8th International IFAC Symposium on
Dynamics and Control of Process Systems
Preprints Vol.1, June 6-8, 2007, Cancún, Mexico

COORDINATING MULTIPLE OPTIMIZATION-BASED CONTROLLERS: NEW OPPORTUNITIES AND CHALLENGES

James B. Rawlings [*,1] **Brett T. Stewart** [*,2]

[*] *Department of Chemical and Biological Engineering*
University of Wisconsin, Madison, WI

Abstract: The status of using many, distributed optimization-based controllers for feedback control of large-scale, dynamic processes is presented and evaluated. We show that modeling the interactions between subsystems and exchanging trajectory information among subsystem model predictive controllers (MPCs) is insufficient to provide even closed-loop stability. The cause of this closed-loop instability is competition between the local agents. We next discuss the cooperative distributed MPC framework, in which the objective functions of the local MPCs are modified to achieve systemwide control objectives. This approach provides guaranteed nominal stability and performance properties, but at the cost of a high degree of communication between the local controllers. We next discuss the issue of taking advantage of the structure of the connections between the subsystems to reduce the required communication. The paper concludes by briefly presenting seven current and unsolved research challenges.
Copyright ©2007 IFAC

Keywords: Distributed control, plantwide control, distributed MPC, large-scale MPC

1. INTRODUCTION

Feedback controller design is foremost a *design* problem and design problems, even engineering design problems, have proven remarkably resistant to precise mathematical formulations with unique solutions. Design problems are complex and messy because what people want out of a design is complex and messy. Some of the goals for the design are hard to define and subjective, some of them may be contradictory; at best, the stated goals are usually incomplete. This does not mean that precise mathematical problems and tools for computing their solutions are not useful. They are incredibly useful. But their application is iterative. Because the design goals are complex and difficult to define precisely, the design process is almost invariably iterative. The designer or design team often proposes a precise, but limited, design problem statement, solves it exactly and optimally and then inspects the outcome, often through numerical evaluation of idealized case studies. Some features are deemed desirable, drawbacks are uncovered, surprises, both good and bad, may be revealed. Inspection of the first attempt is meant to inform the designer about the problem. With this extra knowledge in hand, the designer fashions a better, more comprehensive statement of the design goals and the process continues. At some point the current design is deemed satisfactory and the process of implementation commences. Often the implementation process uncovers other

[1] `rawlings@engr.wisc.edu`. Author to whom correspondence should be addressed
[2] `btstewart@wisc.edu`

kinds of flaws that have escaped detection and the design process is revisited and further iterations are required.

If we examine the history of chemical plant design and operation in the U.S., we see a steady increase in the complexity in the interactions between the various units comprising the overall plant. This increase was driven by the gain in economic efficiency offered by these more complex and interactive plant design and operation strategies. The chemical plants of the 1950s, which could be operated fully in manual mode by a team of human operators, would be economically inviable in today's economy and environment. As complexity of the plant increased, automatic monitoring and control systems became a necessity. Multiloop PID control at the unit level became the first automatic control design. In the last twenty years, however, centralized MPC of small to medium-sized multivariable units has largely replaced multiloop PID in the process industries, particularly for the economically important units in the plant.

It is interesting to note in passing that the undergraduate control curriculum offered in chemical engineering departments has not kept pace with this most recent transformation in industrial practice. Further discussion of the educational challenges that have arisen due to this widespread adoption of MPC in control practice are outside the scope of this paper, but control educators are aware of the problem and are working towards addressing it (Edgar *et al.*, 2006).

We appear now to be at an interesting historical juncture. After moving quickly from manual control, to multiloop, distributed PID control, to centralized, model-based MPC for controlling single and small collections of chemical process units, what is the status for improving dynamic performance at the other end of the spectrum: the large-scale, integrated collections of many of these units that comprise the chemical plant? The complexity and incompletely defined nature of the controller *design* problem again arises. What do we expect of this large-scale controller design? How important is it that parts of the process can be removed and added back to the overall control system. What model size and complexity is allowed? What is the time scale for online controller decision making? How important is it to make an evolutionary transition from the current control technology to the next technology. Is a complete overhaul and redesign allowed or forbidden? How do the operations personnel evaluate the various model and controller maintenance issues that they face? Is a large, centralized control system monolithic and difficult to maintain? Or is it easier to maintain than collections of smaller models?

In this paper, we do not try to answer all of these difficult questions. We assume instead that completely centralized control is not likely to be the method of choice for large-scale problems. Although we have to let future developments speak to the validity of this assumption, there is reason to expect a distributed approach to large-scale problems to remain a popular choice. [3]

So in what respect is the current historical juncture interesting? We have faced the tension between centralized and distributed decision making before, in many different contexts. The move from distributed PID to MPC of small systems was essentially a move towards centralized decision making. This technology gained support because the performance benefit was large. The main theme in this paper is that the current situation is interesting because the local agents are so capable. Imagine trying to coordinate the decision making of a collection of multiloop PID controllers. If the overall system performance is not going well, what are your options to change the behavior? You can modify the local agents by adjusting their three knobs: P, I and D. The impact of turning all of these knobs up and down on the overall system performance is far from obvious. To know, we generally have to do a simulation and a simulation requires starting conditions and gives only a single forecast. We would need many simulations to assess the impact of a change. Choosing which simulations are sufficient to show the entire range of behavior is an unsolved problem. Human intervention and judgment during this learning process is way too slow. We can try larger surgeries: re-configure assignments of measurements with valves as various types of operational problems are encountered. A skillful process control engineer who has experience with a specific process can do remarkable things with this toolset. But centralized decision making based on a multivariable model allows a richer set of actions to be evaluated and produces generally better operation.

Next imagine we wish to coordinate the decision making of a collection of MPC controllers. How can we modify the behavior of these local agents? The situation is strikingly different. Each agent has a specific model of part of the process that it uses to forecast outcomes of its decisions. The model is available for the asking. Each agent has a specific cost function that it optimizes to make its decision. The cost function is available for the asking. Each agent encodes the differences between its forecast and the measurement in a specific way. This model is available for the asking. We argue in this paper that the impact of changing the local agents' models, cost functions,

[3] Consider, for example, a centrally planned economy versus a local, market-driven economy.

and feedback structures can be reliably assessed without extensive online simulation. Because we know precisely what the agent is trying to do and what model it is using to do it, we can predictably and reliably alter its behavior. We do not need to design a centralized supervisor to accomplish this task. The design and maintenance of such a supervisor is no easier than the design and maintenance of the fully centralized controller for the large-scale system.

If we coordinate optimization-based controllers, the local agents have a rich structure that we can modify, with predictable and transparent outcomes, to suit overall performance goals. Coordinating paired PID controllers is problematic because the agents are so limited. Coordinating MPC controllers starts in a strikingly different place. The potential is greater. This paper provides an overview of some of the opportunities and challenges in coordinating MPC controllers.

2. COMMUNICATION AND COOPERATION AMONG SUBSYSTEM CONTROLLERS

We require some terminology and notation to describe the total system, or plant, its decomposition into subsystems, or units, and the components of the controller design problem.

Models. Consider a total system (**plant**) to be comprised of M interconnected subsystems (**units**). Let $(y_i(t), u_i(t))$ be the (p_i, m_i) dimensional vectors of (output, manipulated) variables of the i-th subsystem at time t, in which $i \in \mathbb{I}_M = (1, \dots, M)$. We assume a finite dimensional linear time invariant model is suitable to describe the dynamics between any manipulated input u_i and any measured output y_j

$$\begin{aligned} x_{ij}(k+1) &= A_{ij}x_{ij}(k) + B_{ij}u_j(k) \\ y_i(k) &= \sum_j C_{ij}x_{ij}(k) \qquad i, j \in \mathbb{I}_M \end{aligned}$$

in which integer k is the sample time. The **decentralized model** is the set of M models between u_i and y_i, $i \in \mathbb{I}_M$. The **centralized model** is the set of M^2 models between all inputs and outputs: u_i and y_j, $i, j \in \mathbb{I}_M$. The **interaction model** for subsystem i is the set of $M-1$ models between the inputs of other subsystems and outputs of subsystem i: u_j and y_i $j \in \mathbb{I}_M, j \neq i$. The centralized model is the union of the decentralized model and the M sets of interaction models.

Objectives. We assume the objective function of each subsystem can be expressed as a quadratic function of that subsystem's input and outputs evaluated at discrete sample times in a prediction horizon

$$\Phi_i = \sum_{l=1}^{N} L_i(y_i(k+l|k), u_i(k+l|k))$$

in which L is the stage cost, $L_i(y, u) = y'Q_iy + u'R_iu$, $Q_i, R_i > 0$, and N is the forecast horizon. The forecast of outputs is computed from the forecast of inputs (the decision variables) under different model assumptions described below. We assume a suitable objective function for the total system is a convex combination of the subsystem objective functions

$$\Phi = \sum_{i \in \mathbb{I}_M} w_i\Phi_i \qquad \sum_{i \in \mathbb{I}_M} w_i = 1 \qquad w_i > 0 \quad (1)$$

Communication. For the distributed controllers, each control problem is solved by a local subsystem controller. We also consider solving these control problems iteratively with a communication strategy between the iterations of the subsystem controllers. Because an MPC optimization provides a *trajectory* of inputs and not just the current input, we communicate a trajectory of inputs at each iteration. The iteration of decision variables is defined as

$$\boldsymbol{u}_i^p = w_i\boldsymbol{u}_i^{*(p)} + (1-w_i)\boldsymbol{u}_i^{p-1} \qquad i \in \mathbb{I}_M$$

At each iteration, p, the trajectory of inputs is a convex combination of the current local optimal solution and the previous iteration. After the iteration converges or the available computation limit is reached, the first input in the trajectory is injected into the system and the next measurement is obtained.

Controller design. We consider the following four controller design choices (Venkat *et al.*, 2006*a*). An **agent** refers to any optimizer working at the subsystem level.

(1) **Centralized control.** Single controller. The model is the centralized model of the total system and the objective function is the objective function of the total system. Achieves optimal nominal control performance.
(2) **Decentralized control.** M controllers. Each model is the local subsystem model. Each objective function is the local subsystem objective function. Design ignores all interactions between units.
(3) **Communication-based control.** M controllers. Each model is the local subsystem model plus interaction models. Each objective function is the local objective function. For controller i at iterate p, forecast of inputs from controllers $j \neq i$ at iterate $p-1$ are available.
(4) **Feasible-cooperative control.** M controllers. Each model is the local unit model plus interaction models. Each objective function is a copy of the total objective function. For controller i at iterate p, forecast of inputs

from controllers $j \neq i$ at iterate $p-1$ are available.

3. GOOD CONTROLLERS GONE BAD — DENSELY CONNECTED SYSTEMS

The centralized controller and decentralized controller define two limiting design extremes. Centralized control accounts for all possible interactions, large and small, whereas decentralized control ignores them completely. In decentralized control the local agents have no knowledge of each others' actions. It is well known that the nominal closed-loop system behavior under decentralized control can be arbitrarily poor (unstable) if the system interactions are not small. The following reviews provide general discussion of this and other performance issues involving decentralized control (Siljak, 1991; Lunze, 1992; Larsson and Skogestad, 2000; Cui and Jacobsen, 2002).

The next level up in complexity from decentralized control is communication-based control. In this framework, the agents have interaction models and communicate at each iteration (Jia and Krogh, 2002; Motee and Sayyar-Rodsari, 2003; Dunbar and Murray, 2006). The big advantage of communication-based control over decentralized control is that the agents have accurate knowledge of the effects of all other agents on their local objectives. The basic issue to analyze and understand in this setup is the competition between the agents. That analysis is the subject of noncooperative game theory (Başar and Olsder, 1999). To illustrate some of the issues, consider a simple two-subsystem problem with scalar inputs and outputs and forecast horizon of $N = 1$. Figure 1 shows the possible behavior. The Pareto optimal solution is defined by combining the two local objectives with equal weight. The Pareto optimal solution is the one found by a centralized controller. The Nash equilibrium is defined as a point satisfying the optimality conditions for each local agent. In the top figure, iterating the two local controllers under communication converges to the stable Nash equilibrium, which is near the Pareto optimal solution. Communication-based control is likely to provide good closed-loop system behavior in this scenario. By changing the cost functions, we create the middle figure in which the Nash equilibrium is far from the Pareto solution. The converged solution obtained using a communication-based strategy is far from optimal, and the closed-loop system using this controller may not be even stable. Finally, the bottom figure shows a case in which the Nash equilibrium is close to the Pareto solution but the Nash equilibrium is not stable. In this case, the communication-based iterates do not converge to the Nash equilibrium.

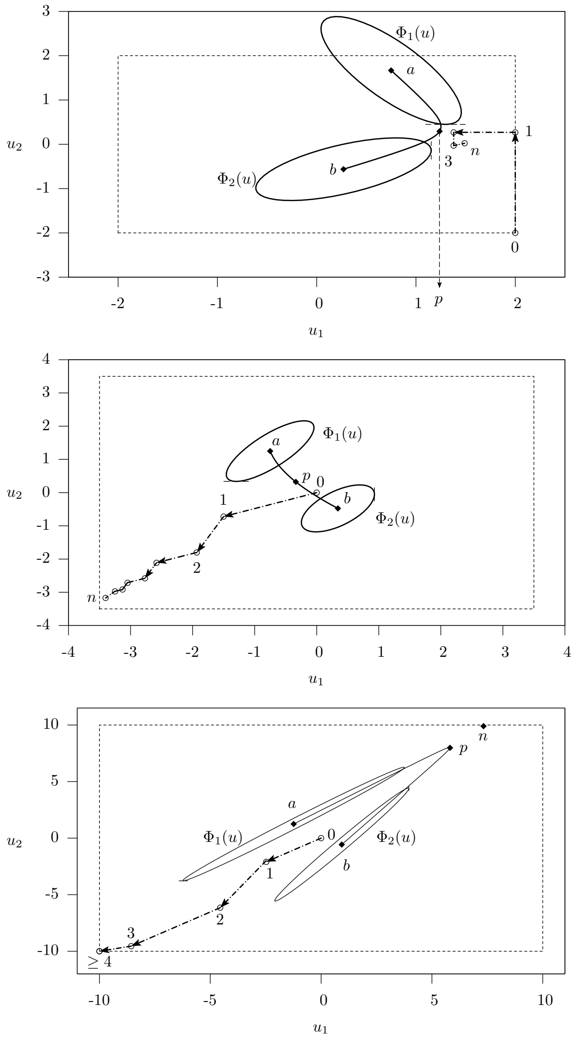

Fig. 1. Top: Nash equilibrium is stable and near the Pareto optimal solution. Middle: Nash equilibrium is stable but not near the Pareto optimal solution. Bottom: Nash equilibrium is unstable.

The iterates converge to a point on the boundary of the feasible region, which is far from the Pareto solution. Again, closed-loop instability of the communication-based control system is likely. Note that none of the undesirable behavior is caused by a lack of knowledge about the overall system. All agents have complete information about the effects of all the other agents' actions.

Therefore, if the overall system is composed of strongly interacting subsystems, closed-loop instability of decentralized and communication-based control seems unavoidable. One alternative to ensure closed-loop stability, of course, is to use a single centralized controller. But there are other alternatives. We can maintain the distributed structure of the M local controllers, but change the objective functions so that the local agents cooperate. Changing the cost function is a simple matter of rewriting the data in the local agents QP subproblems. In fact, the data for var-

	Dist. Col.		React/Sep.		OL Unstable	
	Λ_{cost}	$\Delta\Lambda_{\text{cost}}\%$	Λ_{cost}	$\Delta\Lambda_{\text{cost}}\%$	Λ_{cost}	$\Delta\Lambda_{\text{cost}}\%$
Cent-MPC	1.72	0	2.0	0	1.78	0
Decent-MPC	∞	∞	∞	∞	3.53	98.3
Comm-MPC	∞	∞	∞	∞	3.53	98.2
FC-MPC (1 iterate)	6.35	269.2	2.13	6	2.03	13.9
FC-MPC (10 iterates)	1.74	1.32	2.01	0.09	1.8	0.8

Table 1. Closed-loop performance comparison of centralized MPC (Cent-MPC), decentralized MPC (Decent-MPC), communication-based MPC (Comm-MPC) and FC-MPC.

ious levels of cooperation, varying from decentralized control to fully cooperative control, can be stored so that different control scenarios can be loaded and implemented. This approach provides an evolutionary path from a current decentralized technology to something approaching centralized control, but without removing the local control structure that may already be in place.

Venkat (Venkat, 2006) studied a number of small typical chemical process examples and found it is rather easy to generate closed-loop instability for decentralized and communication-based control systems. Table 1 summarizes some of these results. The examples are: (i) a 2×2 transfer function of a distillation column, (ii) a reactor/separator obtained by linearizing a nonlinear, fundamental model at a desired steady state, and (iii) a an open-loop unstable mathematical example. Decentralized and communication-based MPC are unstable for the first two examples and increase the cost compared to centralized MPC by almost 100% in the third example. Cooperative MPC is always closed-loop stable. The nominal performance of the closed-loop system improves with the number of iterations of the local controllers and converges to the centralized solution. The rate of convergence to centralized performance depends strongly on the type of subsystem interactions – for example, the distillation column requires more iterations than the reactor/separator or the unstable system.

Properties of feasible-cooperative MPC. The following properties of cooperative MPC have been established (Venkat *et al.*, 2006*a*; Venkat *et al.*, 2006*b*).

(1) The iterations generated by the cooperative MPC algorithm are systemwide feasible.
(2) Control based on any intermediate termination of the algorithm provides nominal closed-loop stability and zero steady-state offset.
(3) If iterated to convergence, the distributed MPC algorithm achieves optimal, centralized MPC control.
(4) To handle output instead of state feedback, a distributed estimator design strategy can be implemented, in which each estimator is stable and uses only local measurements to estimate subsystem states. The combined distributed estimator-distributed regulator is feasible and closed-loop stable for all iteration numbers in the case of decaying estimate error.

4. TOPOLOGY OF TYPICAL CHEMICAL PROCESSES

In Section 2 an MPC cooperation strategy with guaranteed performance was summarized. Yet, while ensuring stability and centralized-like behavior, it requires a completely connected communication strategy. Every agent in the plant communicates with all the others. This level of communication makes sense because each subsystem in the plant, in open loop, may affect all the others. This communication is *not desirable*, however, because it requires the final unit to communicate with the initial unit, even if these processes are connected only through many intermediate units. This drawback motivates developing a strategy in which closed-loop stability, at least, is guaranteed but unnecessary communication is eliminated. In this section we focus on the characterization of such a strategy.

In a typical chemical process, subsystems are connected through material, energy, and information flows. These flows generally pass from subsystem to subsystem, so that each subsystem directly interacts only with its nearest neighbors. Interactions beyond nearest neighbors occur through intermediate subsystems. Therefore, non-nearest

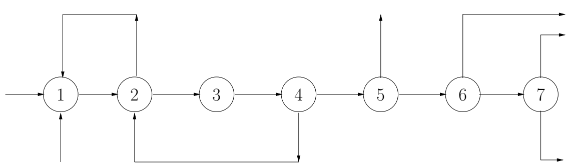

Fig. 2. Ethylene glycol flowsheet. 1. Feed tank 2. Preheater 3. Reactor 4. Evaporators 5. Light end columns 6. Mono ethylene glycol column 7. Higher glycol recovery

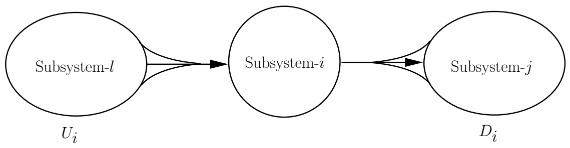

Fig. 3. The set U_i is the set of *upstream* subsystems and the set D_i is the set of *downstream* subsystems. Subsystem i is linked to all other subsystems in the flowsheet by the upstream subsystems $l \in U_i$ and the downstream subsystems $j \in D_i$.

neighbor interactions may be modeled as the product of multiple nearest neighbor interactions in series. Also, in many sections of the plant, a given subsystem directly interacts only with its downstream subsystems. For example, in Figure 2, subsystem 1 affects subsystem 2 directly, but affects subsystem 4 by way of subsystems 2 and - 3. This topology may be exploited to reduce the amount of communication required for cooperative control.

Given the structure above, a different interaction model can be derived. This model assumes the states of a given subsystem are a function of only the subsystem's states and inputs and the states of the upstream subsystems as in Figure 3. Specifically,

$$x_i(k+1) = A_{ii}x_i(k) + \sum_{l \in U_i} A_{il}x_l(k) + B_i u_i(k) \quad (2)$$

in which $U_i \subseteq \mathbb{I}_M$ is the set of nearest neighbor subsystems *upstream* of subsystem i. This model implies a fundamentally different set of interactions than used in Section 2 in which input-to-state interactions are considered. This model instead considers state-to-state interactions. These interaction models are equivalent however. The equivalence is revealed by recursively substituting for all $x_l(k)$ into $x_i(k+1)$, and recovering a strict input-to-state model. More importantly, by considering this state-to-state model, communication is reduced. Defining the set U_i also defines the set of subsystems that directly affect subsystem i. Typically U_i is a subset of the entire plant's subsystems, which reduces the interactions that must be considered.

In MPC, each optimizer is given the task of predicting the effect of its inputs on a given objective. In cooperative MPC, this objective is the performance of the entire plant Φ. As in Section 2 the entire plant's objective is the convex sum of the subsystems' objective Φ_i. Using Equation (2) to predict how the inputs of subsystem i affect the plant objective, the summation in (1) is over only i and the subsystems downstream from i

$$\phi_i = \sum_{r \in \{i, D_i\}} w_r \Phi_r \quad (3)$$

in which ϕ_i is the reduced centralized objective for subsystem i and D_i is the set of nearest neighbor subsystems *downstream* from subsystem i. The reduction of terms in the summation follows from examining each term. If no decision variables appear in a term of the summation, then this term is constant and does not affect the solution. Using (2) for the prediction of the plant objective, the decision variable $u_i(k)$ appears in (1) only in the ith term and jth terms such that $i \in U_j$. The latter may be restated as $j \in D_i$. After substituting (2) into (3) the MPC subproblem is obtained. The controller for subsystem i must consider the *state* trajectories of the upstream subsystems U_i and the *input* trajectories of the downstream subsystems D_i. Qualitatively, in order to make an optimal input decision, subsystem i must know where the upstream subsystems are going and be able to forecast the downstream subsystems. But notice it does not need a forecast of any other subsystem. This behavior is acceptable because the other subsystems make these forecasts, and account for these non-nearest neighbor interactions.

This reduction of communication does not come for free, however. The MPC optimization returns an optimal trajectory of inputs. The states, which are communicated after local optimizations, must be synchronized with the optimal inputs to obtain the optimal state trajectory. In general, these states cannot be obtained using the local model. The trouble arises in the presence of recycle. Consider the ethylene glycol plant in Figure 2 in which subsystems 1 and 2 are involved in a recycle. According to Equation (2), the states of subsystem 1 affect subsystem 2, but at the same time the states of subsystem 2 affect subsystem 1. These states must then be obtained *simultaneously.* Therefore an extra step, and extra time, is needed to iteratively synchronize the states, or one of the MPCs solves for the states of both systems. Adding time to the optimization is never a good idea, however, and, in a distributed optimization, allowing one of the subsystems to solve for all states is not much different than centralized control.

Consider the special case without recycle. In this case, material flows only downstream and

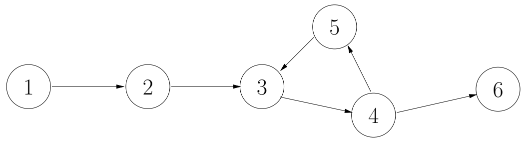

Fig. 4. Periodic recycle. Flow is mostly in the direction from the first subsystem to the last with local recycle occurring periodically throughout the flowsheet.

the problem of downstream subsystems affecting upstream subsystems is avoided. This property yields an important effect: the synchronization step can be solved locally on each subsystem. To synchronize, subsystem i must know only the states, already available from the optimization step, of the subsystems upstream. Therefore no extra communication is needed. The reduced communication requirement in distributed MPC in this case is analogous to the relative gain array (RGA) becoming a diagonal matrix for a triangular matrix of transfer functions (Ogunnaike and Ray, 1994, p.737).

A typical plant has many recycle streams, and the recycle streams may be difficult to identify, as in the ethylene glycol example in Figure 2. How can communication be reduced in this situation? We propose a hybrid strategy. Total communication, as in Section 2, is implemented between subsystems involved in the recycle while reduced communication, described in this section, is used between subsystems not in the recycle. The total communication and reduced communication areas exchange information and iterate in parallel. For example, in Figure 4, subsystems 3–5 are involved in recycle, so full communication is used between these subsystems. Between all others, reduced communication may be used. Similar techniques are used in the optimal matching and shrinking algorithms in graph reduction (Cook *et al.*, 1998, pp.127–198) and in the analysis of electrical networks (Chan, 1969, pp.263–280). The analysis of this simple example is straightforward, but more complicated flowsheet topologies may require general and sophisticated tools. In Figure 2, for example, must subsystem 1 communicate with subsystem 4 directly, or is it sufficient to communicate only with subsystem 2?

Consider the extremes of the recycle problem. Figure 4 represents the most basic recycle, in which flow moves, on average, downstream with a recycle occurring occasionally. This periodic recycle may be controlled via the hybrid strategy outlined above. Figure 5 is the other extreme of recycle. Subsystem 1 is affected directly by subsystem 6, the final unit in the flowsheet. Must all subsystems communicate in this example or is there a more elegant way to reduce communication? In general, a plant is a combination of these recycle extremes.

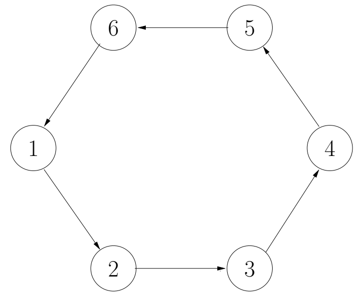

Fig. 5. Total recycle. The last subsystem in the process flows into first.

The next step in this line of research is to derive a framework for identifying these recycles and specifying the necessary communication strategies to handle them.

5. FUTURE CHALLENGES

The following issues represent open research challenges. Progress on any of these issues will likely further the development of a more comprehensive and reliable controller design strategy suitable for the large-scale, and challenging applications faced by practitioners.

Exploiting the structures in application domains. As a research community, we have just begun to think about the types of structure arising in applications. The topological connections of chemical processes described in Section 4 is one such example. Recognizing and exploiting these kinds of structures may prove critical to success in many fields.

Coupled input constraints. The FC-MPC algorithm converges to the *optimal centralized* solution when the input constraints are uncoupled between subsystems. For input constraints *coupled between subsystems*, nominal closed-loop stability is still guaranteed, but performance remains an open question. At present we do not have any bounds on the degree of suboptimality that can be caused by coupled input constraints.

Communication disruption. The goal of a distributed control design system is to maintain a high level of overall system performance with as little communication as possible between the subsystems. Reliable strategies are required for handling possible disruptions and delays in the communication of input trajectories among subsystems. It is expected that the closed-loop system can be destabilized with incorrect input trajectory information due to information loss or delays. It would be desirable to have lower performance backup strategies available if communication fails completely. A plant's previous decentralized control system may be suitable for this task, but other options may be investigated such as those considered in recent work on control over networks (Baliga and Kumar, 2005; Casavola *et al.*, 2006; Imer *et al.*, 2004).

Closed-loop Identification of subsystem interactions. The distributed MPC framework requires both subsystem models and models of the important interactions between subsystems. While closed-loop identification is a well-studied field, tailoring the existing techniques for distributed MPC is a relatively recent research area (Gudi and Rawlings, 2006). Improvements in techniques for closed-loop identification for distributed MPC will likely prove critical for practical implementation. Reliable integration of the algorithm used for closed-loop identification with the algorithm for distributed MPC may have significant impact in the process industries.

Robustness to model errors. Handling uncertainty in the controller model remains a key issue that needs to be addressed. Interaction models are typically identified using closed-loop operating data. Typically, for small plant-model mismatch, the feedback in standard MPC is adequate to obtain good closed-loop performance. When the plant-model mismatch is more significant, robust distributed MPC design may be necessary. With this issue in mind, a thorough investigation into robustness theory for distributed MPC needs to be undertaken. Establishing properties such as robust feasibility and stability in the distributed MPC setting could prove to be both useful and interesting. Construction of disturbance invariant sets (Kolmanovsky and Gilbert, 1998; Rakovic *et al.*, 2004) for each subsystem could prove useful to establish robust stability.

Partial cooperation. Partial cooperation is a method that allows the control designer to enforce limitations on the ways optimization-based controllers use their available inputs to meet performance specifications (Venkat *et al.*, 2005). It has the nice side benefit of reducing communication among subsystem controllers. But closed-loop properties for the partial cooperation framework are not available. Techniques for further reducing communication among subsystems, without compromising closed-loop stability, should be investigated.

Time-scale separation and fast subsystems. To implement cooperative distributed MPC for systems with fast sampling rates, one may require techniques that allow a quick evaluation of the MPC optimization problem. The possibility of employing explicit MPC techniques (Bemporad and Filippi, 2003; Bemporad *et al.*, 2002; Pannocchia *et al.*, 2006; Tondel *et al.*, 2003) for distributed MPC should be investigated. One complication in distributed MPC is that the input trajectories for interconnected subsystems' MPCs are additional parameters for each MPC optimization problem. The dimensionality of the parameter space consequently, is much greater in distributed MPC.

Zero-order holds have been the method of choice in centralized MPC applications. First-order holds offer many advantages for fast systems. For the same control performance, a first order hold allows a larger sampling time than a zero order hold; this feature by itself allows further iteration of the distributed MPC system. At this time we know little about the advantages and disadvantages of first-order holds in asynchronous, distributed MPC.

6. CONCLUSION

Given the long introduction, the conclusion can be brief. We have argued that coordinating optimization-based controllers offers interesting capabilities for tackling control system design for large-scale processes. Optimization-based local agents operate at a high level. By modifying their objective functions, the agents cooperate. By modifying their interaction models, the agents are aware of the impact of their decisions on other parts of the overall system. By taking advantage of the process topology, we can reduce the required communication between the agents. There is no free lunch. If the system is highly interactive, it requires a high degree of communication among the local agents, or a centralized controller. But a highly interactive overall system is unlikely to be the rule in large-scale chemical processes. Chemical process design does not typically produce this kind of highly interactive overall system. Material, energy and information generally flows sequentially from unit to unit with smaller levels of recycle and integration providing the major interactive coupling. Because controller design for large-scale systems is a complex design problem, many interesting research issues remain. These include: exploiting system structure, handling input constraint coupling and disruption in communication, identifying interaction models online, ensuring robustness to model errors, imposing controller restrictions, and treating time-scale separation.

Further research in this field will benefit greatly from close contact between theory and practice. The major downside risk to avoid is producing elegant controller design methodologies that no one uses. Because this line of research is motivated by practical needs, close collaboration between researchers and practitioners is a prerequisite for evaluating progress and defining new research opportunities.

REFERENCES

Başar, Tamer and Geert Jan Olsder (1999). *Dynamic Noncooperative Game Theory.* SIAM. Philadelphia.

Baliga, Girish and P. R. Kumar (2005). A middleware for control over networks. In: *Proceedings of the Joint 44th IEEE Conference on Decision and Control and European Control Conference.* Seville, Spain. pp. 482–487.

Bemporad, A. and C. Filippi (2003). Suboptimal explicit receding horizon control via approximate multiparametric quadratic programming. *IEEE Trans. Auto. Cont.* **117**(1), 9–38.

Bemporad, A., M. Morari, V. Dua and E.N. Pistikopoulos (2002). The explicit linear quadratic regulator for constrained systems. *Automatica* **38**(1), 3–20.

Casavola, Alessandro, Maurizio Papini and Giuseppe Franzé (2006). Supervision of networked dynamical systems under coordination constraints. *IEEE Trans. Auto. Cont.* **51**(3), 421–437.

Chan, Shu-Park (1969). *Introductory Topological Analysis of Electrical Networks.* Holt, Rinehart and Winston Series in Electrical Engineering, Electronics, and Systems. Holt, Rinehart and Winston, Inc.. New York.

Cook, William J., William H. Cunningham, William R. Pulleyblank and Alexander Schrijver (1998). *Combinatorial Optimization.* Wiley-Interscience Series in Discrete Mathematics and Optimization. Wiley-Interscience. New York.

Cui, H. and E.W. Jacobsen (2002). Performance limitations in decentralized control. *J. Proc. Cont.* **12**, 485–494.

Dunbar, William B. and Richard M. Murray (2006). Distributed receding horizon control with application to multi-vehicle formation stabilization. *Automatica* **2**(4), 549–558.

Edgar, Thomas F., Babatunde A. Ogunnaike, James J. Downs, Kenneth R. Muske and B. Wayne Bequette (2006). Renovating the undergraduate process control course. *Comput. Chem. Eng.* **30**, 1749–1762.

Gudi, Ravindra D. and James B. Rawlings (2006). Identification for Decentralized Model Predictive Control. *AIChE J.* **52**(6), 2198–2210.

Imer, Orhan C., Serder Yuksel and Tamer Basar (2004). Optimal control of dynamical systems over unreliable communication links. In: *IFAC Symposium on Nonlinear Control Systems, Stuttgart, Germany.* pp. 1521–1524.

Jia, Dong and Bruce H. Krogh (2002). Min-max feedback model predictive control for distributed control with communication. In: *Proceedings of the American Control Conference.* Anchorage,Alaska. pp. 4507–4512.

Kolmanovsky, Ilya and Elmer G. Gilbert (1998). Theory and computation of disturbance invariant sets for discrete-time linear systems. *Mathematical Problems in Engineering* **4**(4), 317–367.

Larsson, Truls and Sigurd Skogestad (2000). Plantwide control- A review and a new design procedure. *Mod. Ident. Control* **21**(4), 209–240.

Lunze, J. (1992). *Feedback Control of Large Scale Systems.* Prentice-Hall. London, U.K.

Motee, Nader and Bijan Sayyar-Rodsari (2003). Optimal partitioning in distributed model predictive control. In: *Proceedings of the American Control Conference.* Denver,Colorado. pp. 5300–5305.

Ogunnaike, Babatunde A. and W. Harmon Ray (1994). *Process Dynamics, Modeling, and Control.* Oxford University Press. New York.

Pannocchia, Gabriele, James B. Rawlings and Stephen J. Wright (2006). Fast, large-scale model predictive control by partial enumeration. Accepted for publication in *Automatica.*

Rakovic, S. V., E. C. Kerrigan, K. I. Kouramas and D. Q. Mayne (2004). Invariant approximations of robustly positively invariant sets for constrained linear discrete-time systems subject to bounded disturbances. Technical report. Department of Engineering, University of Cambridge. Cambridge, UK. CUED/F-INFENG/TR.473.

Siljak, Dragoslav. D. (1991). *Decentralized Control of Complex Systems.* Academic Press. London.

Tondel, P, T.A. Johansen and A Bemporad (2003). An algorithm for multi-parametric quadratic programming and explicit MPC solutions. *Automatica* **39**(3), 489–497.

Venkat, Aswin N. (2006). Distributed Model Predictive Control: Theory and Applications. PhD thesis. University of Wisconsin-Madison.

Venkat, Aswin N., James B. Rawlings and Stephen J. Wright (2005). Distributed model predictive control of large-scale systems. Assessment and Future Directions of Nonlinear Model Predictive Control, Freudenstadt-Lauterbad, Germany.

Venkat, Aswin N., James B. Rawlings and Stephen J. Wright (2006*a*). Stability and optimality of distributed, linear model predictive control. part 1: state feedback. Submitted to *Automatica.*

Venkat, Aswin N., James B. Rawlings and Stephen J. Wright (2006*b*). Stability and optimality of distributed, linear model predictive control. part 2: output feedback. Submitted to *Automatica.*

ACKNOWLEDGMENT

The authors would like to thank Dr. A.N. Venkat and Professors S.J. Wright and I.A. Hiskens for helpful discussion of these ideas. The authors gratefully acknowledge the financial support of the NSF through grant #CTS-0456694, and the industrial members of the Texas-Wisconsin Modeling and Control Consortium.