SIMPLE IMPLEMENTATION OF DYNAMIC OPTIMAL OPERATION

Sridharakumar Narasimhan [*], **Sigurd Skogestad** [*,1]

[*] *Dept of Chemical Engineering, Norwegian University of Science and Technology (NTNU), Trondheim, Norway*

Abstract: The computational effort involved in solution of real-time optimization problems can be very demanding. Hence, simple but effective implementation of optimal policies are attractive. A formal definition of simplicity is presented for a certain class of systems. Some Heat Exchanger Network problems can be formulated as Linear Programs. For this class of problems, a simple and effective implementation of optimal operation policy that avoids the need for online optimization is described.

Keywords: Optimality, self-optimizing control, offline optimization, control structure design

1. INTRODUCTION

Optimal operation of a chemical process can generally be formulated as a dynamic optimization problem. Let $x \in \mathbb{R}^{nx}$ be the state variables, $u \in \mathbb{R}^{nu}$ the set of available manipulations and $d \in \mathbb{R}^{nd}$ denote the set of disturbances affecting the plant. Let $J(x,u,d)$ be an economic objective that is to be minimized. The dynamic optimization problem that we seek to solve is:

$$\min_u J(x,u,d)$$

$$\text{subject to } \left.\begin{aligned} \dot{x} &= f(x,u,d), \\ h(x,u,d) &= 0, \end{aligned}\right\} \tag{1}$$

$$g(x,u,d) \leq 0, \tag{2}$$

where (1) represents the model and (2) represents the process design constraints. The solution and resulting centralized implementation of this problem is very complex and is seldom used in practice. Hence, a number of simplifications are needed. The first simplification is to use time scale separation. In a typical complex continuous chemical process, the centralized optimizing controller can be conceptually broken down into several layers, for eg., into regulatory control, supervisory control and optimization layer with the layers arranged in an increasing level of hierarchy (Refer Fig 1). Generally, the interaction between the layers is such that at the optimization layer, we try to determine optimal values of the setpoints or reference values by optimizing an economic cost function and the control layers are devoted to ensuring that the setpoints are maintained (Findeisen *et al.*, 1980). The efficient vertical decomposition between the optimization and control layers requires that the economic objective is determined largely by the slow time scale. In the simplest case, steady state optimization may be sufficient. It also requires that one can identify con-

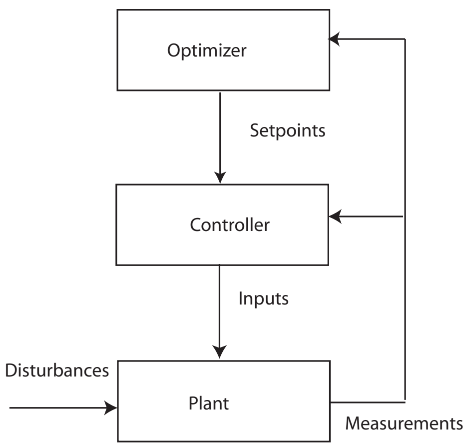

Fig. 1. Hierarchial decomposition

[1] Corresponding author. Email: skoge@chemeng.ntnu.no

trolled variables and set-points for the same that need to be changed infrequently in spite of disturbances and noise on the fast time scales. This is the concept of self-optimizing control where we try to achieve an acceptable loss with constant setpoint values for the controlled variables (Skogestad, 2004*b*). More generally, the idea is to use offline optimization to generate optimal trajectories, choose appropriate controlled variables such that disturbances and other changes are handled by local feedback loops.

In addition to the above ideas, one also attempts to look for other simplifications, viz., in terms of implementing the optimal solution. The focus of this contribution is to derive simple, but effective implementation of the optimal policies by primarily exploiting the known properties of the optimal solution. We adopt the position that in principle, offline calculations can be performed easily and online computations need to be minimized. Thus, offline optimization and analysis should be used to simplify the online implementation and make it robust. It may not be known *a priori* if such a simplification is possible and hence it is natural to analyze classes of systems. A similar approach is used by researchers in operations research or algorithms where the properties of particular systems are used to develop specific tailor-made algorithms or solution procedures. However, the difference is that our focus is online implementation. In particular, we seek to characterize the properties of the optimal solution offline and use this characterization to simplify the implementation of the optimal solution by minimizing online calculations and using feedback.

These ideas are further motivated in Section 2 with examples and developed further for a specific class of linear systems in subsequent sections.

2. STRUCTURE OF THE OPTIMAL SOLUTION

Given an abstraction of the optimization problem that we seek to solve (1-2), an optimal solution satisfies certain properties. For example the steady state and the more general dynamic version of the optimization problem require the KKT conditions and the Pontryagin Minimum Principle to be satisfied respectively. We shall discuss the structure of the optimal solution for certain classes of systems by drawing on some examples.

It is well known that the optimal solution to an infinite time dynamic optimization problem with a quadratic performance objective and linear dynamic model can be expressed as a time invariant feedback law: $u = -Kx$ (Kalman, 1963; Bryson, 1999). In other words, the optimal solution to a dynamic optimization problem has a simple feedback solution that can be implemented effectively and also has the additional property of robustness inherent to feedback systems. Recently, this idea has been extended to control of linear constrained systems (Pistikopoulos *et al.*, 2002). The authors show that the control law is a piecewise continuous linear function of the states for the finite horizon problem (model predictive control) and the infinite time horizon problem (constrained linear quadratic regulation). The optimal solution is explicitly calculated offline. A different example is that of elevator dispatch during uppeak traffic (Pepyne, 1997), where the authors show that the structure of the optimal dispatching policy that minimizes the average or discounted waiting time is a threshold based policy. The determination of the constant gain K (in the optimal control problem) or the thresholds (in the elevator despatch problem described above) is non-trivial. However, these essentially involve off-line calculations and thus are in consonance with the requirements specified previously.

Another example is the problem of maximizing throughput in a network. Under certain conditions, optimal operation of the plant is equivalent to maximizing throughput. The solution to the problem of determining maximum flow in a flow network has the property that maximal amount of a flow is equal to the capacity of a minimal cut (Nemhauser and Wolsey, 1999). Hence, the maximum flow is limited by a bottleneck and optimal operation can be achieved by focussing on the bottleneck. This results in the following implementation policy: identify the bottleneck and maintain maximum flow at the bottleneck (Aske *et al.*, 2006). The authors have described how this can be achieved by a control hierarchy using a coordinator MPC at the top and several local MPCs at the lower levels.

In the following sections, we give a formal definition of simplicity for a certain class of systems and give examples of a simple implementation for a class of systems that can be formulated as linear programs.

3. WHAT IS A SIMPLE STRATEGY?

Since the primary aim of this contribution is to demonstrate simple but effective implementation of an optimal solution, we would like to define simplicity. As a general definition would be difficult, we consider a special case of switching between active constraints.

At the optimal solution, a subset of the inequality constraints are active. The set of constraints that are active at the optimal solution could form a subset of the controlled variables (Arkun and Stephanopoulos, 1980). When the optimal solution changes due the effect of disturbances, the sign of the Lagrange multipliers can be used to determine new search directions to move the plant operation towards optimality (Arkun and Stephanopoulos, 1980). If the model equations, constraints and objective are linear (Linear Program), it is well known that the optimal solution (if it exists) is at a vertex of the simplex. Hence, the available degrees of freedom can be used to control the active constraints. Under the effect of disturbances, the optimal vertex can move and and it is possible that the set of active constraints change. Hence implementation of the optimal policy is aided if we are able to characterize the set of possible optimal vertices. It is possible to determine the set of optimal vertices and the cost functions as functions of these parameters using parametric programming (Gal, 1979) or explicitly solving the linear program. It is possible to perform these calculations offline. Since the focus is on simple implementation, offline computations (even extensive) may be preferred to online computation. A similar approach is adopted by (Pistikopoulos *et al.*, 2002) where the solution to the dynamic MPC optimization problem is computed offline using parametric programming.

As a simple example, consider the following scenario with 3 manipulations u_1, u_2, u_3 and 2 controlled variables y_1 and y_2. We assume that the manipulations are bounded below. Characterizing the optimal solution in different possible operating regimes gives rise to the following table, where A and I denote that the corresponding manipulations are saturated and not saturated respectively. In Region 1, u_1 has saturated to

Table 1. Motivating example

Region	u_1	u_2	u_3
1	A	I	I
2	I	A	I

its lower constraint and so it has to be given up as a manipulating variable. u_2 and u_3 are however within the feasible region and so can be used to control y_1 and y_2 using an appropriate feedback control structure. In Region 2, u_2 has saturated and so u_1 and u_3 can be used to control y_1 and y_2.

Developing this idea further, we seek to characterize the optimal solution by decomposing the operating region into smaller regions. In each region, we seek to control set of variables through an appropriate feedback control structure. The effect of the disturbances is to change the optimal value of the cost index J and/or the set of constraints that are active at the solution. It must be noted that it may not be possible to arrive at such a decomposition for all systems. For such a decomposition, a strategy D is defined as:

$$D = \{(U_1, Y_1, r_1, c_1), (U_2, Y_2, r_2, c_2), \ldots, (U_k, Y_k, r_k, c_k)\}, \tag{3}$$

where U_i is the set of indices of manipulated variables, Y_i is the set of indices of controlled variables, r_i is the set of reference trajectories for the controlled variables in Y_i and $c_i = c_i(U_i, Y_i, r_i)$ is the complexity of the controller i which could be a function of the structural complexity (Skogestad, 2004*a*) and computational complexity. Intuitively, the relationship between controller complexity and performance can be described as in Fig. 2. Controller performance at point P is maximized. However, if the loss in performance at Q is acceptable, it may be preferred over P as the controller complexity of Q is lower. Point R (and indeed all points on the shaded region) is never preferred

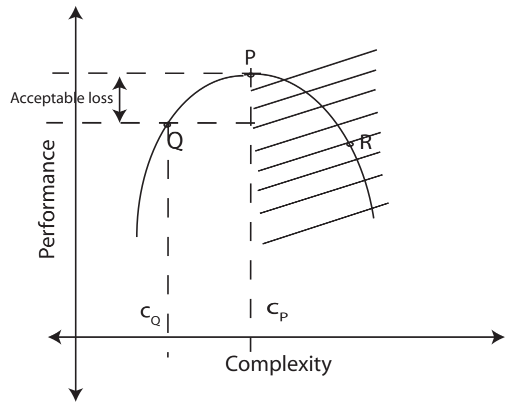

Fig. 2. Performance- complexity tradeoffs

as it is inferior to P in both controller performance and complexity. Thus, it is clear that there is an inherent trade-off between controller performance and complexity.

Having computed these tuples offline, an online switching mechanism may be used to switch between the different control laws. Associated with the switching between two neighbouring regimes i, j is the cost of switching c_{ij}^s. The complexity index of such a strategy is then defined as:

$$S = \sum_i c_i + \sum_i \sum_{j,\ j \neq i} c_{ij}^s. \tag{4}$$

A strategy D_1 is preferable to another strategy D_2 if the complexity index of D_1 is lower than that of D_2. Although it can be quite difficult to determine the quantities described above, the above definition suggests us as what to look for in a simple implementation. For example, we could require that the controllers and the switching logic or process be easily implementable, the measurement errors and implementation errors in the control be small etc.

In the context of the linear programs, the strategy essentially would be to move from one vertex, which is characterized by the set of active constraints to another. This is trivially possible if all disturbances are measured, which might not possible in practice. Another possibility is to estimate the disturbances online using parameter estimation techniques or estimators or soft sensors. However, for such an implementation, (4), c_i would be very high. Since the focus is on simple implementation, such a strategy would not be preferable and hence, we seek implementations with low c_i, but acceptable levels of performance.

Having lowered c_i, the next step would be to choose an appropriate switching technique that results in low c_{ij}^s. An example of a simple constraint switching technique is split range control where, when a manipulation saturates another manipulation is used. More detailed information on implementation issues can be found in (Lersbamrungsuk *et al.*, 2006; Glemmestad, 1997). In the above example, it is clear that u_1 and u_2 can be combined in a split range pair, i.e., when u_1 saturates, u_2 is available as a manipulating variable and vice versa. The set of manipulations that need to be combined in a split range can be determined by solving an Integer Linear Program (ILP). However, it must be noted that it may not be possible to use such a split range control structure for all linear systems.

In the succeeding section, we describe a Heat Exchanger Network and the optimal operation policy for the same based on the above ideas.

4. OPTIMAL OPERATION OF HEAT EXCHANGER NETWORKS

The general mathematical model of a Heat Exchanger Network (HEN) yields nonlinear equations and hence, optimization of the same would involve solving a nonlinear programming problem. However, under certain conditions, the problem of optimal operation of a HEN can be reformulated as a Linear Program (Aguilera and Marchetti, 1998; Lersbamrungsuk *et al.*, 2006). The advantages are that the optimization problem can be solved efficiently and the properties of the optimal solution are readily characterized. The following HEN (Fig. 3) is a modified example from (Aguilera and Marchetti, 1998).

The network consists of 3 process-process exchangers, 3 utilities and 4 streams. The outlet temperatures of all streams are to be controlled and maintained at target values. Inlet temperatures of all streams are assumed to be unmeasured disturbances. Nominal inlet and outlet temperatures (targets) are indicated in the figure. The objective function to be minimized is the overall utility consumption. The optimization problem can be solved offline for the expected operating regime (or using the method of parametric programming). Following is a table listing which manipulations are saturated in different regions of the disturbance space, using the same notation as in Table 1.

Table 2. List of saturated manipulations

Set of active constraints	Q_{c1}	Q_{c2}	Q_h	u_{b1}	u_{b2}	u_{b3}
1	A	I	A	I	I	I
2	A	A	I	I	I	I
3	I	A	I	A	I	I
4	I	I	A	A	I	I
5	I	I	A	I	I	A

Since the optimal operation problem is a LP, the optimal solution is at a vertex. Hence, the problem of optimal operation is one of identifying the correct active constraints and implementing them. The split range control structure described earlier is used to switch between constraints. The manipulations that need to be combined in a split range structure are determined by solving an ILP. This results in the following control structure (Table 3 and Fig. 4) that allows a split range implementation and also ensures that the operation is optimal. u_{b2} does not saturate and so does not appear in an split range and is used to control T_{c2out}. Q_{c2} and Q_h are combined in a split range pair to control T_{h2out}, u_{b3} is used to control T_{c1out}, u_{b1} is used to control T_{h1out} and Q_{c1} is used when either u_{b1} or u_{b3} are saturated. In the spirit of the earlier definition of simplicity, it can be seen that the switching can be implemented easily with the split range control architecture. The local controllers can be implemented using the controller of choice, for eg., PI. Such an implementation has low controller complexity with acceptable levels of performance and low switching costs and hence, in the spirit of (4) is an example of a simple strategy.

Table 3. Controller structure for optimal operation

Region	Controller pairing			
	T_{c1out}	T_{c2out}	T_{h1out}	T_{h2out}
1	u_{b3}	u_{b2}	u_{b1}	Q_{c2}
2	u_{b3}	u_{b2}	u_{b1}	Q_h
3	u_{b3}	u_{b2}	Q_{c1}	Q_h
4	u_{b3}	u_{b2}	Q_{c1}	Q_{c2}
5	Q_{c1}	u_{b2}	u_{b1}	Q_{c2}

5. CONCLUSIONS

The need for simple implementation of optimal operation policies was motivated with examples. Simplicity of operation was formally defined for a certain class of systems. The optimal operation of a HEN can be reformulated as a LP. An example of a simple implementation of optimal operation of a HEN was provided using a simple switching strategy and feedback. However, it may be possible that such a simple implementation is not possible for all linear systems. The type of systems for which such a solution is possible needs to be investigated.

Acknowledgment

The authors would like to thank V. Lersbamrungsuk for assistance in generating the results for the HEN.

6. REFERENCES

Aguilera, N. and J.L. Marchetti (1998). Optimizing and controlling the operation of heat-exchanger networks. *AIChE Journal* **44**(5), 1090–1104.

Arkun, Y. and G. Stephanopoulos (1980). Studies in the synthesis of control structures for chemical processes: Part IV. Design of steady-state optimizing control structyres for chemical process units. *AIChE Journal* **26**(6), 975–991.

Aske, E., S. Strand and S. Skogestad (2006). Coordinator mpc with a focus on maximizing throughput. In: *16th European Symposium on Computer Aided Process Engineering and 9th International Symposium on Process Systems Engineering*. pp. 1203–1208.

Bryson, A.E. (1999). *Dynamic optimization*. Addison Wesley.

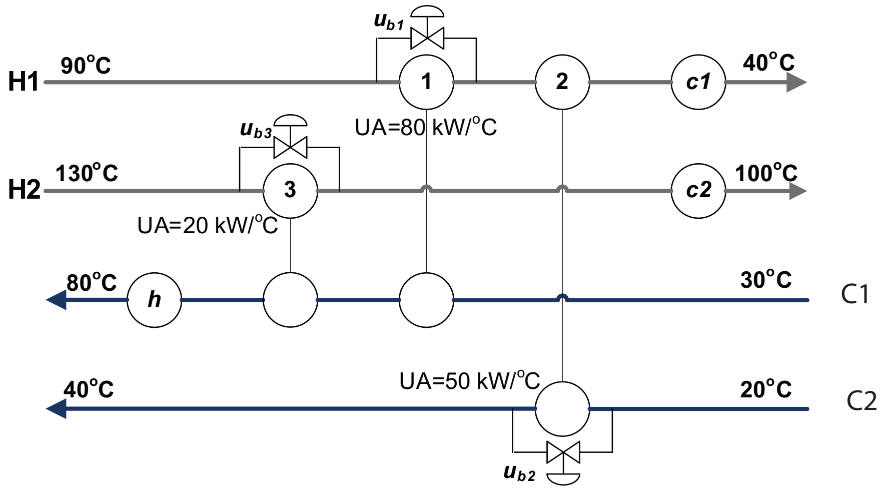

Fig. 3. Heat Exchanger Network

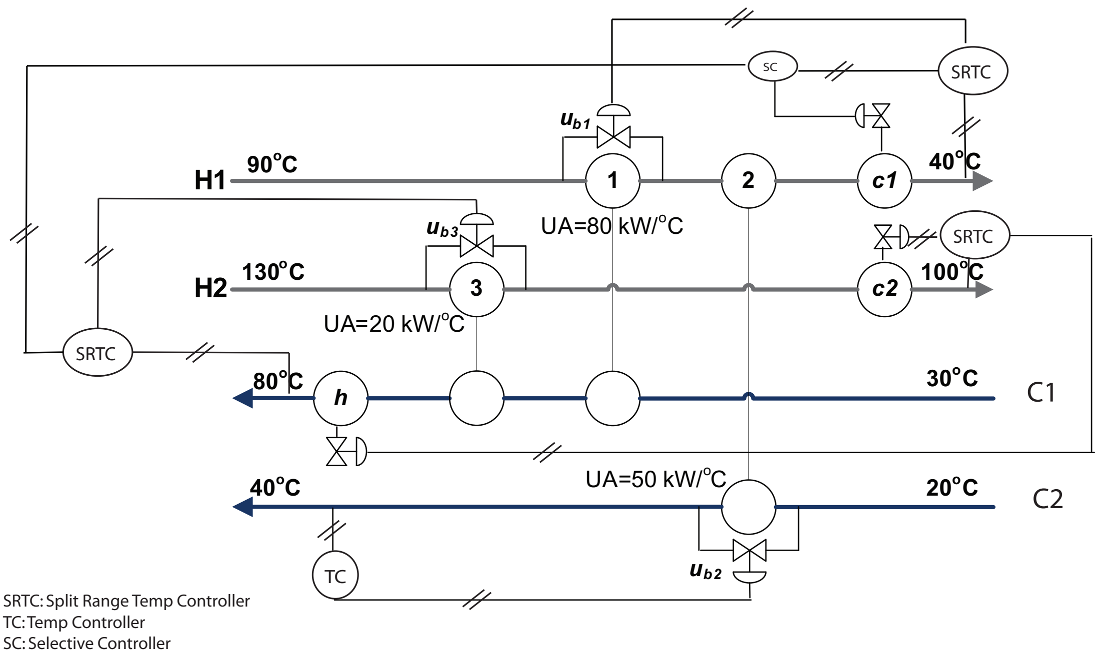

Fig. 4. Control structure for optimal operation of HEN

Findeisen, W., F.N. Bailey, M. Bryds, M. Malinowski, P. Tatjewski and A. Wozniak (1980). *Control and Coordination in Hierarchical Systems*. John Wiley and sons.

Gal, T. (1979). *Postoptimal analyses, parametric programming and related topics*. McGraw-Hill.

Glemmestad, B. (1997). Optimal operation of integrated processes: Studies on Heat Recovery Systems. PhD thesis. Norwegian University of Science and Technology.

Kalman, R.E. (1963). When is a linear control system optimal. In: *Joint Automatic Control Conference, Minneapolis, USA*.

Lersbamrungsuk, V., S. Skogestad and T. Srinophakun (2006). A simple strategy for optimal operation of heat exchanger networks. In: *International Conference on Modeling in Chemical and Biological Engineering Sciences, Bangkok, Thailand*.

Nemhauser, G.L. and L.A. Wolsey (1999). *Integer and combinatorial optimization*. John Wiley and Sons.

Pepyne, D.L. Cassandras, C.G. (1997). Optimal dispatching control for elevator systems during uppeak traffic. *IEEE Transactions on Control Systems Technology* **5**, 629–643.

Pistikopoulos, E.N., D. Dua, N.A. Bozinis, A. Bemporad and M. Morari (2002). On-line optimization via off-line parametric optimization tools. *Computers and Chemical Engineering* **26**, 175–185.

Skogestad, S. (2004*a*). Control structure design for complete chemical plants. *Computers and Chemical Engineering* **28**, 219–234.

Skogestad, S. (2004*b*). Near-optimal operation by self-optimizing control: From process control to marathon running and business systems. *Computers and Chemical Engineering* **29**, 127–137.