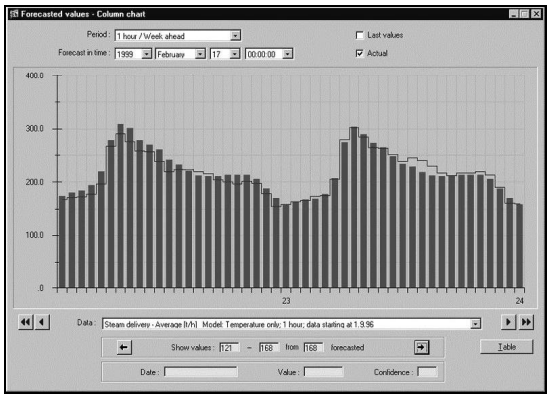

Figure 4: A comparison of predicted (bars) and actual (line) steam.

Reference Application

The first operating application of data-centric forecasting and decision making is an operator advisory system in a large combined heat and power company in the Czech Republic. The company supplies heat to a district heating network and electricity to a power grid. The whole system is composed of five generation plants, a steam pipeline network totaling 60 miles, and five primary, partially interconnected hot-water pipeline networks of a total length of 46 miles.

Data-centric forecasting is being applied to predict the total steam, heat, and electricity demand, heat demand in individual hot-water pipelines, and total gas consumption. The forecasts are performed in three horizons—15-minute average one day ahead, 1-hour average one week ahead, and 1-day average one month ahead (Figure 4). Altogether, this requires the computation of 1,632 forecasts every 15 minutes, 2,856 forecasts every hour, and 1,581 forecasts every midnight. The use of highly optimized data marts makes it possible to perform all the computations, including thousands of database queries, while still leaving enough time for other analytic applications.

Data-centric decision making is applied to optimization of set-points (supply temperature and pressure) on hot-water pipeline networks and to economic load allocation over 16 boilers. Insufficient instrumentation of hot-water pipeline networks is solved by complementing data-centric optimization with a simple deterministic model-based procedure for evaluating the objective function. The lack of data is thus compensated for with prior knowledge, combining the underlying physical laws and some simplifying assumptions. The solution can be adapted to a wide range of heat distribution networks with different levels of system knowledge and process monitoring. Data-centric modeling takes into account the outdoor temperature, current (or planned) electricity production, time of day, day of week, whether it is a working day/holiday, and the recency of data. Electricity production is considered to reflect the effect of operators' decisions in addition to external conditions.

Incomplete measurements on the boilers do not allow for online estimation of combustion efficiency for each boiler separately. Data-centric optimization is configured so as to exploit only available information. From the total fuel consumptions of generation plants and the total heat generated, the overall efficiency of the current boiler configuration is calculated. Data-centric optimization then searches in the process history and suggests possible improvements. For frequent situations and configurations, important for the company, enough points in the history are retrieved and reliable results are obtained. Different configurations are tested and evaluated while taking into account the costs of reconfiguration. One of the most valuable features of data-centric optimization is that it automatically adapts to changing operating conditions (e.g., variations in fuel calorific value, changes in boiler parameters, and aging of equipment).

Summary

A criticism frequently voiced about the data-centric approach is that it is incapable of modeling plant behavior in previously unvisited operational regimes and of handling changes in the plant, such as minor equipment failures, catalyst aging, and general wear and tear. In general, as with any statistical model, the quality of data-centric models depends crucially on the availability and quality of historical data. The lack of data can be compensated only by prior knowledge. One option is to populate the database with both actual and virtual data, the latter coming from a simulation model or domain expert (Kulhavý and Ivanova, 1999).

Thus, the data-centric approach itself suggests a solution to the problem of integrating data and knowledge. A historical database can be merged with a database populated with examples generated based on heuristics or conventional models. Different weightings can be associated with records that depend on their source. Thus historical data can be preferred for operational regions that are well represented in the recent history of plant operation, whereas "synthetic" data can be preferred for contexts for which process history provides few associated samples. In effect, a synthesis of multiple types of information sources can be achieved using the database as a common foundation.

To sum up, data-centric models can be applied to a range of problems in the process industries, subject only to an ability to satisfy the database-intensive search requirements and the absence of a single compact model that would permit closed-form analytic solutions. The former will cease to be a real constraint in a few years due to continuing progress in database and computer technol-

ogy. The latter issue is fundamental—with on-demand modeling, all decision making becomes iterative. For this reason, the data-centric model should not be considered a replacement for the classical results of decision and control theory, but as a tool that can extend the reach of automation and control to processes that have not been amenable to analytic methods.

Optimization with Adaptive Agents

The term *agent* is used in multiple senses in the computational intelligence and related communities. We use it here to mean software objects that represent problem domain elements. Agents can, for example, represent units and equipment in a chemical plant, parts of an electricity transmission and distribution network, or suppliers and consumers in supply chains (García-Flores et al., 2000). An agent must be capable of modeling the input/output behavior of a system at a level of fidelity appropriate for problems of interest. Couplings between systems, whether material or energy flows, sensing and control data, or financial transactions, can be captured through interagent communication mechanisms. In some sense, agent-based systems can be seen as extensions of object orientation, although the extensions are substantive enough that the analogy can be misleading.

One common use of agents is to develop bottom-up, componentwise models of complex systems. With subsystem behaviors captured with associated agents, the overall multiagent system can constitute a useful model of an enterprise such as a process plant or even an industry structure. Given some initial conditions and input streams, the evolution of the computational model can track the evolution of the physical system. Often a quantitative match will be neither expected nor obtained, but if the essential elements driving the dynamics of the domain are captured, even qualitative trends can provide insight into and guidance for system operation.

The increasing interest in agent systems can be attributed in part to advances in component software technology. Provided that common interface specifications are defined, agents can be developed in different programming languages and can be executing on different processes or computers. Agents can vary from very simple to extremely sophisticated, and heterogeneous agents may work together in a single application. "Plug-and-play" protocols for agent construction encourage code reuse and modularity while enabling agent developers to work in a variety of programming languages. Generic agent toolkits are now available (e.g., swarm `www.swarm.org`, Lost Wax `www.lostwax.com`, and Zeus `http://193.113.209.147/projects/agents/zeus/`) that can facilitate the design of agent applications for diverse problems. Languages for interagent communication for broad-based applications have also been developed. One such language, KQML, has been fairly widely adopted (Finin et al., 1994).

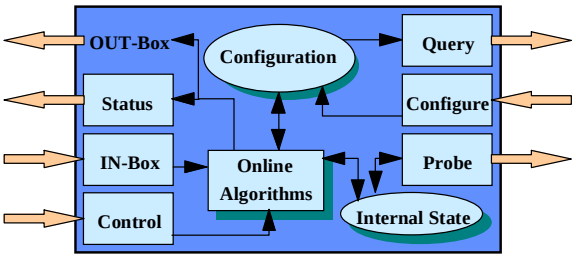

Figure 5: Abstract agent architecture.

Decision Making and Adaptation in Agents

Figure 5 shows an example abstract agent architecture, identifying some of the functions that are often incorporated. The In and Out boxes are the ports through which the agent exchanges messages with other agents; Status and Control are interfaces for the agent simulation framework; and the Query, Configure, and Probe features are used for initialization, monitoring, debugging, and so on. An agent can maintain considerable internal state information, which can be used by algorithms that implement its decision logic (its input/output behavior). Other online algorithms can be used to adapt the decision logic to improve some performance criteria (which may be agent-centric or global). The adaptation mechanism is often internal to an agent, but it need not necessarily be so.

The adaptation mechanism may be based on genetic algorithms (Mitchell, 1996), genetic programming (Koza, 1992), evolutionary computing (Fogel, 1995), statistical techniques, or artificial neural networks. These algorithms act on structures within an agent, but the adaptation is often based on information obtained from other agents. For example, a low-performing agent may change its program by incorporating elements of neighboring, better-performing agents.

The choice of learning algorithm interacts strongly with how the agent represents its decision-making knowledge and the kind of feedback available. For example, LISP programs may lend themselves to genetic programming. In a supervised learning situation, neural networks may employ an algorithm such as back-propagation or Levenburg-Marquardt minimization. However, in many agent applications, only weaker information is available (e.g., a final score, forcing agents to rely on reinforcement learning algorithms). The learner must solve temporal and spatial credit assignment problems—determining what aspect of its sequence of decisions led to the final (or intermediate) score and what part of its internal representation is responsible. Strategies such as temporal differencing and Q-learning have been proposed for this (Watkins and Dayan, 1992).

Multiple learning mechanisms can be incorporated

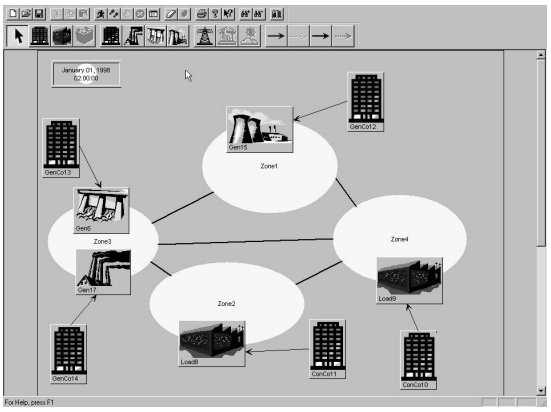

Figure 6: SEPIA interface showing a four-zone, three-generator, two-load scenario.

into the same agent environment, or even the same agent. "Classifier systems" frequently incorporate a genetic algorithm to evolve new classifiers over time. The use of multiple adaptation mechanisms can offer unsuspected advantages. One well-known example is the Baldwin effect, in which learning interacts with evolution even though the learned information is not inherited (Hinton and Nowlan, 1987).

Example: A Simulator for Electric Power Industry Agents

One industry where the integration of business and physical realms has recently taken on a new importance is electric power. Deregulation and competition in the power industries in several countries over the last few years have resulted in new business structures. At the same time, generation and transmission facilities impose hard physical constraints on the power system. For a utility to attempt to maximize its profit while ensuring that its power delivery commitments can be accommodated by the transmission system—which is simultaneously being used by many other utilities and power generators—economics and electricity must be jointly analyzed.

A prototype modeling and optimization tool recently developed under the sponsorship of the Electric Power Research Institute provides an illustration. The tool, named SEPIA (for Simulator for Electric Power Industry Agents), integrates classical power flow models, a bilateral power exchange market, and agents that represent both physical entities (power plants) and business entities (generating companies). Through a full-featured GUI, users can define, configure, and interconnect agents to set up specific industry-relevant scenarios (see Figure 6). A scenario can be simulated with adaptation capabilities enabled in selected agents as desired. For example, a generation company agent can search a space of pricing structures to optimize its profits subject to its generation constraints, the transmission capabilities of the system, and the simultaneous optimization of individualized criteria by other agents that may bear a cooperative or competitive relationship to it. Two reusable and configurable learning/ adaptation mechanisms have been incorporated within SEPIA: Q-learning and a genetic classifier system.

SEPIA models a power system as a set of zones interconnected by tie-lines. A transmission operator agent conducts security analysis and available transfer capacity calculation, including first contingency checks for all proposed transactions. A dc power flow algorithm and a simplified ac algorithm are included for this purpose.

Tools such as SEPIA have several potential uses:

- To identify optimized (although not necessarily optimal) operational parameters (such as pricing structures or consumption profiles in the power industry application).
- To determine whether an industry or business structure is stable or meets other criteria (such as fairness of access).
- To evaluate the potential benefits of new technology (e.g., superconducting cables and high-power transmission switching devices).
- To generally help decision makers gain insight into the operation and evolution of a complex system that may not be amenable to more conventional analysis techniques.

Further details on SEPIA are available in Harp et al. (2000). A self-running demonstration can be downloaded from the project Web site at `http://www.htc.honeywell.com/projects/sepia`.

Summary

Over the last several years, a technological convergence—increasing processing power and memory capacities, component software infrastructure developments, adaptive agent architectures—has made possible the development of a new class of simulation and optimization tools. With careful design, these tools can be used by nonexperts, and they can provide a level of decision support and insight to analysts and planners. It is, however, important to recognize that agency architectures—although the object of significant attention in both the software research community and the high-technology media—is no panacea (Wooldridge and Jennings, 1998). Problems that will benefit from an agent-oriented solution are those that have an appropriate degree of modularity and concurrency. Even then, considerable effort must be invested to endow agents with the necessary domain knowledge. For optimization applications, adaptation capabilities must be carefully enabled in agents to ensure that the flexibility of the agent-based approach

does not immediately imply drastic computational inefficiency.

Conclusions

We conclude with some "slogans" that reflect the motivations and philosophy underlying the research reported above.

Embrace pluralism. The more complex and encompassing the problems we attempt to solve, the less likely that any one solution approach will suffice. Key problem characteristics-such as the degree of knowledge and the amount of data available-will vary considerably across the range of enterprise optimization applications. Covering the space of problems of interest requires a multipronged research agenda and the development of approaches that are applicable across the diversity of problem instances.

Leverage existing foundations. New classes of problems need not mean new built-from-scratch solutions. Especially from a pragmatic perspective, an ability to create technology that can "piggy-back" on existing infrastructure can be a key differentiator. The extent of disruption of systems and processes is always a consideration in the adoption of new research results.

Exploit IT advances. Advances in hardware, software, and communication platforms do not just allow more complex algorithms to be run; they also suggest new ways of thinking about problems. The doing away with traditional technology constraints can be a liberating event for the research community, although it is important to remember that the inertia of a mature, established industry is a significant constraint in its own right.

Pursue multidisciplinary collaborations. Another corollary of complexity is that its management is a multidisciplinary undertaking (Samad and Weyrauch, 2000). In the current context, this is not only a matter of marrying control theory, chemical engineering, and computer science. As we attempt to automate larger-scale systems and pursue the autonomous operation of entire enterprises, any delimiting of multidisciplinary connections seems arbitrary.

It appears to be a law of automation that the larger the scale of the system to be automated, the more specialized and less generic the solution. At one extreme, the PID controller is ubiquitous across all industries (process, aerospace, automotive, buildings, etc.) for single-loop regulation. At the multivariable control level, MPC is the technology of choice for the process industries but has had little impact in others. For enterprise optimization, effective solutions will likely be even more domain-specific. The fact that we have discussed three very different technologies does not imply a fundamental uncertainty about which one of these will ultimately be the unique "winner"; rather, it reflects a fundamental belief that process enterprise optimization is too complex and diverse a problem area for any one solution approach to satisfactorily address.

References

Bain, M. L., K. W. Mansfield, J. G. Maphet, W. H. Bosler, and J. P. Kennedy, Gasoline blending with an integrated on-line optimization, scheduling, and control system, In *Proc. National Petroleum Refiners Association Computer Conference*, New Orleans, LA (1993).

Bemporad, A. and M. Morari, "Control of systems integrating dynamics, logic, and constraints," *Automatica*, **35**(3), 407–427 (1999).

Bodington, E., *Planning, Scheduling, and Control Integration in the Process Industries*. McGraw-Hill, New York (1995).

Bottou, L. and V. Vapnik, "Local learning algorithms," *Neural Computation*, **4**, 888–900 (1992).

Box, G. E. P. and G. M. Jenkins, *Time Series Analysis, Forecasting and Control*. Holden-Day, San Francisco (1970).

Cleveland, W. S., "Robust locally-weighted regression and smoothing scatterplots," *J. Amer. Statist. Assoc.*, **74**, 829–836 (1979).

Cybenko, G., Just-in-time learning and estimation, In Bittanti, S. and G. Picci, editors, *Identification, Adaptation, Learning*, NATO ASI Series, pages 423–434. Springer-Verlag, New York (1996).

del Toro, J. L., Computer-integrated manufacturing at the Monsanto Pensacola plant, Presented at AIChE Spring Meeting, Houston, TX (1991).

Deng, K. and A. W. Moore, Multiresolution instance-based learning, In *Proc. 14th Int. Joint Conference on Artificial Intelligence*. Morgan Kaufmann (1994).

Finin, T., R. Fritzson, D. McKay, and R. McEntire, KQML as an agent communication language, In *Proc. Third International Conference on Information and Knowledge Management (CIKM'94)*. ACM Press (1994). Online at `http://www.cs.umbc.edu/kqml/papers/kqml_acl.ps`.

Fogel, D. B., *Evolutionary Computation: Toward a New Philosophy of Machine Intelligence*. IEEE Press, Piscataway, NJ (1995).

García-Flores, R., X. Z. Wang, and G. E. Goltz, Agent-based information flow for process industries' supply chain modeling, In *Proc. Process Control and Instrumentation 2000*, Glasgow (2000).

Hall, J. and T. Verne, "RCU optimization in a DCS gives fast payback," *Hydrocarbon Process.*, pages 85–92 (1993).

Hall, J. and T. Verne, "Advanced controls improve operation of Lubes Plant dewaxing unit," *Oil and Gas J.*, pages 25–28 (1995).

Härdle, W., *Applied Non-parametric Regression*. Cambridge University Press (1990).

Harp, S. A., S. Brignone, B. F. Wollenberg, and T. Samad, "SEPIA: A simulator for electric power industry agents," *IEEE Cont. Sys. Mag.*, pages 53–69 (2000).

Hastie, T. J. and R. J. Tibshirani, *Generalized Additive Models*. Chapman & Hall, London (1990).

Hinton, G. E. and S. J. Nowlan, "How learning can guide evolution," *Complex Systems*, **1**, 495–502 (1987).

Kalman, R. E., "A New Approach to Linear Filtering and Prediction Problems," *Trans. ASME, J. Basic Engineering*, pages 35–45 (1960).

Koza, J., *Genetic Programming*. MIT Press, Cambridge, MA (1992).

Kulhavý, R. and P. Ivanova, Memory-based prediction in control and optimisation, In *14th World Congress of IFAC*, volume H, pages 289–294, Beijing, China (1999).

Lu, J. and J. Escarcega, RMPCT: Robust MPC technology simplifies APC, Presented at AIChE Spring Meeting, Houston, TX (1997).

Lu, J., Multi-zone control under enterprise optimization: Needs, challenges and requirements, In *Proc. International Symposium on Nonlinear MPC: Assessment and Future Directions*, Ascona, Switzerland (1998).

Mitchell, M., *An Introduction to Genetic Algorithms*. MIT Press, Cambridge, MA (1996).

Nath, R., Z. Alzein, R. Pouwer, and M. Leseur, On-line dynamic optimization of an ethylene plant using Profit(r) Optimizer, Paper presented at NPRA Computer Conference, Kansas City, MO (1999).

Peterka, V., Bayesian approach to system identification, In Eykhoff, P., editor, *Trends and Progress in System Identification*, pages 239–304, Elsmford, NY. Pergamon Press (1981).

Prett, D. M. and C. E. Garcìa, *Fundamental Process Control*. Butterworths, Boston (1988).

Samad, T. and J. Weyrauch, editors, *Automation, Control and Complexity: An Integrated View*. John Wiley and Sons, Chichester, UK (2000).

Schaal, S. and C. G. Atkeson, "Robot juggling: An implementation of memory-based learning," *Control Systems Magazine*, **14**, 57–71 (1994).

Sheehan, B. and L. Reid, "Robust controls optimize productivity," *Chemical Engineering* (1997).

Sjöberg, J., Q. Zhang, L. Ljung, A. Benveniste, B. Deylon, P.-Y. Glorennec, H. Hjalmarsson, and A. Juditsky, "Nonlinear black-box modeling in system identification: A unified overview," *Automatica*, **31**, 1691–1724 (1995).

Smith, F. B., An FCCU multivariable predictive control application in a DCS, The 48th Annual Symposium on Instrumentation for the Process Industries (1993).

Verne, T. and J. Escarcega, Multi-unit refinery optimization: Minimum investment, maximum return, In *The Second International Conference and Exhibition on Process Optimization*, pages 1–7, Houston, TX (1998).

Watano, T., K. Tamura, T. Sumiyoshi, and P. Nair, Integration of production, planning, operations and engineering at Idemistu Petrochemical Company, Presented at AIChE Spring Meeting, Houston, TX (1993).

Watkins, C. J. C. H. and P. Dayan, "Q-Learning," *Machine Learning*, **8**, 279–292 (1992).

West, M. and J. Harrison, *Bayesian Forecasting and Dynamic Models*. Springer-Verlag, New York (1989).

Wooldridge, M. and N. R. Jennings, Pitfalls of agent-oriented development, In *Proc. of the 2nd Int. Conf. on Autonomous Agents (AA'98)*, pages 69–76, New York. ACM Press (1998).