Available online at www.sciencedirect.com
SCIENCE DIRECT®
Computers and Chemical Engineering 29 (2004) 127–137
Computers & Chemical Engineering
www.elsevier.com/locate/compchemeng

Near-optimal operation by self-optimizing control: from process control to marathon running and business systems☆

Sigurd Skogestad*

Department of Chemical Engineering, Norwegian University of Science and Technology, N-7491 Trondheim, Norway

Received 7 August 2003; received in revised form 2 June 2004; accepted 20 July 2004
Available online 11 September 2004

Abstract

The topic of this paper is how to implement optimal decisions in an uncertain world. A study of how this is done in real systems—from the nationwide optimization of the economy by the Central Bank to the optimal use of resources in a single cell—shows that a common approach is to use feedback strategies where selected controlled variables are kept at constant values. For example, in order to optimize the wealth of a country (overall objective), the Central Bank may to attempt to keep the inflation constant (selected controlled variable) by adjusting the interest rate (independent input variable). The underlying idea is that the system behavior is indirectly optimized by keeping selected controlled variables at given constant values (setpoints). In the paper this idea of "self-optimizing control" is explained and illustrated on a large number of examples.

Keywords: Optimal operation; Active constraint; Controlled variable; Control structure design; Robustness; Uncertainty

1. Introduction

Consider the national economy, the government, companies and businesses, consumers, chemical process plants, biological systems, and so on. For all these systems we have available degrees of freedom (decisions) u that we want to use in order to optimize the operation (system behavior). We are here *not* concerned with the optimization of these systems (which is certainly very interesting), but rather on how the decisions are *implemented*.

A major problem in making and implementing the right decision is that the world changes and is uncertain. These changes and uncertainties, which we cannot affect, are here denoted *disturbances* d. They include changes in exogenous variables (such as the outdoor temperature), parameter variations in the system (e.g. aging of system components), as well as uncertainty in our model (if any) of the system.

One approach for adapting or correcting for disturbances is to use any new information about the system behavior and the disturbances to reoptimize the decision variables ("on-line optimization"). A simpler strategy is to implement some simple rule such that the system somehow "optimizes itself" without the need for actually performing on-line optimization. We here consider the strategy of "self-optimizing control" (Skogestad, 2000) where the rule is to keep some "magic" controlled variable at a constant value (setpoint). The idea is extremely simple, but it may nevertheless be difficult to grasp. The objective of this paper is two-fold: (1) provide some simple examples that illustrate the concept of self-optimizing control; (2) show that the concept is universal and may be applied for a wide range of systems.

The idea of self-optimizing control is explained in more detail in the next section. In the rest of this section we consider optimal operation and discuss the difficulty in achieving it in practice.

Assume that optimal operation of the system can be quantified in terms of a scalar cost function (performance index)

☆ Extended Version of Paper Presented at the Eighth International Symposium on Process Systems Engineering (PSE'2003), Kunming, China, June 2003/January 2004.

* Tel.: +47 73594154; fax: +47 73594080.
E-mail address: skoge@chemeng.ntnu.no (S. Skogestad).

0098-1354/$ – see front matter © 2004 Elsevier Ltd. All rights reserved.
doi:10.1016/j.compchemeng.2004.07.011

J_0 which is to be minimized with respect to the available degrees of freedom (manipulated variables; inputs) u_0:

$$\min_{u_0} J(x, u_0, d) \tag{1}$$

subject to the constraints

$$g_1(x, u_0, d) = 0, \qquad g_2(x, u_0, d) \leq 0 \tag{2}$$

Here d represents the exogenous disturbances that affect the system, including the effect of changes in the model (typically represented by changes in the function g_1), changes in the specifications (constraints), and changes in the parameters (prices) that enter in the cost function (and possibly in the constraints). x represents the internal states. We have available measurements $y = f_0(x, u_0, d)$ that give information about the actual system behavior during operation. Note that y may include measured values of the disturbances d, as well as known or measured values of the independent variables u_0. For simplicity, we assume pseudo-steady-state behavior and do not in this paper include time as a variable. The equality constraints ($g_1 = 0$) include the model equations, which give the relationship between the independent variables (u_0 and d) and the states (x). The system must generally satisfy several inequality constraints ($g_2 \leq 0$); for example, we usually require that selected variables are positive. The cost function J is in many cases a simple linear function of the independent variables with prices as parameters. In many cases it is more natural to formulate the optimization problem as a maximization of the profit P, which may be formulated as a minimization problem by selecting $J = -P$.

In most cases some subset g_2' of inequality constraints g_2 are active (i.e. $g_2' = 0$ at the optimal solution). Implementation to achieve this is usually simple: we adjust a corresponding number of degrees of freedom u_0 such that these active constraints are satisfied (the possible errors in enforcing the constraints should be included as additional disturbances). For example, consider short-distance (e.g. 100 m) running where the objective is minimize the running time ($J = T$) and the independent variable u_0 is the energy input (or something similar). For a reasonable well-trained runner the optimal solution lies on the constraint of maximum energy. Implementation is then easy; the runner simply runs at maximum speed (applies maximum energy input) throughout the race.

In many cases the active constraints consumes all the available degrees of freedom. For example, if the original problem is linear (linear cost function with linear constraints g_1 and g_2), then it is well known from Linear Programming theory that there will be no remaining unconstrained variables. For nonlinear problems (e.g. the model g_1 is nonlinear), the optimal solution may be unconstrained, and such problems are the focus of this paper. For example, consider long-distance (e.g. marathon) running where the objective is minimize the running time ($J = T$) and the independent variable u_0 is the energy input. In this case the optimal solution does not lie on the constraint of maximum energy, at least not at the beginning of the race, and the runner has to select an appropriate energy input in order to optimize the behavior. How should this be done in practice?

From this example it is clear that the selection of an appropriate operational policy may be a difficult issue for problems with unconstrained degrees of freedom. Such problems are therefore the focus of the rest of this paper. We assume in the following that the active constraints are fulfilled (implemented), and we write, for simplicity, the remaining unconstrained problem in reduced space in the form

$$\min_{u} J(u, d) \tag{3}$$

where u represents the remaining unconstrained degrees of freedom, and where we have also eliminated the states $x = x(u, d)$ by making use of the model equations g_1. For any value of the disturbances d we can then solve the (remaining) unconstrained optimization problem (3) and obtain $u_{\text{opt}}(d)$ for which

$$\min_{u} J(u, d) = J(u_{\text{opt}}(d), d) \stackrel{\text{def}}{=} J_{\text{opt}}(d)$$

The solution of such problems has been studied extensively, and is not the issue of this paper. In this paper the issue is implementation, and how to handle variations (known or unknown) in d in a simple manner.

In the following we let d^* denote the nominal value of the disturbances. Let us first assume that the disturbance variables are constant, i.e. $d = d^*$. In this case implementation is simple: we keep u constant at $u_s = u_{\text{opt}}(d^*)$ (here u_s is the "setpoint" or desired value for u), and we will have optimal operation. However, there are two problems with this policy. First, it is usually not possible in practice to get exactly $u = u_s$ due to an implementation error $n = u - u_s$ (Skogestad, 2000). Second, the disturbance d changes, so $u_{\text{opt}}(d^*)$ is no longer optimal. What value should we select for u_s in this case? Two "obvious" approaches are:

1. If we do not have any information on how the system behaves during actual operation (no measurement y), or if it is not possible to adjust u once it has been selected, then the optimal policy is to find the best "average" value u_s for the expected disturbances, which would involve "backing off" from the nominally optimal setpoints by selecting u_s different from $u_{\text{opt}}(d^*)$. The solution to this problem is quite complex, and depends on the expected disturbance scenario. For example, we may use stochastic optimization (Birge & Louveaux, 1997). In any case, operation may generally be far from optimal for a given disturbance d.
2. In this paper we assume that the unconstrained degrees of freedom u may be adjusted freely and that we have information (measurements $y = f_y(u, d)$) about the actual operation. If we then have a model of the system, we may image using these measurements to estimate the actual disturbance $\tilde{d}$, and based on this perform a reoptimization

to compute a new optimal value $u_{opt}(\tilde{d})$, which is subsequently implemented, $u = u_{opt}(\tilde{d})$.

Both of these "obvious" approaches are complex and require a detailed model of the system, and are not likely to be used in practice, except in special cases. Is there any simpler approach that may work?

2. Implementation of optimal operation: self-optimizing control

Generally, we find that the decision making and implementation in real systems is done in a hierarchical manner (Findeisen et al., 1980) with setpoints for selected controlled variables being send from one layer to the one below. This corresponds to a simple feedback strategy where the degrees of freedom (manipulated variables; inputs) u are adjusted to keep selected controlled variables c at constant values c_s ("setpoints") (see Fig. 1). Here c is a selected subset or combination of the available measurements y. The idea is to get "self-optimizing control" where "near-optimal operation" is indirectly achieved, without the need for continuously solving the above optimization problem.

2.1. Example: Central Bank

Consider the role of the Central Bank in a country, which has available one degree of freedom, namely the interest rate (u). The measurements y may in this case include the inflation rate (y_1), the unemployment rate (y_2), the consumer spending (y_3) and the investment rate (y_4). In addition, we also know the chosen interest rate ($y_5 = u$). The simplest policy would be to do nothing, that is, to keep the interest rate constant (corresponds to the choice $c = y_5 = u$). A more common policy today is for the Central Bank to adjust the interest rate (u) in an attempt to keep the inflation rate constant (corresponds to the choice $c = y_1$). A typical desired value (setpoint) for the inflation rate is $c_s = 2.5\%$.

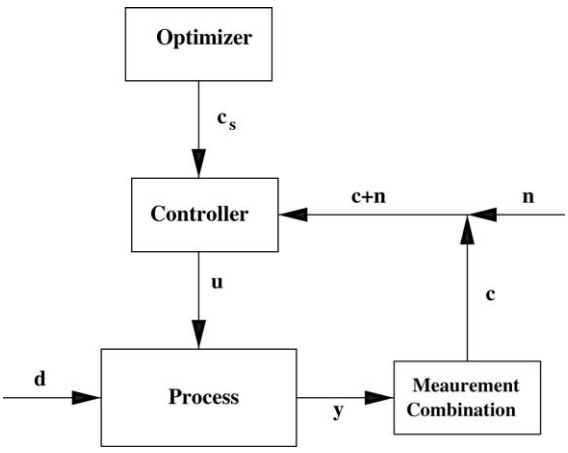

Fig. 1. Implementation of optimal operation with separate layers for optimization and control. Self-optimizing control is when near-optimal operation can be achieved with c_s constant, in spite of disturbances d and implementation error n.

What is the motivation behind attempting to keep c constant at c_s? Obviously, the idea must be that the optimal value of c, denoted $c_{opt}(d)$, depends only weakly on the disturbances d, such that by keeping c at this value, we indirectly obtain optimal, or at least near-optimal, operation (Morari, Stephanopoulos, & Arkun, 1980). More precisely, we may define the loss L as the difference between the actual value of the cost function obtained with a specific control strategy, e.g. adjusting u to keep $c = c_s$, and the truly optimal value of the cost function, i.e.

$$L(u, d) = J(u, d) - J_{opt}(d) \tag{4}$$

Self-optimizing control (Skogestad, 2000) is when we can achieve an acceptable loss with constant setpoint values for the controlled variables (without the need to reoptimize when disturbances occur).

Let us summarize how the optimal operation may be implemented in practice:

1. A subset of the degrees of freedom in u_0 are adjusted in order to satisfy the optimally active constraints.
2. The remaining unconstrained degrees of freedom in u_0 (denoted u) are adjusted in order to keep selected controlled variables c at constant desired values (setpoints) c_s.

Ideally, this results in "self-optimizing control" where no further optimization is required, but in practice some infrequent update of the setpoints c_s may be required. If the set of active constraints changes, then one generally has to change the set of controlled variables c, or at least change their setpoints, since the optimal values generally change in a discontinuous manner when the set of active constraints change.

It is usually straightforward to select variables corresponding to the optimally active constraints. For example, if we want to drive a car from A to B along a straight road in the shortest possible time, subject to the constraint of staying below the speed limit, then speed should be selected as the controlled variable (c) and its setpoint (c_s) should be the given speed limit.

As mentioned, the difficult issue is to select controlled variables for the unconstrained degrees of freedom u. For example, if we want to drive a car from A to B along a very winding road in the shortest possible time, then most likely we must drive slower than the speed limit in order to stay on the road ("remain feasible"). A possible constant setpoint strategy in this case could be, for example, to keep the speed at a constant value $c_s = k\sqrt{s}$ where s is the straight line path ahead and k is a constant. The reason behind this policy could be that the distance for stopping the car increases with the square of the speed.

We next present some simple examples to illustrate the above ideas.

2.2. Example 2: cake baking

Let us consider the final process in cake baking, which is to bake it in an oven. Here there are two independent variables, the heat input ($u_1 = Q$) and the baking time ($u_2 = t$). It is a bit more difficult to define exactly what J is, but it could be quantified as the average rating of a test panel (where 1 is the best and 10 the worst). One disturbance will be the room temperature. A more important disturbance is probably uncertainty with respect to the actual heat input, for example, due to varying gas pressure for a gas stove, or difficulty in maintaining a constant firing rate for a wooden stove. In practice, this seemingly complex optimization problem is solved by using a thermostat that keeps a constant oven temperature (e.g. keep $c_1 = T_{\text{oven}}$ at 200 degC) and keeping the cake in the oven for a given time (e.g. choose $c_2 = u_2 = 20$ min). This feedback strategy, based on measuring the oven temperature c_1, gives a self-optimizing solution where the heat input (u_1) is adjusted to correct for disturbances and uncertainty. The optimal value for the controlled variables (c_1 and c_2) are obtained from a cook book. An improved strategy may be to measure also the temperature inside the cake, and take out the cake when a given temperature is reached (i.e. $u_2 = t$ is adjusted to get a given value of $c_2 = T_{\text{cake}}$).

2.3. Example 3: long distance running

Consider a runner who is participating in a long-distance race, for example a marathon. The cost function to be minimized is the total running time, $J = T$. The independent variable u is the energy input (or something similar). Of course, the runner may perform some "on-line" optimization of his/her body, but this is not easy (especially if the runner is alone), and a constant setpoint policy may probably be more efficient. Does there exist any "magic" self-optimizing variable c that may be kept constant?

A common and simple strategy is to run at the same speed as the other runners (e.g. $c = y_1 =$ distance to best runner, with $c_s = 1$ m). However, this may give infeasibility if one is no longer able to maintain this speed. Also, it does not work if the runner is alone.

Another strategy is to keep constant speed ($c = y_2 =$ speed). However, this policy is not good if the terrain is hilly (d = slope of terrain), where it is clearly optimal to adjust the speed. This policy, as well as the previous one, may also give *infeasibility*, since the runner may not able to maintain the desired speed if the terrain is uphill.

A better self-optimizing strategy for a lone runner may be to keep a constant heart rate ($c = y_3 =$ heart rate) or a constant lactate concentration in the muscles ($c = y_4 =$ lactate level). In these cases, a constant setpoint strategy seems more reasonable, as the speed will be reduced while running uphill.

2.4. Example 4: biology

Biological systems, for example a single cell, contain very complex chemical and biochemical reaction networks, of which significant parts have the function of a feedback control systems (Savageau, 1976; Doyle & Csete, 2002). Indeed, Doyle (lecture, Santa Barbara, February 2002) speculates that many of the supposedly unimportant genes in biological systems are related to control, and compares this with an airplane (or a chemical plant) where the majority of the number of the parts of the system is related to control. Biological systems at the cell level are obviously not capable of performing any "on-line" optimization of its overall behavior. Thus, it seems safe to assume that biological systems by natural selection through millions of years must have developed simple self-optimizing control strategies of the kind discussed in this paper. For biologists it is a challenge is to find out how these complex systems work and what the controlled variables are. Also, if one can identify the controlled variables, then one can imagine performing "reverse engineering" in an attempt to identify the cost function J_0 that nature has been attempting to minimize.

2.5. Example 5: portfolio management

Assume that we want to decide what fraction of our savings should be in stocks and what fraction should be in the bank or in bonds. In this case we want to maximize the future value of the savings, and the manipulated input is the buying (or selling) of stocks. One constant setpoint strategy is to rebalance the portfolio such that we always have a fixed mix (e.g. $c_s = 50\%$) of our capital in stocks and the rest in bonds or bank savings (Fleten, Hyland, & Wallace, 2002; Perold & Sharpe, 1988). This means that we will sell stocks when their value increases and buy stocks when their value goes down.

2.6. Example 6: business systems and KPIs

Business systems are very complex with a large number of degrees of freedom (u), measurements, disturbances and constraints. The overall objective of the system is usually to maximize the profit (or more specifically, the net present value of the future profit, $J = -\text{NPV}$) (although, businesses are often criticized for using other shorter-term objectives, like maximizing the present share value, but we will leave that discussion). In any case, it is clear that few managers base their decisions on performing a careful optimization of their overall operation Instead, managers often make decisions about "company policy", which in many cases involved keeping selected controlled variables (c) at constant values. For example, the recently very popular approach of identifying "value metrics" or *key performance indicators* (KPIs) for the business (e.g. Koppel (2001)), may be viewed as the selection of appropriate controlled variables, that is $c =$ KPI.

Some examples of KPIs may be

- energy consumption per unit produced;
- number of accidents per unit produced;
- number of employees per unit produced;
- research spending per unit produced;
- size of administration relative to production staff;
- time for the business to respond to an order from a customer;
- fraction of manual control loops in the plant.

One may think that the value of the above KPIs should be minimized, but this is not the case since it would imply non-optimal operation with "overspending". The optimal values for the KPIs are typically obtained by comparing oneself with other successful businesses. This is done by "benchmarking" to find the "best business practice". However, it is less obvious what variables to select as KPIs? In theory, the results in the next section may be used to find the optimal set of KPIs. This assumes that we have knowledge about the sensitivity matrix F for how the measurements depend on the disturbances d, and that we disregard the implementation error.

3. Selection of controlled variables

What should we control? As mentioned, we should generally control variables corresponding to the optimally active constraints. We here consider the remaining unconstrained variables (if any) for which the selection of controlled variables is a difficult issue.

To answer this question quantitatively we need to evaluate the loss imposed by keeping the selected controlled variables at constant setpoints. The loss L is defined as $L = J - J_{\text{opt}}(d)$, where J is the cost obtained when the controlled variables c are kept constant and $J_{\text{opt}}(d)$ is the lowest achievable cost with the given disturbance. However, this loss evaluation requires a model of the system, so we will first consider some more qualitative rules for selecting controlled variables.

3.1. Qualitative rules

To approach the problem in a systematic manner, it is useful to consider the reasons why a constant setpoint policy may not be optimal. Generally, there are two reasons, namely the presence of (1) disturbances d and (2) implementation errors n. This is illustrated in Fig. 2 where we see that the actual value $c = c_s - n$ of the controlled variable should ideally correspond to the optimal value $c_{\text{opt}}(d)$ where J has its minimum. This has the following implications for the choice of controlled variables c:

- In order to minimize the effect of disturbances d, we obviously want the *optimal* value of c to remain constant. That is, the sensitivity of $c_{\text{opt}}(d)$ to changes in d should be as *small* as possible.

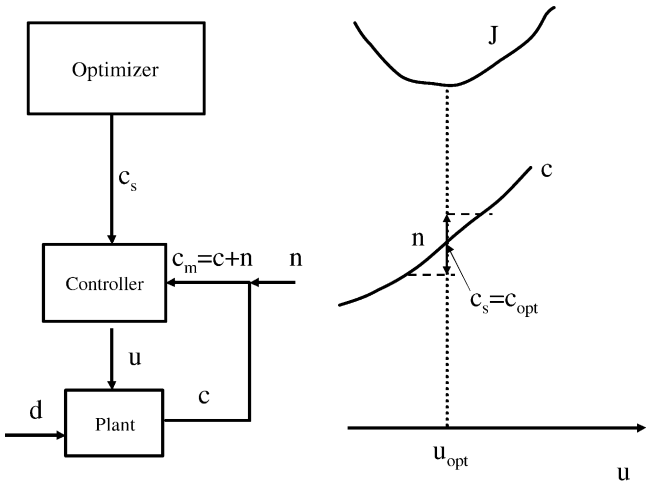

Fig. 2. Feedback implementation of unconstrained optimum.

- In order to minimize the effect of implementation errors n, the sensitivity of c to changes in the independent variable u should be as *large* as possible. This is maybe not so obvious, but it can be quite easily understood from Fig. 2: a large sensitivity corresponds to a steep slope of the c-curve and then a given n will only weakly increase the cost J. (Equivalently, the relationship between c and J should be as "flat" as possible (Skogestad, 2000).)

From this, Skogestad (2000) formulated the following rules for selecting controlled variables c:

1. The optimal value of c should be insensitive to disturbances.
2. c should be easy to measure and control accurately.
3. c should be sensitive to changes in the (steady-state) degrees of freedom. (Equivalently, the cost J as a function of c should be flat.)
4. For cases with more than one unconstrained degrees of freedom, the selected controlled variables should be independent.

The first rule minimizes the effect of disturbances d. The second rule reduces the magnitude of n. The last two rules minimize the effect of the implementation error n. The four rules may be summarized by the following single rule (Skogestad, & Postlethwaite, 1996):

Select controlled variables c for which the controllable range is large compared to the sum of optimal variation and control error.

Here the "controllable range" is the range that c may reach by varying the inputs (degrees of freedom) u, the "optimal variation" is the expected variation in c_{opt} due to disturbance, and the "control error" is the implementation error n.

3.2. Minimum singular value rule

The above rules may be quantified in terms of the minimum singular value rule Skogestad & Postlethwaite, 1996:

Select controlled variables c that maximize the minimum singular value $\underline{\sigma}(G)$ of the appropriately scaled steady-state gain matrix G from inputs u to c.

By inputs u is here meant the unconstrained steady-state degrees of freedom. Note that appropriate scaling is important:

- Scale the candidate variables c such that their expected variation is similar. Specifically, dividing each variable by its optimal variation + implementation error gives each variable an expected variation of magnitude 1.
- Scale the inputs u such that they have similar effect on the cost J (only necessary if we have two or more unconstrained degrees of freedom).

A detailed derivation of the minimum singular value is given in Halvorsen, Skogestad, Morud, & Alstad (2003). It is shown that for small disturbances (local behavior) and with the scaling mentioned above, the expected worst-case loss is bounded by

$$L \leq \frac{\overline{\sigma}(J_{uu})}{2} \frac{1}{\underline{\sigma}(G)^2}$$

From this we see that the expected loss is inversely proportional to the square of $\underline{\sigma}(G)$. The matrix J_{uu} is the second derivative (Hessian) of the cost function with respect to the inputs u, and is thus independent of c. The bound is tight for the scalar case, and usually tight also for cases with more than one input. A more serious limitation is usually that the above bound holds only locally, that is, in the range where J_{uu} is constant. This is often not the case, for example, the optimum may be close to infeasibility (in some other variable or close to "the edge of a cliff").

3.3. Direct evaluation of loss

A direct evaluation of the loss for the expected disturbances and implementation errors avoids the above-mentioned problems with local behavior and infeasibility. The method requires a model of the system and involves the following steps (Skogestad, 2000):

- *Step 1*. Determine the degrees of freedom for optimization.
- *Step 2*. Define optimal operation in terms of a cost function J and operational constraints.
- *Step 3*. Identify the important disturbances.
- *Step 4*. Use the model to find the optimal operation (nominally and with disturbances).
- *Step 5*. Identify active constraints (and control these).
- *Step 6*. For the remaining unconstrained degrees of freedom: evaluate the loss with constant setpoints for alternative controlled variables.
- *Step 7*. Evaluate more carefully the alternatives with a small loss, including controllability analysis.

The procedure has been applied to a large number of process case studies, including the optimal operation of distillation columns (Skogestad, 2000), the Tennesee Eastman challenge process (Larsson, Hestetun, Hovland, & Skogestad, 2001) and the reactor-recycle process (Larsson, Govatsmark, Skogestad, and Yu, 2003).

One disadvantage with this "brute force" method is that it is requires a lot of computations, especially since there is no limit on the possible candidate controlled variables that may be evaluated (in Step 6). It may therefore be important to limit the number of alternatives to evaluate in detail, and methods for this are discussed in detail by Larsson et al. (2001). One effective method is to eliminate choices with a small minimum singular value.

Note that we have assumed that the cost depends only on the steady-state behavior. This is reasonable in most cases. The dynamic (control) behavior comes only into Step 7, and if the control properties are not acceptable, then one needs to go back and evaluate more candidates.

3.4. Optimal measurement combination

Let y denote the available measurements (on-line information about the system behavior), e.g. temperature, inflation rate or energy consumption. We make the simplifying (but very reasonable) assumption that the number of controlled variables (s) is equal to the number of unconstrained degrees of freedom (u). In most cases the controlled variables c are selected simply as a subset of the measurements y, but more generally we may allow for variable combinations and write $c = h(y)$ where the function $h(y)$ is free to choose.

One important disadvantage with the two above methods (singular value rule and direct loss evaluation) is that they can only be used to check given choices for c. If the loss is acceptable then we have "self-optimizing control" and it does not really matter of there exists an even better choice. However, if the loss is not acceptable, then we have no way of knowing if there simply does not exist any self-optimizing scheme or if we have overlooked some "magic" controlled variable $c = h(y)$. Unfortunately, there exists no general method for finding the optimal ("magic") controlled variable, and thus knowing what the best achievable loss is.

The situation is better if we consider the local behavior where it is sufficient to consider linear measurement combinations

$$\Delta c = Hy \tag{5}$$

where the constant matrix H is free to choose. This is discussed next.

3.4.1. Without implementation error

It is possible to find the locally optimal linear measurement combination H as proposed by Halvorsen et al. (2003). The local-behavior assumption makes it possible to use the maximum singular value of a matrix M to effectively evaluate the loss for all expected disturbances and implementations error. However, a numerical search to find the best linear combination is still required. In addition to being computationally demanding the method also provides little insight into how to combine the measurements in order to get a small loss.

3.4.2. With implementation error

The simple method of Alstad and Skogestad (2002) requires much less computations and gives more insight into how to select good controlled variables. The main restriction, besides being a local method, is that the implementation error is not considered.

Interestingly, Alstad and Skogestad (2002) show that with no implementation error it is always possible to find a measurement combination with zero loss, provided we have enough measurements. More precisely, we need as many measurements as there are independent variables (inputs plus disturbances).

The derivation is surprisingly simple; in general, the optimal value of the y values depend on the disturbances d, and we may write this dependency as $y_{\text{opt}}(d)$. For "small" disturbance changes we may linearize this relationship to get

$$\Delta y_{\text{opt}}(d) = F\Delta d \tag{6}$$

where the sensitivity $F = \mathrm{d}y_{\text{opt}}(d)/\mathrm{d}d$ is a constant matrix. We would like to find a variable combination $\Delta c = H\Delta y$ such that $\Delta c_{\text{opt}} = 0$. We get $\Delta c_{\text{opt}} = H\Delta y_{\text{opt}} = HF\Delta d = 0$. This should be satisfied for any value of Δd, so we must require that H is selected such that

$$HF = 0 \tag{7}$$

i.e. H must be in the left null space of F. This is always possible provided we have at least as many (independent) measurements y as we have independent variables (u and d), i.e. number of y = number of u + number of d (Alstad & Skogestad, 2002).

3.4.3. Example 1: Central Bank (continued)

For this problem we have u = interest rate and J = −National Product. A constraint in this problem is $u \geq 0$ (because a negative interest rate will result in an unstable situation), but in most cases this constraint will not be active, so we have an unconstrained optimization problem with one degree of freedom. The measurements y may include the inflation rate (y_1), the unemployment rate (y_2), the consumer spending (y_3) and the investment rate (y_4). There are many disturbances, for example, d_1 ="the mood" of the consumers, d_2 = global politics, including possible wars, d_3 = oil prices, d_4 = weather, d_5 = technology changes, etc. As mentioned earlier, a common policy is to attempt to keep the inflation rate constant, i.e. $c = y_1$. However, recall from the requirement $HF = 0$ for zero loss, that we need an extra measurement for every disturbances, so with the large number of disturbances it is unlikely that this choice, based on a single measurement, is always self-optimizing. Even if we assume that there is only a single major disturbance (e.g. d_1 =consumer mood), then from the results presented above we need to combine at least two measurements (number of y = number of u + number of $d = 1 + 1 = 2$). This could, for example, be a corrected inflation goal based on using the interest rate, $c = h_1 y_1 + h_2 u$, but more generally we could use additional measurements, $c = h_1 y_1 + h_2 y_2 + h_3 y_3 + h_4 y_4 + h_5 u$. The parameters for such a corrected inflation goal could be obtained by reoptimizing the model for the national economy with alternatives disturbances, using the approach just outlined.

3.5. Summary: selection of controlled variables

There are broadly three classes of systems when it comes to the use of self-optimizing control.

- *Class A.* Systems for which we have no model, and where on-line optimization is therefore not possible. This class of problems is ideally suited for self-optimizing control, but since there is no model we cannot obtain the controlled variables in a systematic manner. An example is long-distance running. In such cases the self-optimizing controlled variables, if they exist, must be obtained by trying alternative choices on the real system, for example, through evolution.
- *Class B.* Systems that can be modelled, but where on-line optimization is impractical or costly. Self-optimizing control is also well suited for such problems, and the controlled variables may be obtained using the methods mentioned above. A generic model usually suffices because the selection of controlled variables is a *structural* issue which usually does not depend strongly on the specific parameter values.
- *Class C.* Simple systems that are easy to model and optimize on-line (like the blending case considered in the next section). In this case it may be better to use on-line optimization, rather than using a large effort to find self-optimizing controlled variables, if they exist. In particular, this is the case for problems where the active constraints change with time.

Finally, it is stressed again that we in this section have assumed that the set of active constraints remains constant. If the active constraints change, then generally the best set of "unconstrained" controlled variable (and setpoints) will also change.

4. Example 7: optimal blending of gasoline

The following example is included for illustrative purposes, as an on-line model-based approach is simple and

would probably be preferred for this process. Nevertheless, we here assume that we want to use a constant setpoint strategy. The example then illustrates clearly the importance of selecting the right controlled variables, and illustrates nicely of the nullspace method of Alstad and Skogestad (2002) for selecting optimal measurement combinations.

Problem statement. We want to make 1 kg/s of gasoline with at least 98 octane and not more than 1 wt.% benzene, by mixing the following four streams:

- Stream 1: 99 octane, 0% benzene, price $p_1 = (0.1 + m_1)$ \$/kg.
- Stream 2: 105 octane, 0% benzene, price $p_2 = 0.200$ \$/kg.
- Stream 3: 95 octane, 0% benzene, price $p_3 = 0.12$ \$/kg.
- Stream 4: 99 octane, 2% benzene, price $p_4 = 0.185$ \$/kg.

The maximum amount of stream 1 is 0.4 kg/s. The disturbance (d) is the octane contents in stream 3 which may vary from 95 (its nominal value) and up to 97. We want to obtain a self-optimizing strategy that "automatically" corrects for this disturbance.

Solution. The degrees of freedom for this problem are

$$u_0 = [\, m_1 \; m_2 \; m_3 \; m_4 \,]^{\mathrm{T}}$$

where m_i (kg/s) represents the mass flows of the individual streams. The optimization problem is to minimize the cost of the raw material:

$$J(u_0) = \sum_i p_i m_i = (0.1 + m_1)m_1 + 0.2m_2 + 0.12m_3 + 0.185m_4$$

subject to 1 equality constraint (given product rate) and 7 inequality constraints:

$$m_1 + m_2 + m_3 + m_4 = 1$$

$$m_1 \geq 0, \qquad m_2 \geq 0, \qquad m_3 \geq 0, \qquad m_4 \geq 0$$

$$m_1 \leq 0.4$$

$$99m_1 + 105m_2 + dm_3 + 99m_4 \geq 98$$

$$2m_4 \leq 1$$

At the nominal operating point (where the octane number in stream 3 is $d^* = 95$) the optimal solution is to have

$$u_{0,\mathrm{opt}}(d^* = 95) = [\, 0.26 \; 0.196 \; 0.544 \; 0 \,]^{\mathrm{T}}$$

which gives $J_{\mathrm{opt}}(d^*) = 0.13724$ \$. We find that three constraints are active (the product rate equality constraint, the non-negative flowrate for m_4 and the octane constraint). The same three constraints remain active when we change the octane number in stream 3 from 95 to 97, where the optimal solution is to have

$$u_{0,\mathrm{opt}}(d = 97) = [\, 0.20 \; 0.075 \; 0.725 \; 0 \,]^{\mathrm{T}}$$

which corresponds to $J_{\mathrm{opt}}(d = 97) = 0.126$ \$.

The proposed control strategy is then to use three of the degrees of freedom in u_0 to control the following variables (active constraint control):

1. keep the product rate at 1 kg/s;
2. keep the octane number at 98;
3. keep $m_4 = 0$.

This leaves one unconstrained degree of freedom. We would like to achieve self-optimizing control. To this effect, we now evaluate the loss imposed by keeping alternative single controlled variables c constant at their nominal optimal values, $c_{\mathrm{s}} = c_{\mathrm{opt}}(d^*)$. The measurements available are in this case a subset of u_0, namely

$$y = [\, m_1 \; m_2 \; m_3 \,]^{\mathrm{T}}$$

Here we have excluded m_4 since it is kept constant at 0, and thus is independent of d and u. Let us first consider keeping individual flows constant (and the three others adjusted to satisfy the active constraints on product rate, octane number and zero flow for m_4). We find when d is changed from 95 to 97:

- $c = m_1$ constant at 0.26: $J = 0.12636$ corresponding to loss $L = 0.12636 - 0.126 = 0.00036$;
- $c = m_2$ constant at 0.196: infeasible (requires a negative m_3 to satisfy the octane constraint);
- $c = m_3$ constant at 0.544: $J = 0.13182$ corresponding to loss $L = 0.13182 - 0.126 = 0.00582$.

Let us now obtain the optimal variable combination that gives zero loss. We consider a linear variable combination of the measurements y:

$$\Delta c = H\Delta y = h_1\Delta m_1 + h_2\Delta m_2 + h_3\Delta m_3$$

The relationship between the optimal value of y and the disturbance is exactly linear in this case and we have

$$\Delta y_{\mathrm{opt}} = F\Delta d = \begin{pmatrix} 0.20 - 0.26 \\ 0.075 - 0.196 \\ 0.725 - 0.544 \end{pmatrix} \frac{\Delta d}{2} = \underbrace{\begin{pmatrix} -0.0300 \\ -0.0605 \\ 0.0905 \end{pmatrix}}_{F} \Delta d$$

(we divide by 2 because the disturbance is 2 octane number units). To get a variable combination with zero loss we must

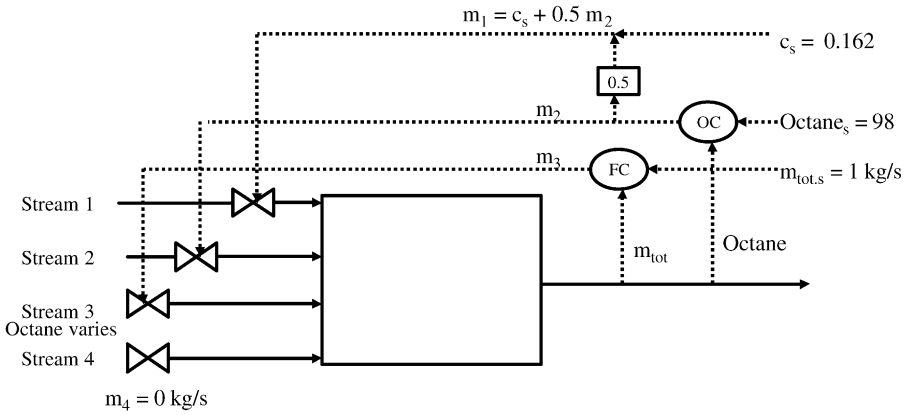

Fig. 3. Possible implementation of optimal blending with $c = m_1 - 0.496m_2$ as the "self-optimizing" controlled variable.

have $HF = 0$ or

$$-0.0305h_1 - 0.0605h_2 + 0.0905h_3 = 0$$

In this case we have 1 unconstrained degree of freedom (u) and 1 disturbance (d), so we need to combine at least 2 measurements to get a variable combination with zero loss. This is confirmed by the above equation which may always be satisfied by selecting one element i, H equal to zero. We therefore have an infinite number of possible combinations of variables with zero loss. For example, the following three combinations of two variables give zero loss:

1. $c = m_1 - 0.496m_2$ constant at 0.162: zero loss (derived by setting $h_3 = 0$ and choosing $h_1 = 1$);
2. $c = 3.02m_1 + m_3$ constant at 1.328: zero loss (derived by setting $h_2 = 0$ and choosing $h_3 = 1$);
3. $c = 1.496m_2 + m_3$ constant at 0.837: zero loss (derived by setting $h_1 = 0$ and choosing $h_3 = 1$).

A possible implementation of optimal blending with single-loop controllers and $c = m_1 - 0.496m_2$ as the "self-optimizing" controlled variable is shown in Fig. 3. Pairings of inputs and outputs, for example the largest flow is used to control the total flowrate, have been included for illustration. However, note that the choice of pairings does not influence the steady-state costs as long as the controllers have integral action. This control structure is steady-state optimal as long as the active constraints do not change. Disturbances in the feed octane in stream 3 are detected by the octane controller (OC) which adjusts m_2, and m_1 is then adjusted to keep c at its setpoint $c_s = 0.162$. Price changes may be included as a correction on the setpoint c_s (see Section 5).

As mentioned, there are an infinite number of variable combinations c of the three measurements (m_1, m_2, m_3) with zero disturbance loss. However, if we also were to include the implementation error, then there would be a single optimal combination of the three measurements in terms of the overall expected loss (Halvorsen et al., 2003) (see Section 5).

5. Discussion

5.1. Change in prices

By "prices" we mean the parameters or weights that enter in the cost function.

In the above example, the prices were assumed constant, but from simple considerations it is clear that, unless the active constraints change, price changes will not affect the selection of controlled variables. The reason is that prices appear only in the optimization part of the block diagram in Fig. 1 and do not effect the measurements y and also not the controlled variable $c = h(y)$. In other words, no matter which variables c we choose to control, there will be no "self-correction" to price changes.

However, prices changes will of course influence the optimal value of the variables, and since prices are generally known, we would like to include some "price correction". This may be done in two different ways:

1. Make the setpoint c_s a function of the prices p (this is probably the simplest and most obvious approach). Specifically, for a price change $\Delta p = p - p^*$, the corrected setpoint is

   $$c_s = c_s(p^*) + HF_p\Delta p$$

   where the matrix $F_p = \mathrm{d}y_{opt}/\mathrm{d}p$ is the optimal sensitivity of the measurements to prices (see below).
2. Keep constant setpoints, and instead include the (known) prices as extra "measured disturbances" d (Skogestad, 2004).

5.1.1. Example 7 (continued)

Let us return to the blending example, and consider the case where the price of stream 2 may vary. Specifically, changing the price p_2 from 0.2 to 0.21 gives the same set of active constraints and the following new optimal flows

$$u_{0,opt}(p_2 = 0.21, d = 95) = [\,0.28\ 0.188\ 0.532\ 0\,]^T$$

The sensitivity to price changes is then

$$\Delta y_{\text{opt}} = [\,2.0\ -0.8\ -1.2\,]^{\text{T}}\Delta p_2$$

If we, for example, select $c = m_1 - 0.496m_3 = y_1 - 0.496y_3$ as the controlled variable, then the price-corrected setpoint is $c_s = 0.162 + (2 - 0.496(-0.8))\Delta p_2 = 0.162 + 2.40\Delta p_2$.

Note that the above discussion assumes that the price changes are sufficiently small such that the active constraints do not change. Larger price changes are likely to change the set of active constraints, and through this strongly affect the choice of controlled variables. In other words, there is usually not a single set of controlled variables that will be the best for all prices.

5.2. Implementation error

One issue which we have not discussed so far is the implementation error n, which is the difference between the actual controlled variable c and its desired value ($n = c_s - c$). In some cases there may be no implementation error, but this is relatively rare.

5.2.1. Example 1 (continued)

Let us again consider the Central Bank. A simple policy would be to do nothing, that is keep the interest rate constant (i.e. select $c = u$). In this case there would be no implementation error. However, a more common policy is to attempt to keep the inflation rate constant ($c = y_1$), and in this case there will generally be a difference n between the actual inflation rate (c) and its desired value (c_s), because of (i) poor dynamic control and (ii) an incorrect measurement of the inflation rate.

5.2.2. Remark

In this special case (Section 2.1) there is no implementation error when using the "no-control" (open-loop) policy with $c = u$, but this is not at all a general rule. For example, in a wood-fired pizza oven (Section 2.2) our inability to keep the heat input (u_1) at a constant desired value, may be a key reason for avoiding the open-loop policy ($c_1 = u_1$).

In any case, the implementation error n generally needs to be taken into account, and it will affect the optimal choice for the controlled variables. Specifically, with implementation error it is no longer possible to find a set of controlled variables that give zero loss. One way of seeing this is to consider the implementation error n as a special case of a disturbance d. Recall that to achieve zero loss, we need to add one extra measurement y for each disturbance. However, no measurement is perfect, so this extra measurement will also have an associated error ("noise"), which may again be considered as an additional disturbance, and so on.

Unfortunately, the implementation error makes it much more difficult to find the optimal measurement combination, $c = h(y)$, to use as controlled variables. As mentioned earlier, numerical approaches may be used, at least locally (Halvorsen et al., 2003), but these are quite complicated.

Finally, it should be noted that Fig. 1 is a bit misleading as it (i) only includes the contribution to n from the measurement error, and (ii) gives the impression that we directly measure c, whereas we in reality measure y, i.e. n in Fig. 1 represents the combined effect on c of the measurement errors for y.

5.3. Model uncertainty

Model uncertainty, the differences between the actual system and its model, is usually not very important when implementing a "self-optimizing" constant setpoint policy. This follows since the model is not explicitly used in a constant setpoint policy, but rather we are using a feedback implementation based on measurements from the actual plant. It may be desirable to use the model to obtain the optimal setpoints c_s, but alternatively we may attempt to obtain c_s by observing the actual behavior. A model *is* needed when using the above rocedure to select the best controlled variable (with minimum loss), but since we are using this model to make structural rather than parametric decisions, it is obviously not critical if there is some mismatch between the system and the model, as long as its structural properties are correct.

6. Conclusion

Most real systems are operated by keeping selected "controlled variables" at given values ("setpoints"). The goal is to have "self-optimizing control" which is when near-to-optimal operation is achieved with constant setpoints (or infrequent updates). Many real examples have been presented in this paper.

There are broadly three classes of problems when it comes to the use of self-optimizing control. (A) A self-optimizing constant setpoint strategy is obviously ideally suited for systems that are difficult to model, and thus cannot be optimized on-line. An example is long-distance running. However, in such cases the self-optimizing controlled variables, if they exist, must be obtained by trial and error. (B) Self-optimizing control is also well suited for systems that can be modelled, but where on-line optimization is impractical or costly. In such cases, controlled variables may be obtained using the methods mentioned in this paper based on a "generic" model of the system. (C) For simple systems that are easy to model and optimize, like the blending case, it is probably better to use on-line optimization.

For class B, where a model is available, the controlled variables may be obtained in a systematic manner. The first step is to quantify the operational objectives through a scalar cost function J to be minimized. Next, the system is optimized with respect to its degrees of freedom u_0. From this we identify the "active constraints" which are implemented as such. Finally, if there remains unconstrained degrees of freedom u, we must identify appropriate controlled variables c to keep at

constant setpoints. Several methods exist for this, including the minimum singular value rule, "brute force" loss evaluation and optimal measurement combination. These methods have not been discussed in detail, as the main objective of this paper has been to give the reader an understanding of the idea of self-optimizing control.

References

Perold, A. F., & Sharpe, W. F. (1988). Dynamic strategies for asset allocation. *Financial Analysts Journal*, 44(1), 16–27.

Alstad, V., & Skogestad, S. (2002). Robust operation by controlling the right variable combination. *AIChE Annual Meeting, Indianapolis, USA*, 2002 (Available from the home page of S. Skogestad).

Birge, J. R., & Louveaux, F. (1997). *Introduction to stochastic programming*. Springer.

Doyle, J. C., & Csete, M. E. (2002). Reverse engineering of biological complexity. *Science*, *295*, 1664–1669.

Findeisen, W., Bailey, F. N., Brdys, M., Malinowski, K., Tatjewski, P., & Wozniak, A. (1980). *Control and coordination in hierarchical systems*. John Wiley & Sons.

Fleten, S. -E., Hyland, K., & Wallace, S. W. (2002). The performance of stochastic dynamic and fixed mix portfolio models. *European Journal of Operational Research*, 37–49.

Halvorsen, I. J., Skogestad, S., Morud, J. C., & Alstad, V. (2003). Optimal selection of controlled variables. *Industrial Engineering and Chemical Research*, *42* (14), 3273–3284.

Koppel, L. B. (2001). Business process control: the outer loop. *Proceedings of the Symposium on Chemical Process Control (CPC'6), Tuscon, Arizona, January 2001*.

Larsson, L., Hestetun, K., Hovland, E., & Skogestad, S. (2001). Self-optimizing control of a large-scale plant: the Tennessee Eastman process. *Industrial Engineering and Chemical Research*, *40*, 4889–4901.

Larsson, T., Govatsmark, M. S., Skogestad, S., & Yu, C. C. (2003). Control structure selection for reactor, separator and recycle processes. *Industrial Engineering and Chemical Research*, *42*, 1225–1234.

Morari, M., Stephanopoulos, G., & Arkun, Y. (1980). Studies in the synthesis of control structures for chemical processes: Part I. *AIChE Journal*, *26* (2), 220–232.

Savageau, M. A. (1976). *Biochemical systems analysis*. Addison-Wesley.

Skogestad, S. (2000). Plantwide control: the search for the self-optimizing control structure. *Journal of Process Control*, *10*, 487–507.

Skogestad, S. (2004) Integration of optimal operation and control. In: P. Seferlis & M.C. Georgiadis (Eds.), *The integration of process design and control*. Computer-Aided Chemical Engineering Series No.17. Elsevier, pp. 485–500.

Skogestad, S., & Postlethwaite, I. (1996). *Multivariable feedback control*. John Wiley & Sons.