Chapter 2
Optimal Use of Measurements for Control, Optimization and Estimation using the Loss Method: Summary of Existing Results and Some New

Sigurd Skogestad and Ramprasad Yelchuru and Johannes Jäschke

Abstract The objective of this chapter is to study the optimal use of measurements and measurements combinations, $\boldsymbol{c} = \boldsymbol{H}\boldsymbol{y}$ in optimization and estimation based on the loss method.

2.1 Introduction

In this paper we consider a (steady-state) unconstrained quadratic optimization problem with linear measurement relationships. The main objective is to find a linear measurement combination, $\boldsymbol{c} = \boldsymbol{H}\boldsymbol{y}$, such that control of these indirectly leads to close-to-optimal operation with a small loss L, in spite of unknown disturbances, $\boldsymbol{d}$, and measurement noise (error), $\boldsymbol{n}^y$. If the original optimization problem is constrained, then we assume that any active constraints are kept constant (controlled) and we consider the lower-dimensional unconstrained subspace. Depending on the disturbance range considered, there may be several constrained regions, and the procedure of finding $\boldsymbol{H}$ needs to be repeated in each constrained region. Switching between the regions will then be needed, and we will show that monitoring the controlled variables $\boldsymbol{c} = \boldsymbol{H}\boldsymbol{y}$ in neighboring regions can be used for switching.

Sigurd Skogestad
Department of Chemical Engineering, Norwegian University of Science and Technology in Trondheim, Norway, e-mail: skoge@chemeng.ntnu.no

Ramprasad Yelchuru
Department of Chemical Engineering, Norwegian University of Science and Technology in Trondheim, Norway, e-mail: ramprasad.yelchuru@chemeng.ntnu.no

Johannes Jäschke
Department of Chemical Engineering, Norwegian University of Science and Technology in Trondheim, Norway, e-mail: jaschke@chemeng.ntnu.no

What we here call the “loss method” is the same as what is called the “exact local method” in these papers (Halvorsen et al (2003); Alstad et al (2009)).

The new material in this summary paper is mainly related to using data $(\boldsymbol{Y}, \boldsymbol{X})$ as the basis, and for example, to use the “loss” method for regression, see section 5, data approach 1 and 2.

2.2 Problem Formulation

2.2.1 Classification of Variables

- $\boldsymbol{u}$ - inputs (degrees of freedom) for optimization and control (it does not actually matter what they are as long as they form an independent set)
- $\boldsymbol{d}$ - disturbances, including parameter changes.
- $\boldsymbol{y}$ - all available measurements (will later call a subset of these for $\boldsymbol{x}$ in accordance with statistics notation). The manipulated variables (MVs, often the same as the inputs $\boldsymbol{u}$) are generally included in the measurement set $\boldsymbol{y}$. This will allow, for example, for simple control policies where the inputs are kept constant. Of course, the set $\boldsymbol{y}$ also includes measured disturbances ($\boldsymbol{d}_m$, a subset of $\boldsymbol{d}$).
- $\boldsymbol{n}^y$ - measurement noise (error) for $\boldsymbol{y}$, $\boldsymbol{y_m} = \boldsymbol{y} + \boldsymbol{n}^y$.
- $\boldsymbol{p}$ - prices = weights that enter into cost function (do not affect $\boldsymbol{y}$)

2.2.2 Cost Function

The objective is to choose the input $\boldsymbol{u}$ to minimize the quadratic cost function

$$\boldsymbol{J}(\boldsymbol{u}, \boldsymbol{d}) = \boldsymbol{u}^T \boldsymbol{Q}_1 \boldsymbol{u} + \boldsymbol{d}^T \boldsymbol{Q}_2 \boldsymbol{d} + \boldsymbol{u}^T \boldsymbol{Q}_3 \boldsymbol{d} \tag{2.1}$$

Note that for simplicity, we have not included linear terms in the cost function. Any linear term in $\boldsymbol{u}$ can be removed by shifting the zero point for $\boldsymbol{u}$ to be at the optimal point. On the other hand, a linear term in $\boldsymbol{d}$ cannot be counteracted by choosing the input $\boldsymbol{u}$, so excluding it does not change the solution. The same applies to any constant term in the cost.

If we compare (2.1), with a second-order Taylor series expansion of the cost around a nominal point $(\boldsymbol{u}^*, \boldsymbol{d}^*)$, then we have that

$$\boldsymbol{Q}_1 = \frac{1}{2}\boldsymbol{J}^*_{uu}, \boldsymbol{Q}_2 = \boldsymbol{J}^*_{ud}, \boldsymbol{Q}_3 = \frac{1}{2}\boldsymbol{J}^*_{dd}$$

and $\boldsymbol{u}$ represents deviation from the optimal point $(\boldsymbol{u}^*, \boldsymbol{d}^*) = (0, 0)$ at which $\boldsymbol{J}_u^* = 0$.

2.2.3 Measurement Model

A linear model is assumed for the effect on $\boldsymbol{u}$ and $\boldsymbol{d}$ on measurements $\boldsymbol{y}$ (deviation variables)

$$\boldsymbol{y} = \boldsymbol{G}^y \boldsymbol{u} + \boldsymbol{G}_d^y \boldsymbol{d} = \tilde{\boldsymbol{G}} \begin{bmatrix} \boldsymbol{u} \\ \boldsymbol{d} \end{bmatrix} \tag{2.2}$$

2.2.4 Assumptions

- No constraints ($\boldsymbol{u}$ spans unconstrained subspace)
- We want to find as many controlled variables as there are degrees of freedom, $n_c = \dim(\boldsymbol{c}) = \dim\,(\boldsymbol{u}) = n_u$. Then $\boldsymbol{H}\boldsymbol{G}^y$ is a square $n_u \times n_u$ matrix
- We use at least as many measurements as there are degrees of freedom, $n_y \geq n_u = n_c$.

2.2.5 Expected Set of Disturbances and Noise

We write $\boldsymbol{d} = \boldsymbol{W}_d\ \boldsymbol{d}'$ where $\boldsymbol{W}_d$ is a diagonal matrix giving the expected magnitude of each disturbance and $\boldsymbol{d}'$ is a normalization vector of unit magnitude.

Similarly, $\boldsymbol{n}^y = \boldsymbol{W}\boldsymbol{n}^y\ \boldsymbol{n}^{y'}$ where $\boldsymbol{W}_{\boldsymbol{n}^y}$ is a diagonal matrix with the magnitude of the noise for each measurement, and the vector $\boldsymbol{n}^{y'}$ is of unit magnitude.

More precisely, the combined normalization vectors for disturbances and measurement noise are assumed to have 2-norm less than 1,

$$\left\| \begin{matrix} \boldsymbol{d}' \\ \boldsymbol{n}^{y'} \end{matrix} \right\|_2 \leq 1 \tag{2.3}$$

The choice of the 2-norm (rather than, for example, the vector infinity-norm) is discussed in the Appendix of Halvorsen et al (2003).

2.2.6 Problem

Given that

$$\boldsymbol{H}\underbrace{(\boldsymbol{y}+\boldsymbol{n}^y)}_{\boldsymbol{y}_m} = c_s \quad \text{(constant = 0 nominally)} \tag{2.4}$$

find the optimal $\boldsymbol{H}$ such that "magnitude" of the loss

$$L = \boldsymbol{J}(\boldsymbol{u},\boldsymbol{d}) - \boldsymbol{J}_{opt}(\boldsymbol{d}) \tag{2.5}$$

is minimized for the "expected" $\boldsymbol{d}$ and $\boldsymbol{n}^y$.

The "expected" set of the disturbances and noise is defined above.

The "magnitude" of the loss still needs to be defined. Two possibilities are considered.

- Worst-case loss, L_{wc}.
- Average loss, L_{avg}.

2.2.7 Examples of this Problem

1. Identify controlled variables, $\boldsymbol{c} = \boldsymbol{H}\boldsymbol{y}$ ("squaring down"). Then use feedback control to adjust $\boldsymbol{u}$ such that $\boldsymbol{c}_m = \boldsymbol{H}\boldsymbol{y}_{\boldsymbol{m}} = \boldsymbol{c}_s$.
2. Find invariants for quadratic optimization problems.
3. Obtain estimate of primary variables, $\boldsymbol{c} = \hat{\boldsymbol{y}}_1 = \boldsymbol{H}\boldsymbol{y}$.
   Problem: Given that $\hat{\boldsymbol{y}}_1 = \boldsymbol{H}\boldsymbol{y}$ find optimal $\boldsymbol{H}$ such that magnitude of $||\boldsymbol{y}_1 - \hat{\boldsymbol{y}}_1||$ is minimized for the expected $\boldsymbol{d}$'s and $\boldsymbol{n}^y$'s.

2.2.8 Comments on the Problem

1. The controlled variables are $\boldsymbol{c} = \boldsymbol{H}\boldsymbol{y}$ and the objective is to find the non-square $n_c \times n_y$ matrix $\boldsymbol{H}$ (note that $n_c = n_u$). In general, $\boldsymbol{H}$ is a "full" combination matrix. However, it may also be interesting to consider control of individual measurements, in which case $\boldsymbol{H}$ is a "selection" matrix with n_u number of columns with single 1 and the rest of columns are zero (mathematically $\boldsymbol{H}\boldsymbol{H}^T = I$).
2. Minimizing (the magnitude of) the loss L is close to but not quite the same as minimizing the cost J. In some cases they give identical results in terms of the optimal $\boldsymbol{H}$, for example, if we consider the average loss or cost for given disturbances (because then the same cost function is subtracted). So it seems it is the same for the 2-norm (Frobenius) of $\boldsymbol{M}$ (see below). However, there will be some difference if we consider the worst-case loss or cost.

2.3 Solution to Problem: Preliminaries

The objective is to derive the solution to the above problem. It has been previously been known as the "exact local method", but we will here call it the loss method. However, first we need some preliminaries

2.3.1 Expression for $u_{opt}(d)$

We want to find the optimal input $\boldsymbol{u}$ for a given disturbance $\boldsymbol{d}$. Expanding the gradient $\boldsymbol{J}_u$ around the nominal point $(\boldsymbol{u}^*, \boldsymbol{d}^*) = (0, 0)$ gives

$$\boldsymbol{J}_u = \boldsymbol{J}_u^* + \boldsymbol{J}_{uu}^* \boldsymbol{u} + \boldsymbol{J}_{ud}^* \boldsymbol{d} = \underbrace{\boldsymbol{J}_u^*}_{=0} + [\boldsymbol{J}_{uu}^* \quad \boldsymbol{J}_{ud}^*] \begin{bmatrix} \boldsymbol{u} \\ \boldsymbol{d} \end{bmatrix}$$

where $\boldsymbol{J}_u^* = \boldsymbol{J}_u(\boldsymbol{u}^*, \boldsymbol{d}^*) = 0$ because the nominal point is assumed to be optimal. To remain optimal, $\boldsymbol{u} = \boldsymbol{u}_{opt}(\boldsymbol{d})$, we must have $\boldsymbol{J}_u = 0$ and we derive

$$\boldsymbol{u}_{opt} = -\boldsymbol{J}_{uu}^{-1} \boldsymbol{J}_{ud} \boldsymbol{d} \tag{2.6}$$

where we have dropped the superscript * (either because we consider small deviations or because we assume that the problem is truly quadratic).

2.3.2 Expression for J around $u_{opt}(d)$

Consider a given disturbance $\boldsymbol{d}$. Then expanding the cost J around a "moving" $\boldsymbol{u}_{opt}(\boldsymbol{d})$ gives

$$J(\boldsymbol{u}, \boldsymbol{d}) = \underbrace{J(\boldsymbol{u}_{opt}(\boldsymbol{d}), \boldsymbol{d})}_{\boldsymbol{J}_{\text{opt}}(d)} + \underbrace{\boldsymbol{J}_u}_{=0}(\boldsymbol{u} - \boldsymbol{u}_{opt}) + \frac{1}{2}(\boldsymbol{u} - \boldsymbol{u}_{opt})^T \boldsymbol{J}_{uu}(\boldsymbol{u} - \boldsymbol{u}_{opt}) \tag{2.7}$$

Here $\boldsymbol{J}_u = 0$ (since we are expanding around an optimal point), so we get the following expression for the loss

$$L(u, d) = J(u, d) - J_{opt}(d) = \frac{1}{2}\boldsymbol{z}^T \boldsymbol{z} = \frac{1}{2}\,||\boldsymbol{z}||_2^2 \tag{2.8}$$

where

$$\boldsymbol{z} = \boldsymbol{J}_{uu}^{1/2}(\boldsymbol{u} - \boldsymbol{u}_{opt}(d)) \tag{2.9}$$

2.3.3 Expression for J_u around Moving $u_{opt}(d)$

A similar expansion, but now of the gradient gives

$$\boldsymbol{J}_u = \underbrace{\boldsymbol{J}_u(\boldsymbol{u}_{opt})}_{=0} + \boldsymbol{J}_{uu}(\boldsymbol{u} - \boldsymbol{u}_{opt}) = \boldsymbol{J}_{uu}(\boldsymbol{u} - \boldsymbol{u}_{opt}) \tag{2.10}$$

Combining this with (2.9) gives $\boldsymbol{z} = \boldsymbol{J}_{uu}^{-1/2}\boldsymbol{J}_u$ and we have the "Johannes expression" for the loss

$$L(u,d) = J(u,d) - J_{opt}(d) = \frac{1}{2}\left|\left|\boldsymbol{J}_{uu}^{-1/2}\boldsymbol{J}_u\right|\right|_2^2 \tag{2.11}$$

From these expressions we see that minimizing the loss L is equivalent to minimizing $||\boldsymbol{z}||_2$, which is equivalent to minimizing $\left|\left|\boldsymbol{J}_{uu}^{1/2}(\boldsymbol{u} - \boldsymbol{u}_{opt})\right|\right|_2$ or $\left|\left|\boldsymbol{J}_{uu}^{-\frac{1}{2}}\boldsymbol{J}_u\right|\right|_2$.

Thus, we have the important conclusion that minimizing the loss is equivalent to minimizing the weighted 2-norm of the gradient $\boldsymbol{J}_u$, with the weight being given by the matrix $\boldsymbol{J}_{uu}^{-1/2}$. However, for the "normal" case when there are no restrictions (like fixing some elements to zero) on the matrix $\boldsymbol{H}$, we will show below that the weight $\boldsymbol{J}_{uu}^{-1/2}$ does not have any effect on the optimal $\boldsymbol{H}$.

2.3.4 Optimal Sensitivities

Note from (2.6) that we can write $\boldsymbol{u}_{opt} = \boldsymbol{F}_u\boldsymbol{d}$ where $\boldsymbol{F}_u = -\boldsymbol{J}_{uu}^{-1}\boldsymbol{J}_{ud}$. More generally, we can write

$$\boldsymbol{y}^{\text{opt}} = \boldsymbol{F}\boldsymbol{d} \tag{2.12}$$

where $\boldsymbol{F}$ is the optimal sensitivity of the outputs (measurements) with respect to the disturbances. Here, $\boldsymbol{F}$ can be obtained using (2.2) and (2.6),

$$\boldsymbol{y}^{opt} = \boldsymbol{G}^y\boldsymbol{u}_{opt} + \boldsymbol{G}_d^y\boldsymbol{d} = (-\boldsymbol{G}^y\boldsymbol{J}_{uu}^{-1}\boldsymbol{J}_{ud} + \boldsymbol{G}_d^y)\boldsymbol{d}$$

that is,

$$\boldsymbol{F} = (-\boldsymbol{G}^y\boldsymbol{J}_{uu}^{-1}\boldsymbol{J}_{ud} + \boldsymbol{G}_d^y) \tag{2.13}$$

However, $\boldsymbol{J}_{uu}$ can be difficult to obtain, especially if one relies on numerical methods, and also taking the difference can introduce numerical inaccuracy. Thus, for practical use it is often better to obtain $\boldsymbol{F}$ from its definition, $\boldsymbol{F} = d\boldsymbol{y}^{opt}/d\boldsymbol{d}$, by numerically reoptimizing the model for the disturbances.

2.4 The Loss Method

Now we are finally ready to derive the main results.

2.4.1 The Loss Variable z as a Function of Disturbances and Noise

We start from the loss expression in (2.8) with $||\boldsymbol{z}||_2^2$ where $\boldsymbol{z} = \boldsymbol{J}_{uu}^{1/2}(\boldsymbol{u}-\boldsymbol{u}_{opt})$. We want to write $\boldsymbol{z}$ as a function of $\boldsymbol{d}$ and $\boldsymbol{n}^y$, The first step is to write $\boldsymbol{u}-\boldsymbol{u}_{opt}$ as a function of $\boldsymbol{c}-\boldsymbol{c}_{opt}$. We have $\boldsymbol{c} = \boldsymbol{H}\boldsymbol{y}$, so

$$\boldsymbol{c} = \boldsymbol{H}\boldsymbol{y} = \boldsymbol{H}\boldsymbol{G}^y\boldsymbol{u} + \boldsymbol{H}\boldsymbol{G}_d^y\boldsymbol{d}$$
$$\boldsymbol{c}_{opt} = \boldsymbol{H}\boldsymbol{y}^{opt} = \boldsymbol{H}\boldsymbol{G}^y\boldsymbol{u}_{opt} + \boldsymbol{H}\boldsymbol{G}_d^y\boldsymbol{d}$$

Thus, $\boldsymbol{c}-\boldsymbol{c}_{opt} = \boldsymbol{H}\boldsymbol{G}^y(\boldsymbol{u}-\boldsymbol{u}_{opt})$, or

$$(\boldsymbol{u}-\boldsymbol{u}_{opt}) = (\boldsymbol{H}\boldsymbol{G}^y)^{-1}(\boldsymbol{c}-\boldsymbol{c}_{opt})$$

where $\boldsymbol{G} = \boldsymbol{H}\boldsymbol{G}^y$ is the transfer function from $\boldsymbol{u}$ to $\boldsymbol{c}$.

The next step is to express $(\boldsymbol{c}-\boldsymbol{c}_{opt})$ as a function of $\boldsymbol{d}$ and $\boldsymbol{n}^y$. From (2.4) we have that $\boldsymbol{H}(\boldsymbol{y}+\boldsymbol{n}^y) = \boldsymbol{c}_s$ (constant), or

$$\boldsymbol{c} = \boldsymbol{H}\boldsymbol{y} = -\boldsymbol{H}\boldsymbol{n}^y + \boldsymbol{c}_s$$

Here, $\boldsymbol{c}_s = 0$, since we assume the nominal point is optimal. From (2.12) we have that $\boldsymbol{c}_{opt} = \boldsymbol{H}\boldsymbol{F}\boldsymbol{d}$. Since the signs for $\boldsymbol{n}^y$ and $\boldsymbol{d}$ do not matter for the expressions we derive below (we can have both positive and negative changes), we derive

$$\boldsymbol{c}-\boldsymbol{c}_{opt} = \boldsymbol{H}(\boldsymbol{F}\boldsymbol{d}+\boldsymbol{n}^y) = \boldsymbol{H}(\boldsymbol{F}\boldsymbol{W}_d\boldsymbol{d}' + \boldsymbol{W}_{\boldsymbol{n}^y}\boldsymbol{n}^{y'}) = \boldsymbol{H}[\boldsymbol{F}\boldsymbol{W}_d \;\; \boldsymbol{W}_{\boldsymbol{n}^y}]\begin{bmatrix}\boldsymbol{d}'\\ \boldsymbol{n}^{y'}\end{bmatrix}$$

Note that $\boldsymbol{W}_d$ and $\boldsymbol{W}_{\boldsymbol{n}^y}$ are usually diagonal matrices, representing the magnitude of the disturbances and measurement noises, respectively.

2.4.2 Loss for Given H, Disturbance and Noise (Analysis)

In summary, we have derived that for the given normalized disturbances $\boldsymbol{d}'$ and for the given normalized measurement noises $\boldsymbol{n}^{y'}$ the loss is given by

$$L = \frac{1}{2}\boldsymbol{z}^T\boldsymbol{z} \tag{2.14}$$

where

$$\boldsymbol{z} = \boldsymbol{J}_{uu}^{1/2}(\boldsymbol{u} - \boldsymbol{u}_{opt}) = \underbrace{\boldsymbol{J}_{uu}^{1/2}(\boldsymbol{H}\boldsymbol{G}^y)^{-1}\boldsymbol{H}\boldsymbol{Y}}_{\boldsymbol{M}(\boldsymbol{H})}\begin{bmatrix}\boldsymbol{d}'\\ \boldsymbol{n}^{y'}\end{bmatrix} \tag{2.15}$$

$$\boldsymbol{Y} = [\boldsymbol{F}\boldsymbol{W}_d \;\; \boldsymbol{W}_{\boldsymbol{n}^y}] \tag{2.16}$$

2.4.3 Worst-case and Average Loss for Given H (Analysis)

The above expressions give the loss for the given $\boldsymbol{d}$ and $\boldsymbol{n}^{y'}$, but the issue is the find the "magnitude" of the loss for the set bounded as

$$\left\|\begin{bmatrix}d'\\ n^{y'}\end{bmatrix}\right\|_2 \le 1 \tag{2.17}$$

Here "magnitude" can be defined in different ways, and the worst case loss (Halvorsen et al, 2003) and average loss (Kariwala et al, 2008) for a given $\boldsymbol{H}$ are given by

$$L_{wc} = \frac{1}{2}\bar{\sigma}(\boldsymbol{M})^2 \tag{2.18}$$

$$L_{avg} = \frac{1}{6(n_y + n_d)}\,\|\boldsymbol{M}\|_F^2 \tag{2.19}$$

where

$$\boldsymbol{M}(\boldsymbol{H}) = \boldsymbol{J}_{uu}^{1/2}(\boldsymbol{H}\boldsymbol{G}^y)^{-1}\boldsymbol{H}\boldsymbol{Y} \tag{2.20}$$

Here $\bar{\sigma}(\boldsymbol{M})$ denotes the singular value (induced 2-norm) of the matrix $\boldsymbol{M}(\boldsymbol{H})$, and $\|\boldsymbol{M}\|_F$ denotes the Frobenius norm (normal 2-norm) of the matrix $\boldsymbol{M}$. Use of the norm of $\boldsymbol{M}$ to analyze the loss is known as the "exact local method".

2.4.4 Loss Method for Finding Optimal H

The optimal $\boldsymbol{H}$ can be found by minimizing either the worst-case loss (2.18) or the average loss (2.19). Fortunately, (Kariwala et al, 2008) prove that the $\boldsymbol{H}$ that minimizes the average loss in equation (2.19) is super optimal, in the sense that the same $\boldsymbol{H}$ minimizes the worst case loss in (2.18). Hence, only minimization of the Frobenius norm in (2.19) is considered in the rest of the paper. The scaling factor $\frac{1}{6(n_y+n_d)}$ does not have any effect on the solution of the equation (2.19) and hence it is omitted in the problem formulation. Similarly, the square does not effect the optimal solution and can be omitted.

In summary, the problem is to find the combination matrix $\boldsymbol{H}$ that minimizes the Frobenius norm of $||\boldsymbol{M}||$ in (2.19), that is,

$$\min_{\boldsymbol{H}} \left\| \boldsymbol{J}_{uu}^{1/2}(\boldsymbol{H}\boldsymbol{G}^y)^{-1}\boldsymbol{H}\boldsymbol{Y} \right\|_F \tag{2.21}$$

where $\boldsymbol{Y} = [\boldsymbol{F}\boldsymbol{W}_d \;\; \boldsymbol{W}_{\boldsymbol{n}^y}]$. We call this the minimum loss method for finding optimal linear measurement combinations, $\boldsymbol{c} = \boldsymbol{H}\boldsymbol{y}$.

The objective in (2.21) is to find the nonsquare $n_u \times n_y$ matrix $\boldsymbol{H}$ (note that $n_u = n_c$). In most cases it may be recast as a convex optimization problem as given in (2.23) below. The exception is if $\boldsymbol{H}$ has a specified structure, for example, $\boldsymbol{H}$ is a selection matrix, which is discussed in Section 2.6.

Further Comments

1. Using the norm of $\boldsymbol{M}$ to analyze the loss is known as the "exact local method" and finding the optimal $\boldsymbol{H}$ is the "exact local method optimization problem". However, in this paper we simply call it the "loss method".
2. To include changes in the weights in the cost function $\boldsymbol{p}$ (prices), we need to find the optimal sensitivity to price changes, $\boldsymbol{y}^{opt} = \boldsymbol{F}_p\boldsymbol{p}$ The corrected setpoint for the variables $\boldsymbol{c} = \boldsymbol{H}\boldsymbol{y}$ is then

$$\boldsymbol{c}_s = \boldsymbol{H}\boldsymbol{y}^{opt} = \boldsymbol{H}\boldsymbol{F}_p\boldsymbol{p} \tag{2.22}$$

3. The effect (transfer function) from $\boldsymbol{c}_s$ to $\boldsymbol{z}$ is $\boldsymbol{M}_n = \boldsymbol{J}_{uu}^{1/2}(\boldsymbol{H}\boldsymbol{G}^y)^{-1}$, and from $\boldsymbol{c}_s$ to $\boldsymbol{u}$ is $\boldsymbol{G}^{-1} = (\boldsymbol{H}\boldsymbol{G}^y)^{-1}$. Since there are extra degrees of freedom in $\boldsymbol{H}$ which are not set by the optimization problem, either of these ($\boldsymbol{M}_n$ or $\boldsymbol{G}$) can be selected freely; see below for details.

Exercise 2.1. Consider a scalar case ($n_u = n_c = 1$) with no disturbances ($\boldsymbol{F} = 0$) and assume that the measurements $\boldsymbol{y}$ have been scaled such that $\boldsymbol{W}_{\boldsymbol{n}^y} = I$ (noise of equal magnitude on all outputs). For the scalar case, $\boldsymbol{J}_{uu}^{1/2}$ does not matter for the optimization problem which becomes $\min_{\boldsymbol{H}} \left|\left|(\boldsymbol{H}\boldsymbol{G}^y)^{-1}\boldsymbol{H}\right|\right|_F^2$ and we want to find the optimal $\boldsymbol{H}$.

(a) Consider the case with 2 measurements, so $\boldsymbol{G}^y = \begin{bmatrix} g_1 \\ g_2 \end{bmatrix}$ (column vector) and $\boldsymbol{H} = [h_1 \; h_2]$ (row vector), and solve the problem analytically. Also compute the optimal norm, $j = \left|\left|(\boldsymbol{H}\boldsymbol{G}^y)^{-1}\boldsymbol{H}\right|\right|_F$.

(b) Derive the result more generally for the scalar case with any number of measurements, by making use of the definition of the induced 2-norm (singular value), which for a vector gives, $||\boldsymbol{G}^y||_2 = \sqrt{\boldsymbol{G}^{y^T}\boldsymbol{G}^y} = \max_{\boldsymbol{h}} \frac{||\boldsymbol{G}^y\boldsymbol{h}||_2}{||\boldsymbol{h}||_2}$

(note that for a vector the Frobenius norm ($_F$)and 2-norm ($_2$) are the same).

(c) Use the analytical formula presented below, to derive the general result for the multivariable case ($n_u = n_c > 1$).

Solution 2.1.

(a) Two measurements.

$$\boldsymbol{H}\boldsymbol{G}^y = [h_1 h_2] \begin{bmatrix} g_1 \\ g_2 \end{bmatrix} = h_1 g_1 + h_2 g_2 \quad \text{(scalar)}$$

$$j^2 = \left|\left|(\boldsymbol{H}\boldsymbol{G}^y)^{-1}\boldsymbol{H}\right|\right| ||_F^2 = (h_1^2 + h_2^2)/(h_1 g_1 + h_2 g_2)^2 = (1+x^2)/(g_1 + x g_2)^2$$

where $x = h_2/h_1$. Set $d(j^2)/dx = 0$ to find the optimal x. After a little simplification $x - (1+x^2)g_2/(g_1 + x g_2) = 0$ which gives $x = g_2/g_1$. Conclusion:

$$h_2/h_1 = g_2/g_1$$

that is, we prefer to control the measurements corresponding to large elements in $\boldsymbol{G}^y$. Also find: $j^{opt} = 1/\sqrt{(\boldsymbol{g}_1^2 + \boldsymbol{g}_2^2)}$ which we note is equal to $1/\left|\left|\boldsymbol{G}^y\right|\right|_F$

(b) Any number of measurements: Let $\boldsymbol{h} = \boldsymbol{H}^T$ be a column vector. $\boldsymbol{G}^y$ is already a column vector. Since $\boldsymbol{H}\boldsymbol{G}^y = \boldsymbol{h}^T\boldsymbol{G}^y$ is a scalar, it is equal to its transpose and we have that $\boldsymbol{H}\boldsymbol{G}^y = \boldsymbol{G}^{y^T}\boldsymbol{h}$. Our optimization problem then becomes

$$\min_{\boldsymbol{H}} j = \min_{\boldsymbol{h}} \left|\left| \frac{\boldsymbol{h}}{(\boldsymbol{G}^{y^T}\boldsymbol{h})} \right|\right|_F = 1/\left(\max_{\boldsymbol{h}} \frac{\left|\left|\boldsymbol{G}^{y^T}\boldsymbol{h}\right|\right|_2}{||\boldsymbol{h}||_2} \right) = 1/\left|\left|\boldsymbol{G}^y\right|\right|_2$$

We have here made use of the induced 2-norm and the fact that both the Frobenius- and 2-norm are the same for a vector. Thus the optimal j is the inverse of the 2-norm of $\boldsymbol{G}^y$, which generalizes the solution found for the case with two measurements. The optimal $\boldsymbol{h} = c\boldsymbol{G}^y$ (where c is any scalar since only the relative magnitudes matter), that is,

$$\boldsymbol{H}^T = c\boldsymbol{G}^y$$

which generalizes the result above.

(c) Multivariable case (c is no longer required to be a scalar). From (2.25) we derive with $\boldsymbol{Y} = I$ ($\boldsymbol{F} = 0$ and measurement noise of magnitude 1 for all outputs) that an optimal solution is

$$\boldsymbol{H}^T = \boldsymbol{G}^y$$

which generalizes the results above. Thus, for the case where only measurement noise is a concern, and all the measurements have the same noise magnitude, the optimal is to select the measurements according the gain matrix $\boldsymbol{G}^y$. This means that "sensitive" measurements, with large elements in $\boldsymbol{G}^y$ are preferred for control.

2.5 Reformulation of Loss Method to Convex Problem and Explicit Solution

We consider here the "normal" case where $\boldsymbol{H}$ is a "full" matrix (with no structural constraints).

Theorem 2.1 (Reformulation as a convex problem). *The problem in equation (2.21) may seem non-convex, but for the normal case where $\boldsymbol{H}$ is a "full" matrix (with no structural constraints), it can be reformulated as a constrained quadratic programming problem (Alstad et al, 2009)*

$$\begin{aligned} &\min_{\boldsymbol{H}} ||\boldsymbol{HY}||_F \\ \text{s.t. } &\boldsymbol{HG}^y = \boldsymbol{J}_{uu}^{1/2} \end{aligned} \tag{2.23}$$

Proof. From the original problem in equation (2.21), we have that the optimal solution $\boldsymbol{H}$ is non-unique because if $\boldsymbol{H}$ is a solution then $\boldsymbol{H}_1 = \boldsymbol{DH}$ is also a solution for any non-singular matrix $\boldsymbol{D}$ of size $n_c \times n_c$. This follows because

$$\boldsymbol{J}_{uu}^{1/2}(\boldsymbol{H}_1\boldsymbol{G}^y)^{-1}\boldsymbol{H}_1\boldsymbol{Y} = \boldsymbol{J}_{uu}^{1/2}(\boldsymbol{HG}^y)^{-1}\boldsymbol{D}^{-1}\boldsymbol{DHY} = \boldsymbol{J}_{uu}^{1/2}(\boldsymbol{HG}^y)^{-1}\boldsymbol{HY}$$

One implication is that we can freely choose $\boldsymbol{G} = \boldsymbol{HG}^y$, which is a $n_c \times n_c$ matrix representing the effect of $\boldsymbol{u}$ on $\boldsymbol{c}$ ($\boldsymbol{c} = \boldsymbol{Gu}$). Thus, in (2.21) we may use the non-uniqueness of $\boldsymbol{H}$ to set the first part of the expression equal to the identity matrix, which is equivalent to setting $\boldsymbol{HG}^y = \boldsymbol{J}_{uu}^{1/2}$. This identity must then be added as a constraint in the optimization as shown in (2.23). □

The reason for the non-uniqueness is that since $n_y \geq n_c$, $\boldsymbol{H}$ is "fat" $n_c \times n_y$ matrix (with more columns than rows).

Theorem 2.2 (Analytical solution). *Under the assumption that $\boldsymbol{YY}^T$ is full rank, an analytical solution for the problem in (2.23) is (Alstad et al, 2009)*

$$\boldsymbol{H}^T = (\boldsymbol{YY}^T)^{-1}\boldsymbol{G}^y(\boldsymbol{G}^{y^T}(\boldsymbol{YY}^T)^{-1}\boldsymbol{G}^y)^{-1}\boldsymbol{J}_{uu}^{1/2} \tag{2.24}$$

Proof. The result is proved in (Alstad et al, 2009) and is based on first vectorizing the problem and then using standard results from constrained quadratic optimization.

The analytical solution in Theorem 2.2, results in a $\boldsymbol{H}$ satisfying $\boldsymbol{HG}^y = \boldsymbol{J}_{uu}^{1/2}$. However, recall that the optimal solution $\boldsymbol{H}$ is non-unique, and we may use it to derive a simplified analytical solution.

Theorem 2.3 (Simplified analytical solution). *Under the assumption that $\boldsymbol{Y}\boldsymbol{Y}^T$ is full rank, another analytical solution for the problem in (2.23) is*

$$\boldsymbol{H}^T = (\boldsymbol{Y}\boldsymbol{Y}^T)^{-1}\boldsymbol{G}^y \tag{2.25}$$

Proof. This follows trivially from Theorem 2.2, since if $\boldsymbol{H}^T$ is a solution then so is $\boldsymbol{H}_1^T = \boldsymbol{H}^T\boldsymbol{D}$ and we simply select $\boldsymbol{D} = ({\boldsymbol{G}^y}^T(\boldsymbol{Y}\boldsymbol{Y}^T)^{-1}\boldsymbol{G}^y)^{-1}\boldsymbol{J}_{uu}^{1/2} = \boldsymbol{J}_{uu}^{-1/2}{\boldsymbol{G}^y}^T(\boldsymbol{Y}\boldsymbol{Y}^T)^{-1}\boldsymbol{G}^y$, which is a $n_c \times n_c$ matrix. □

Note that the analytical expressions in Theorems 2.2 and 2.3 require $\boldsymbol{Y}\boldsymbol{Y}^T$ to be full rank. This implies that they generally do not apply to the case with no measurement error, $\boldsymbol{W}_{n^y} = 0$, but otherwise they apply for any number of measurements. One exception (but not so common in practice), when the analytical expressions for $\boldsymbol{H}$ do apply also for $\boldsymbol{W}^y = 0$, is when $n_y \leq n_d$, because $\boldsymbol{Y}\boldsymbol{Y}^T$ then remains full rank.

Corollary 2.1 (Important insight). *Theorem 2.3 gives the very important insight that $\boldsymbol{J}_{uu}$ is not needed for finding the optimal $\boldsymbol{H}$, provided we have the normal case where $\boldsymbol{H}$ can be any $n_c \times n_y$ matrix.*

This means that in (2.21) we can replace $\boldsymbol{J}_{uu}^{1/2}$ by any non-singular matrix, and still get an optimal $\boldsymbol{H}$. This can greatly simplify practical calculations, because $\boldsymbol{J}_{uu}$ may be difficult to obtain numerically because it involves the second derivative. On the other hand, we found that $\boldsymbol{F}$, which enters in $\boldsymbol{Y}$, is relatively straightforward to obtain numerically. Although $\boldsymbol{J}_{uu}$ is not needed for finding the optimal $\boldsymbol{H}$, it would be required for finding a numerical value for the loss.

The analytical solutions are useful, in particular for their insights they yield, but for practical calculations it is usually faster and more robust to compute the optimal $\boldsymbol{H}$ by solving the convex quadratic optimization problems. In addition, the convex optimization problems do not need the requirement that $\boldsymbol{Y}\boldsymbol{Y}^T$ is non-singular. Based on the insight in Corollary 2.1, the quadratic optimization in Theorem 2.1 (Alstad et al, 2009), can be further reformulated to a more general form (Yelchuru and Skogestad, 2010)

Theorem 2.4 (Generalized convex formulation). *An optimal $\boldsymbol{H}$ for the problem in (2.23) is*

$$\begin{aligned} &\min_{\boldsymbol{H}} ||\boldsymbol{H}\boldsymbol{Y}||_F \\ &\text{s.t. } \boldsymbol{H}\boldsymbol{G}^y = \boldsymbol{Q} \end{aligned} \tag{2.26}$$

where $\boldsymbol{Q}$ is any non-singular $n_c \times n_c$ matrix, for example, $\boldsymbol{Q} = I$, but $\boldsymbol{Q}$ must be fixed while minimizing $||\boldsymbol{H}\boldsymbol{F}||_F$.

Proof. The result follows from Corollary 2.1, but can more generally be derived as follows. The problem in (2.23) is to minimize $\left\|\underbrace{(\boldsymbol{J}_{uu}^{1/2}(\boldsymbol{H}\boldsymbol{G}^y)^{-1}}_{\boldsymbol{X}}\boldsymbol{H}\boldsymbol{Y})\right\|_F$.

The reason why we can omit the $n_c \times n_c$ matrix $\boldsymbol{X}$, is that if $\boldsymbol{H}$ is an optimal solution then so is $\boldsymbol{H}_1 = \boldsymbol{D}\boldsymbol{H}$ where $\boldsymbol{D}$ is any nonsingular $n_c \times n_c$ (see proof of Theorem 2.1). However, note that the matrix $\boldsymbol{X}$, or equivalently the matrix $\boldsymbol{Q}$, must be fixed during the optimization, so it needs to be added as a constraint. □

The fact that $\boldsymbol{Q}$ can be chosen freely (Theorem 2.4) can be useful for numerical reasons, or finding improved bounds for cases with constraints on H (see below).

Once we have found an optimal $\boldsymbol{H}$ using any of the Theorems above, we can use the non-uniqueness of optimal $\boldsymbol{H}$ to find another $\boldsymbol{H}_1 = \boldsymbol{D}\boldsymbol{H}$ with desired property or structure. For example, one can select $\boldsymbol{D}$ such that $\boldsymbol{G} = \boldsymbol{H}\boldsymbol{G}^y = I$. Alternatively, one can specify selected elements in $\boldsymbol{H}_1$, for example, $\boldsymbol{H}_1 = [I \quad \boldsymbol{K}]$. In the latter case, write $\boldsymbol{H} = [\boldsymbol{H}_l\boldsymbol{H}_r]$ and assume $\boldsymbol{H}_l$ is full rank, then $\boldsymbol{H}_1 = [I \quad \boldsymbol{K}] = [\boldsymbol{D}\boldsymbol{H}_l \quad \boldsymbol{D}\boldsymbol{H}_r]$, and we find $\mathrm{d}_b = \boldsymbol{H}_l^{-1}$ and $\boldsymbol{K} = \boldsymbol{H}_l^{-1}\boldsymbol{H}_r$.

Required information

To find the optimal "full" $\boldsymbol{H}$ using the loss method we need four pieces of information. First, for the measurements we need the optimal disturbance sensitivity ($\boldsymbol{F}$) and input sensitivity (G^y). These are obtained from the model. Next, we must specify the disturbance magnitudes (W_d) and the noise magnitudes (W_{n^y}). The matrix $\boldsymbol{J}_{uu}$ is not needed except when there are structural constraints, as discussed in the next section.

Note that changes (disturbances) in the prices (parameters) in the cost function do not change the optimal $\boldsymbol{H}$, based on the assumption that we still have a quadratic optimization problem with constant weights. However, as given in (2.22) the setpoint for $\boldsymbol{c}$ needs to be adjusted, $\boldsymbol{c}_s = \boldsymbol{H}\boldsymbol{F}_p\boldsymbol{p}$ and for this we need for the measurements the optimal price sensitivity ($\boldsymbol{F}_p$) which can be obtained from the model.

2.6 Structural Constraints on $\boldsymbol{H}$

In the previous section we considered the normal case where $\boldsymbol{H}$ may be any "full" matrix. In terms of selecting controlled variables, $\boldsymbol{c} = \boldsymbol{H}\boldsymbol{y}$, this means that any combination of measurements are allowed. However, in practice there may be constraints on $\boldsymbol{H}$, for example, one wants to use a subset of the measurements or one want to use a decentralized structure for $\boldsymbol{H}$.

We will consider the following special cases

Case 1. No restrictions on $\boldsymbol{H}$. This is the case already considered where Theorems 2.1– 2.4 hold.

Note that key for deriving Theorems 2.1– 2.4 was that if $\boldsymbol{H}$ is a solution then so is $\boldsymbol{H}_1 = \boldsymbol{D}\boldsymbol{H}$ where $\boldsymbol{D}$ is any non-singular matrix.

Case 2. $\boldsymbol{H}$ contains a subset of the measurements but is otherwise full. Theorems 2.1– 2.4 hold also in this case.

The reason is that $\boldsymbol{H}_1 = \boldsymbol{D}\boldsymbol{H}$ will have the same structure as $\boldsymbol{H}$ for any nonsingular $\boldsymbol{D}$. This is because if $\boldsymbol{H}$ has columns equal to zero, then these columns will remain zero in $\boldsymbol{D}\boldsymbol{H}$ even if $\boldsymbol{D}$ is "full". For example, if $\boldsymbol{H} = \begin{bmatrix} 0 & \boldsymbol{x} & 0 & \boldsymbol{x} & \boldsymbol{x} \\ 0 & \boldsymbol{x} & 0 & \boldsymbol{x} & \boldsymbol{x} \end{bmatrix}$ then we can still allow a full $\boldsymbol{D} = \begin{bmatrix} \boldsymbol{x} & \boldsymbol{x} \\ \boldsymbol{x} & \boldsymbol{x} \end{bmatrix}$ (where $\boldsymbol{x}$ is any number) and keep the structure of $\boldsymbol{H}$ in $\boldsymbol{H}_1 = \boldsymbol{D}\boldsymbol{H}$.

Case 3. $\boldsymbol{H}$ contains measurements from disjoint set, so $\boldsymbol{H}$ has a block diagonal (decentralized) structure. Theorems 2.1– 2.4 do *not* hold in this case.

The reason is that for $\boldsymbol{H}_1 = \boldsymbol{D}\boldsymbol{H}$ to have the same structure, $\boldsymbol{D}$ must have a structure similar to $\boldsymbol{H}$. For example, let $\boldsymbol{H} = \begin{bmatrix} \boldsymbol{x} & \boldsymbol{x} & \boldsymbol{x} & 0 & 0 \\ 0 & 0 & 0 & \boldsymbol{x} & \boldsymbol{x} \end{bmatrix}$ then $\boldsymbol{D} = \begin{bmatrix} \boldsymbol{x} & 0 \\ 0 & \boldsymbol{x} \end{bmatrix}$ (where x is any number) and if $\boldsymbol{H} = \begin{bmatrix} \boldsymbol{x} & \boldsymbol{x} & \boldsymbol{x} & \boldsymbol{x} & \boldsymbol{x} \\ 0 & 0 & 0 & \boldsymbol{x} & \boldsymbol{x} \end{bmatrix}$ then $\boldsymbol{D} = \begin{bmatrix} \boldsymbol{x} & \boldsymbol{x} \\ 0 & \boldsymbol{x} \end{bmatrix}$.

Thus, for case 3 we do not have a convex problem formulation, that is, we need to solve the nonconvex problem in (2.21) (with additional constraints on the structure of $\boldsymbol{H}$). This is not surprising as decentralized control is generally a nonconvex problem. Nevertheless, Theorems 2.1 and 2.4, with additional constraints on the structure of $\boldsymbol{H}$, give convex optimization problems that provide upper bounds on the optimal $\boldsymbol{H}$ for case 3. In particular, in Theorem 2.4 4 we may make use of the extra degree of freedom provided by the matrix $\boldsymbol{Q}$ (Yelchuru and Skogestad, 2010).

Also note that, as opposed to cases 1 and 2, J_{uu} is needed to find the optimal solution for case 3. This may seem a bit surprising.

Case 4. Decentralized control using single measurements, that is $n_{ys} = n_c$ where n_{ys} is the number of selected measurements). Theorems 2.1– 2.4 hold also in this case.

This is a special case of case 3 where we use the fewest number of measurements. This case is different from case 2 in that $\boldsymbol{H}$ is a diagonal matrix. The reason why Theorems 2.1– 2.4 hold in this case, is that we can still keep $\boldsymbol{D}$ full because the "non-zero" part of $\boldsymbol{H}$ is square and we can to change it to anything, so we can treat it a special case of "full $\boldsymbol{H}$".

2.7 Some Special Cases: Nullspace Method and Maximum Gain Rule

The general optimization problem is

$$\min_{\boldsymbol{H}} \left|\left| \boldsymbol{J}_{uu}^{1/2}(\boldsymbol{H}\boldsymbol{G}^y)^{-1}\boldsymbol{H}\boldsymbol{Y} \right|\right|_F \tag{2.27}$$

where $\boldsymbol{Y} = [\boldsymbol{F}\boldsymbol{W}_d\boldsymbol{W}_{n^y}]$. The objective is to find the nonsquare $n_c \times n_y$ matrix $\boldsymbol{H}$ (note that $n_u = n_c$). We will here consider some special cases of this problem, which historically were developed before the convex and analytical solutions presented above.

2.7.1 No Measurement Noise: Nullspace Method ("full *H*")

For the special case with no measurement noise, $\boldsymbol{W}_{ny} = 0$, and with more (independent) measurements than (independent) inputs and disturbances, $n_y \geq n_u + n_d$, it is possible to find $\boldsymbol{H}$ such that

$$\boldsymbol{H}\boldsymbol{F} = 0 \tag{2.28}$$

that is, the loss is zero. This is called the "nullspace method" (Alstad and Skogestad, 2007) because $\boldsymbol{H}$ is in the nullspace of $\boldsymbol{F}$. In this case, $\boldsymbol{G}_y$ and $\boldsymbol{W}_d$ do not matter for finding the optimal $\boldsymbol{H}$.

The nullspace method is very simple and has been found to be very useful in applications. Since the nullspace method neglects the effect of measurement error, it is important to use preselect a subset of the measurements that are expected to be insensitive to measurement errors.

Also, one cannot include too many disturbances, because otherwise one cannot satisfy the requirement $n_y \geq n_u + n_d$.

One limitation with the analytical formulas in (2.24) and (2.25) is that they do not give the nullspace method as a special case. This is because $\boldsymbol{Y} = [\boldsymbol{F}\boldsymbol{W}_d \;\; \boldsymbol{W}_{\boldsymbol{n}^y}]$ at most has rank n_d when $\boldsymbol{W}_{n^y} = 0$. Thus, the $n_y \times n_y$ matrix $\boldsymbol{Y}\boldsymbol{Y}^T$ at most has rank n_d and is not invertible because this would require the rank to be n_y. However, the convex optimization problems in Theorems 2.1 and 2.4 do give the nullspace method as a special case.

Comment: In general, with measurement noise included or with few measurements (so $n_y < n_u + n_d$), it is not possible to make $\boldsymbol{H}\boldsymbol{Y}$ zero.

Explicit Expression for *H* for Nullspace Method

The following explicit expression applies for $\boldsymbol{H}$ (Alstad and Skogestad, 2007):

$$\boldsymbol{H} = [\boldsymbol{J}_{uu}\boldsymbol{J}_{ud}](\tilde{\boldsymbol{G}}^y)^{-1} \tag{2.29}$$

Proof. Here is a proof which is much simpler than that given in (Alstad and Skogestad, 2007): Want to find $\boldsymbol{c} = \boldsymbol{H}\boldsymbol{y}$ with zero loss.

1. Measurement relationship: $\boldsymbol{y} = \tilde{\boldsymbol{G}}^y \begin{bmatrix} \boldsymbol{u} \\ \boldsymbol{d} \end{bmatrix}$. Inverting this:
$$\begin{bmatrix} \boldsymbol{u} \\ \boldsymbol{d} \end{bmatrix} = (\tilde{\boldsymbol{G}}^y)^{-1}\boldsymbol{y} \tag{2.30}$$
2. Optimality condition (NCO):
$$\boldsymbol{J}_u = 0 \tag{2.31}$$
3. First-order expansion of gradient:
$$\boldsymbol{J}_u = \boldsymbol{J}_u^* + \boldsymbol{J}_{uu}^*\boldsymbol{u} + \boldsymbol{J}_{ud}^*\boldsymbol{d} = [\boldsymbol{J}_{uu}^* \quad \boldsymbol{J}_{ud}^*] \begin{bmatrix} \boldsymbol{u} \\ \boldsymbol{d} \end{bmatrix}$$
where we use $\boldsymbol{J}_u* = 0$.
4. (2.30) and (2.31) then give: $[\boldsymbol{J}_{uu} \quad \boldsymbol{J}_{ud}]\tilde{\boldsymbol{G}}_y^{-1}\boldsymbol{y} = 0$ and it follows that $\boldsymbol{H} = [\boldsymbol{J}_{uu} \quad \boldsymbol{J}_{ud}](\tilde{\boldsymbol{G}}^y)^{-1}$. □

2.7.2 No Disturbances

The case with no disturbances has limited practical significance, but is nevertheless and interesting limiting cases.

We assume there are no disturbances, $\boldsymbol{W}_d = 0$, and we scale the measurements $\boldsymbol{y}$ so that they all have unity measurement noise, $\boldsymbol{W}_{n^y} = I$. From the analytical expression (2.25), we then have that an optimal solution is
$$\boldsymbol{H}^T = \boldsymbol{G}^y \tag{2.32}$$
This gives the important insight that we prefer sensitive measurements.

2.7.3 An Approximate Analysis Method for the General Case: "Maximum Gain Rule"

The maximum gain rule is an approximate analysis method for a given $\boldsymbol{H}$. If we want to compare (analyze) alternative choices for $\boldsymbol{H}$, for example, alternative choices of individual measurements, then the "maximum gain rule" is effective and gives considerable insight. The maximum gain rule has also been used to find "optimal" $\boldsymbol{H}$, especially for the case where one wants to control individual measurements, and Yu and Kariwala have devised efficient branch and bound methods for solving this problem.

In the "maximum gain rule" one considers the scaled gain matrix $\boldsymbol{G} = \boldsymbol{H}\boldsymbol{G}^y$ from $\boldsymbol{u}$ to $\boldsymbol{c}$. To derive the maximum gain rule, we return to the loss expression

$$J = \frac{1}{2}\boldsymbol{z}^T\boldsymbol{z}$$

where

$$\boldsymbol{z} = \boldsymbol{J}_{uu}^{1/2}(\boldsymbol{u} - \boldsymbol{u}_{opt}) = \boldsymbol{J}_{uu}^{1/2}\boldsymbol{G}^{-1}(\boldsymbol{c} - \boldsymbol{c}_{opt})$$
$$\boldsymbol{c} - \boldsymbol{c}_{opt} = \boldsymbol{H}(\boldsymbol{F}\boldsymbol{d} + \boldsymbol{n}^y)$$
$$\boldsymbol{G} = \boldsymbol{H}\boldsymbol{G}^y$$

Here, $\boldsymbol{c} - \boldsymbol{c}_{opt}$ may be viewed as the "optimal" (or expected) variation in the selected variables, $\boldsymbol{c} = \boldsymbol{H}\boldsymbol{y}$, caused by disturbances and measurement noise. The magnitude of $\boldsymbol{c} - \boldsymbol{c}_{opt} = \boldsymbol{H}\boldsymbol{F}\boldsymbol{d} + \boldsymbol{H}\boldsymbol{n}^y$ is obtained by adding the magnitude of the contributions from $\boldsymbol{H}\boldsymbol{F}\boldsymbol{d}$ and $\boldsymbol{H}\boldsymbol{n}^y$, and we assume in the following that $\boldsymbol{c} - \boldsymbol{c}_{opt} = \boldsymbol{W}_c\boldsymbol{c}'$ where $\boldsymbol{W}_c$ is a diagonal matrix for the expected optimal variation ("optimal span") in $\boldsymbol{c}$ and we assume that all $||\boldsymbol{c}'||^2 \leq 1$ are allowed. $\boldsymbol{c}-\boldsymbol{c}_{opt}$ translates into changes in the inputs $(\boldsymbol{u}-\boldsymbol{u}_{opt})$ by the transformation $\boldsymbol{u} = \boldsymbol{G}^{-1}\boldsymbol{c}$ and to a loss through the matrix $\boldsymbol{J}_{uu}^{1/2}$. We want $(\boldsymbol{u} - \boldsymbol{u}_{opt})$ small, so we want the norm of $\boldsymbol{G}^{-1}$ small. More specifically, the largest (worst-case) value of $||z||_2$ for any allowed $||\boldsymbol{c}'||_2 \leq 1$ is equal to $\bar{\sigma}(\boldsymbol{J}_{uu}^{1/2}\boldsymbol{G}^{-1}\boldsymbol{W}_{\boldsymbol{c}})$, and we want this as small as possible. From singular value properties we have that the $\bar{\sigma}(\boldsymbol{A}^{-1}) = 1/\underline{\sigma}(\boldsymbol{A})$, that is we want to maximize smin$(\boldsymbol{W}_{\boldsymbol{c}}^{-1}\boldsymbol{G}\boldsymbol{J}_{uu}^{-1/2})$.

We have then derived the **maximum gain rule:** *Under the assumption that $||c'||_2 \leq 1$, the worst-case loss is given by $L_{max} = \frac{1}{2}\frac{1}{\underline{\sigma}^2(\boldsymbol{G}_s)}$ where*

$$\boldsymbol{G}_s = S_1\boldsymbol{G}S_2 \tag{2.33}$$

and

$$S_1 = \boldsymbol{W}_{\boldsymbol{c}}^{-1} = diag(1/|\boldsymbol{c}_i - \boldsymbol{c}_{opt,i}|)$$
$$S_2 = \boldsymbol{J}_{uu}^{-1/2}$$

Note that S_1 includes the sum of the optimal variation (as given by the $\boldsymbol{F}$-matrix) and the expected measurement error. Thus, to minimize the loss we should select $\boldsymbol{c} = \boldsymbol{G}\boldsymbol{u}$ with a large minimum singular value of the scaled gain matrix $\boldsymbol{G}_s$.

The only "non-exact" step in deriving this rule comes from the assumption that all $||\boldsymbol{c}'||_2 \leq 1$ are allowed, which means that we neglect some of the variations in $(\boldsymbol{c}-\boldsymbol{c}_{opt})$ that are correlated. Nevertheless, since the presence of measurement noise means that there is always some uncorrelated variation, at least if we consider individual measurements, $\boldsymbol{c} = \boldsymbol{y}$, this implies that we can safely exclude candidate $\boldsymbol{c}$'s with a small gain, that is, with a small value of $\underline{\sigma}(\boldsymbol{G}_s)$.

Note that $\boldsymbol{J}_{uu}$ enters into the maximum gain rule, whereas it is actually not required when $\boldsymbol{H}$ is the optimal full matrix, see (2.26). However, in general $\boldsymbol{J}_{uu}$ must be included, see case 3 in the discussion following (2.26).

- Do we need the maximum gain rule?

One can analyze the loss with alternatives choices for $\boldsymbol{c}$ using the "exact local method", so why do we need the maximum gain rule? The motivation for using the maximum gain rule is at least threefold

1. It is simpler to compute.
2. It given insight, in particular that we want to control "sensitive" variables with a large scaled gain: *Select variables* $\boldsymbol{c}$ *where the "optimal variation"* $(\boldsymbol{c} - \boldsymbol{c}_{opt})$ *(from disturbances, and including measurement noise) is small compared to the "achievable variation"*$(\boldsymbol{c} = \boldsymbol{G}\boldsymbol{u})$ *(from inputs).*
3. The scaled gain matrix is $\boldsymbol{G}_s = S_1 \boldsymbol{G} S_2$. Here, the gain matrix $\boldsymbol{G}$ is obtained by linearizing in a single operating point. To find the "scaling" matrices S_1 and S_2 we need to reoptimize for the disturbances (to find $\boldsymbol{c} - \boldsymbol{c}_{opt}$ needed for S_1) and to find the second derivative with respect to the inputs (to find $S_2 = \boldsymbol{J}_{uu}^{-1/2}$), which can be rather involved calculations. If this information missing, then one may often get good results by estimating the optimal variations to find S_1 (for example, based on operating data) and by setting $S_2 = I$ (one should in this case scale the inputs so that their expected effect on the cost is similar).

This maximum gain rule has also proven to work surprisingly well on many applications. Nevertheless, if one has data for the optimal sensitivity $(\boldsymbol{F})$, then our recommendation is to use the "exact local method" instead of the maximum gain rule. This is because one can analyze alternative choices for $\boldsymbol{c}$ (and $\boldsymbol{H}$) more exactly by computing the norm (Frobenius norm or max. singular value) of $\boldsymbol{M} = \left[\boldsymbol{J}_{uu}^{1/2}(\boldsymbol{H}\boldsymbol{G}^y)^{-1}\boldsymbol{H}\boldsymbol{Y}\right]$. Kariwala and Cao (2009) have derived efficient branch and bound algorithms for finding the measurement choice (optimal structured $\boldsymbol{H}$) that minimize either the norm of $\boldsymbol{M}$ as well as the scaled gain, $\underline{\sigma}(\boldsymbol{G}_s)$. Although the computation times for minimizing the latter are somewhat lower, the benefit is not sufficiently large to justify using the maximum gain rule, provided we have the necessary information available for the first method.

2.8 Indirect Control and Estimation of Primary Variable

These two problems are very similar, and can be written as a special case of the loss method, involving the same matrices.

2.8.1 Indirect Control of $\boldsymbol{y}_1$

The objective is to keep the primary output $\boldsymbol{y}_1$ close to its setpoint, so the cost function is

$$J = (\boldsymbol{y}_1 - \boldsymbol{y}_{1,s})^2$$

However, $\boldsymbol{y}_1$ is not measured, but we have to use some other measurements $\boldsymbol{y}$. Thus, we want to achieve indirect control of $\boldsymbol{y}_1$ by keeping $\boldsymbol{c} = \boldsymbol{H}\boldsymbol{y}$ at a given setpoint.

To find the optimal "full" $\boldsymbol{H}$ using the loss method we need four pieces of information; $\boldsymbol{F}, G^y, W_d, W_{n^y}$. In our case, the optimal sensitivity is $\boldsymbol{F} = (d\boldsymbol{y}^{opt}/d\boldsymbol{d}) = (d\boldsymbol{y}/d\boldsymbol{d})_{\boldsymbol{y}_1}$ It may be obtained by simulations where we keep $\boldsymbol{y}_1$ constant for the various disturbances. Instead of using simulations, we may write $\boldsymbol{y}_1 = \boldsymbol{G}_1\boldsymbol{u} + \boldsymbol{G}_{d1}\boldsymbol{d}$, and then (Hori et al, 2005) $\boldsymbol{J}_{uu} = \boldsymbol{G}_1^t\boldsymbol{G}_1$, $\boldsymbol{J}_{ud} = \boldsymbol{G}_1^T\boldsymbol{G}_{d_1}$, $\boldsymbol{F} = (-\boldsymbol{G}^y\boldsymbol{J}_{uu}^{-1}\boldsymbol{J}_{ud} + \boldsymbol{G}_d^y) = (-\boldsymbol{G}^y\boldsymbol{G}_1^{-1}\boldsymbol{G}_{d_1} + \boldsymbol{G}_d^y)$.

In addition, we need to handle setpoint changes for the primary variable, $\boldsymbol{y}_{1,s}$, which requires changes in the setpoint for $\boldsymbol{c}$. Note that $\boldsymbol{y}_{1,s}$ only affects the cost function and may be viewed as a price variable p, so from (2.22) the required change in the setpoint is $\boldsymbol{\Delta c}_s = \boldsymbol{H}F_p\boldsymbol{\Delta y}_{1,s}$, where F_p may be obtained from the model (exercise: derive the expression!).

2.8.2 Indirect Control of $\boldsymbol{y}_1$ Based on Estimator

Note that we still have not used the available degrees of freedom in $\boldsymbol{H}$. To simplify the setpoint adjustment, we may use the degrees of freedom in $\boldsymbol{H}$ to make $\boldsymbol{H}F_p = I$, or equivalently, $\boldsymbol{c} = \boldsymbol{y}_1$. This means that $\boldsymbol{c}$ should be an estimate of $\boldsymbol{y}_1$. Note that $\boldsymbol{y}_1 = \boldsymbol{G}_1\boldsymbol{u}$ and $\boldsymbol{c} = \boldsymbol{H}\boldsymbol{G}_1\boldsymbol{u}$ (setting $d = 0$ for simplicity). These two gains need to be identical, so we use the extra degrees of freedom in $\boldsymbol{H}$ to make

$$\boldsymbol{H}\boldsymbol{G}^y = \boldsymbol{G}_1 \tag{2.34}$$

It is then easy to include changes in setpoint; we just control $\boldsymbol{c}$ at $\boldsymbol{y}_{1,s}$.

Some comments on this estimator

- What kind of estimator is this? If we look at the problem formulation, then we see that it is be the optimal estimator in the following sense: When we control $\boldsymbol{y}_1$ at the estimate (using the unconstrained degrees of freedom) then this minimizes the deviation from the given value (setpoint), for the expected range of disturbances and measurement noise.
- For practical purposes, when obtaining the model, it may be smart to let the primary outputs be the degrees of freedom, $\boldsymbol{u} = \boldsymbol{y}_1$ that is, to use

"closed-loop data" (this may seem strange, but it is OK). Then we have $\boldsymbol{G}_1 = I$.

2.9 Estimator for $\boldsymbol{y}_1$ Based on Data

The idea is to use the same approach as for the previous problem, but using data instead of a model. The objective is to estimate $\boldsymbol{y}_1$ based on measurements $\boldsymbol{y}$ (which we from now on will call $\boldsymbol{x}$ to follow statistics notation). That is, we want to find

$$\boldsymbol{y}_1 = \boldsymbol{H}\boldsymbol{x}$$

(where $\boldsymbol{x} = \boldsymbol{y}$). The available information is given by the data $\boldsymbol{Y}_{all} = [\boldsymbol{Y}_1; \boldsymbol{X}]$. Note that the data must first be centered.

To use our method, we first need to know the expected optimal variation Y. Here "optimal" means that $\boldsymbol{y}_1$ is constant. In addition, we also need to obtain $\boldsymbol{G}^y$ and $\boldsymbol{G}_1$ from the data. This means that the data must contain "non-optimal" variations in $\boldsymbol{u}$, and not only contain optimal data where $\boldsymbol{u} = \boldsymbol{u}_{opt}(\boldsymbol{d})$.

Comment: The setup is the same as for the previous problem, expect that it is not clear how noise in $\boldsymbol{y}_1$ can be included. It is a bit similar to "implementation error" which has been neglected since we assumed integral action.

2.9.1 Data Approach 1

Here we assume that $\boldsymbol{Y}_{all}$ is obtained from two different sources of data.

1. "Optimal" data with constant $\boldsymbol{y}_1(\boldsymbol{X} = \boldsymbol{Y}_{opt})$: This is closed-loop data for $\boldsymbol{y}$ with $\boldsymbol{y}_1$ constant for various disturbances ($\boldsymbol{d}$) and also with noise. It should be representative data for the expected operation. This directly gives the matrix $\boldsymbol{Y} = \boldsymbol{Y}_{opt}$ (including the weights) needed in (2.26).
2. "Non-optimal" data with constant $\boldsymbol{d}$: This is data for $\boldsymbol{x}$ and $\boldsymbol{y}_1$ collected with varying $\boldsymbol{u}$. From this data we directly obtain $\boldsymbol{G}^y$ and $\boldsymbol{G}_1$. By selecting $\boldsymbol{u} = \boldsymbol{y}_1$, one may also here used closed-loop data (but with $\boldsymbol{y}_1$ varying), in this case $\boldsymbol{G}_1 = I$.
3. Find optimal $\boldsymbol{H}$ using (2.26) with $\boldsymbol{H}\boldsymbol{G}^y = I$.

2.9.2 Data Approach 2: Loss Regression

More generally, we do not have separate "optimal" and "non-optimal" data. Instead, we have combined data $\boldsymbol{Y}_{all}$ where $\boldsymbol{y}_1$ and $\boldsymbol{x}$ vary simultaneously.

Note that we here use the notation from statistics/chemometrics and call the measurements $\boldsymbol{y}$ for $\boldsymbol{x}$.

We can then do a two-step procedure. In the first step, we "split up" the data $\boldsymbol{Y}_{all}$ data to find $\boldsymbol{G}_1$, $\boldsymbol{G}^y$ and $\boldsymbol{Y}_{opt}$, and in step 2 we proceed as "normal" to find the optimal $\boldsymbol{H}$.

Step 1A. We rearrange the data $\boldsymbol{Y}_{all}$ such that the $\boldsymbol{y}_1$ values are in the first rows, and the $\boldsymbol{x} = \boldsymbol{y}$-measurements are in the rest (called $\boldsymbol{X}$),

$$\boldsymbol{Y}_{all} = [\boldsymbol{Y}_1; \boldsymbol{X}]$$

Step 1B. We now want to separate the data into "optimal" and "nonoptimal" data. The data can generally be transformed by multiplying by a (real) unitary matrix $\boldsymbol{V}$, because $||[\boldsymbol{H}\boldsymbol{Y}_{all}]|| = ||[\boldsymbol{H}\boldsymbol{Y}_{all}\boldsymbol{V}]||$ for the 2-norm. Thus, we can use the SVD of

$$\boldsymbol{Y}_1 = \boldsymbol{U}_1\boldsymbol{S}_1\boldsymbol{V}_1^T$$

to transform the data to ("split up the data")

$$\boldsymbol{Y}_{all}\boldsymbol{V}_1 = \begin{bmatrix} \boldsymbol{G}_1 & 0 \\ \boldsymbol{G}^y & \boldsymbol{X}_{opt} \end{bmatrix}$$

Now, we have $\boldsymbol{G}_1$, $\boldsymbol{G}^y$ and $\boldsymbol{Y} = \boldsymbol{X}_{opt}$ and can proceed as normal using our derived formulas, see earlier.

Step 2. Find the optimal $\boldsymbol{H}$ by solving the convex optimization in (2.26) with $\boldsymbol{Y} = \boldsymbol{X}_{opt}$ and the constraint $\boldsymbol{H}\boldsymbol{G}^y = \boldsymbol{G}_1$.

```
% Loss method
%  step 1A
Yall = [Y1; X];
%  step 1B
[u,s,v]=svd(Y1);
Yall1 = Yall*v;
[r1,c1]=size(Yall);
[r2,c2]=size(Y1);
ny=r2;
G1=Yall1(1:ny,1:ny);
Gy=Yall1(ny+1:r1,1:ny);
Xopt = Yall1(ny+1:r1,ny+1:c1);

% step 2,
%Hopt = (pinv(Xopt*Xopt')*Gy)'; %analytical expression
[Hopt,loss]=soc_avg(Gy,Xopt);
D=Hopt*Gy*inv(G1);
Hoptloss=inv(D)*Hopt;
```

Comments

- Alternatively in step 2, provided $\boldsymbol{YY}^T$ has full rank, we may use the analytical expression $\boldsymbol{H}^T = (\boldsymbol{YY}^T)^{-1}\boldsymbol{G}^y$ in (2.25) and then "rescale" $\boldsymbol{H}$ to get a new $\boldsymbol{H}_1 = \boldsymbol{DH}$ which satisfies $\boldsymbol{H}_1\boldsymbol{G}^y = \boldsymbol{G}_1$, which gives $\boldsymbol{H}_1 = \boldsymbol{G}_1(\boldsymbol{HG}^y)^{-1}\boldsymbol{H}$. If $\boldsymbol{YY}^T$ does not have full tank one may use some pseudo inverse (similar to PCR). This adds degrees of freedom to the method. It has not been tested out but some preliminary results are promising.
- The method seems a bit similar to PLS in that we use the data for $\boldsymbol{y}_1$ to affect the x-data (we get $\boldsymbol{X}_{opt}$ from $\boldsymbol{X}$ by using the SVD of $\boldsymbol{Y}_1$, and also use $\boldsymbol{G}^y$ when minimizing $\boldsymbol{HX}_{opt}$.

2.9.2.1 Modification for Case with Too Few Experiments (e.g. Spectroscopic Data)

If we start with a model, then the data matrix $\boldsymbol{Y} = [\boldsymbol{FW}_d \;\; \boldsymbol{W}_{\boldsymbol{n}^y}]$ is a "fat" matrix; this is clear since the noise magnitude matrix $\boldsymbol{W}_{\boldsymbol{n}^y}$ is a square matrix (usually diagonal). Thus, there exists no matrix $\boldsymbol{H}$ such that $\boldsymbol{HY} = 0$.

However, if we start with data and have many measurements (e.g., spectroscopic data), then $\boldsymbol{Y} = \boldsymbol{X}_{opt}$ is likely a thin matrix, and there will exist an (or actually, infinitely many) $\boldsymbol{H}$ such that $\boldsymbol{HY} = 0$. Since the experimental data $\boldsymbol{Y}$ contains measurement noise, this means that $\boldsymbol{H}$ is "fitting" the noise. The proposed method will then be very similar to least squares, although the constraint $\boldsymbol{HG}^y = \boldsymbol{G}_1$ can make it different (as seen from the numerical examples below).

Extra step 1C. To fix this up, one may add "artificial" measurement noise to get a better representation of the actual measurement noise. Since there will always be some independent noise for each measurement, it is suggested to add a diagonal matrix $\boldsymbol{W}_{\boldsymbol{n}^y}$ to the original data

$$\boldsymbol{X}_{extended} = [\boldsymbol{X}_{opt}\boldsymbol{W}_{\boldsymbol{n}^y}]$$

where $\boldsymbol{X}_{opt}$ was found in Step 1B above.

The problem is now to choose $\boldsymbol{W}_{\boldsymbol{n}^y}$. One approach is to estimate it from the data using a preprocessing step. For example, one may do some preliminary regression and from this obtain an estimate of the noise.

A very simple approach, which is tested on some applications below, is to assume that the measurements have been scaled (for example, by the norm of the variation in each measurement), such that they have similar expected magnitudes. Thus, we use $\boldsymbol{W}_{\boldsymbol{n}^y} = w_n\boldsymbol{Y}$ where w_n is a scalar.

```
noise = wn*eye(rx);
Xoptnoise = [Xopt noise];
[Hopt,loss]=soc_avg(Gy,Xoptnoise);
% etc......
```

w_n is a tuning parameter but a systematic approach is the following: Plot the singular values of $\boldsymbol{X}_{opt}$ and select w_n such that the singular values of $\boldsymbol{X}_{extended}$ follow the same trend.

2.9.3 Modification: Smoothening of Data

The loss regression method is based on the loss method where it is assumed that a model is available. When we are using data, then the model has to be extracted first (step 1), and this step of the method is not based on a rigorous approach to the final objective, which is to use the model for future predictions.

Therefore, it is likely that this step may be improved, One possible modifications is suggested next, although it seems from the later numerical tests that it actually may have no effect.

2.9.3.1 Smoothening of $\boldsymbol{y_1}$

The above procedure (including step 1) assumes that all the noise is in the $\boldsymbol{x} = \boldsymbol{y}$. One way of dealing noise in y_1 is to run through the procedure twice.

First, we go through the procedure (steps 1 and 2) and find the optimal $\boldsymbol{H}_0$.

Next, we go through the procedure again, but with a modified step 1A where use $\boldsymbol{H}_0$ to estimate the smoothened (fitted) values of y_1 that correspond to the measured $\boldsymbol{X}$,

$$\boldsymbol{Y}_1^{\text{smooth}} = \boldsymbol{H}_0\boldsymbol{X}$$

and then we use this smoothened data in the other steps,

$$\boldsymbol{Y}_{all} = [\boldsymbol{Y}_1^{\text{smooth}}; \boldsymbol{X}]$$

```
Y1smooth = Hoptloss*X;
% then redo step 1 and 2
Yallsmooth = [Y1smooth; X];
[u,s,v]=svd(Y1smooth);
% etc....
```

However, from the numerical examples this smoothening has no effect on the results, which in some sense is good, but on the other hand it does not offer any extra degrees of freedom.

2.9.4 Numerical Tests

The Matlab files for these tests can be found at the home page of S. Skogestad, and the commands are also listed in the Appendix.

The objective is to find $\boldsymbol{H}$ (called beta in PLS Matlab) such that $\boldsymbol{y} = \boldsymbol{H}\boldsymbol{x}$. Note that PLS in all cases has an extra degree of freedom because it fits $\boldsymbol{y} = \boldsymbol{H}\boldsymbol{x} + \boldsymbol{h}_0$ where $\boldsymbol{h}_0$ is nonzero. This may help for the fitting, but not for validation. In any case, all the data is centered, so we can assume $\boldsymbol{h}_0$ is close to 0.

2.9.5 Test 1. Gluten Test Example from Harald Martens

Data: GlutenStarchNIR

1 $\boldsymbol{y}_1$ (gluten), 100 $\boldsymbol{x}$ (NIR absorbents), 100 data set).

This data has no noise on $\boldsymbol{y}_1$.

We first chose to use the 50 first data set for calibration and the 50 last for validation. Table 2.1 shows the fit $||\boldsymbol{Y}_1 - \boldsymbol{H}\boldsymbol{X}||_F$ to calibration and validation data. The best result for the traditional methods is PCR with 33 principal components (corresponding to a tolerance of 2e-4 when taking the pseudo inverse), which gives a validation fit of 0.3142. PLS can use much fewer components, but the fit is not as good (validation error is 0.3445) as PCR. As expected, the minimum loss method given perfect fit to the data and gives results identical to least squares. Thus, noise has to be added, and a noise level of 2.e-4 gives an error to the validation data (0.3115) which is even better than the best PCR

Table 2.1 Spectroscopic Example: Calibration and Validation data

Cal	Val	Method
0.0000	0.4758	Least squares (ls)
0.0738	0.3471	PCR-41 (tol=1.e-4)
0.1323	0.3142	PCR-33 (tol=2.e-4)
0.1890	0.4106	PCR-12 (tol=1.e-3)
0.0898	0.3608	PLS-8
0.0941	0.3445	PLS-7
0.1303	0.4987	PLS-6
0.0000	0.4735	min.loss
0.0000	0.4735	min.loss w/smooth y1
0.1442	0.3271	min.loss w/noise=5.e-4
0.1011	0.3115	min.loss w/noise=2.e-4

Use of smoothened y_1-data has no effect, which is expected as the fit to the calibration data is perfect.

Table 2.2 Spectroscopic Example: Calibration and Validation data (with sets interchanged)

Cal	Val	Method
0.0000	0.6763	Least squares (ls)
0.1687	0.3873	PCR-12 (tol=5.e-4)
0.1652	0.2357	PLS-7
0.0000	0.6785	min.loss
0.1476	0.3798	min.loss w/noise=5.e-4

Table 2.3 Spectroscopic Example: Calibration (99 measurements) and Validation data (1 rotating measurement)

Cal (avg.)	Val	Method
0.0000	3.0681	Least squares (ls) = min.loss
0.2689	0.3609	PCR (tol=5.e-4)
0.2769	0.3129	PLS-7
0.2471	0.3556	min.loss. w/noise=5.e-4

However, if we interchange the calibration and validation data set, then the results are very different; see Table 2.2. The best is now PLS-7 (val=0.2357), whereas minimum loss with noise is at 0.3798.

Finally, one data set was excluded at a time and used for validation. The average norm of the calibration fit and the norm of the validation fit for the 100 runs are given in Table 2.3. Again the PLS method is best (val=0.3129), whereas the minimum loss method with noise is the second best (val=0.2556).

2.9.6 Test 2. Wheat Test Example from Bjorn Alsberg (Kalivas, 1997)

Data: wheat spectra

2 $\boldsymbol{y}_1$, 701 $\boldsymbol{x}$, 100 data set.

The calibration data contains 50 measurements and the validation data set 50 measurements (specified in files I received).

The results are shown in Table 2.4. The loss method is the best (validation error 2.9 compared to 3.1 for the traditional methods), but the differences are small.

The difference between the methods is small for this test case and in this case the loss method gives better validation (2.9033) than least squares (3.1092) in spite of the fact that the calibration fit is perfect in both cases. This is a case where one would expect it to help to add artificial noise to the loss method. The reason is that we have 701 $\boldsymbol{x}$'s but only 50 data sets for calibration, so the data would not be expected to contain sufficient information about the expected noise. However, the numerical results do not confirm this, and the fit gets worse when we add noise.

$\boldsymbol{Y}_1$ contains both the two y-variables (water and protein); presumably better fit can be obtained by fitting one at a time.

Table showing the fit $||\boldsymbol{Y}_1 - \boldsymbol{H}\boldsymbol{X}||_F$ to calibration and validation data:

Table 2.4 Wheat Spectra Calibration and Validation data

Cal.	Val.	Method
0.0000	3.1092	least squares (= PCR -50)
0.3136	3.0871	PCR-47 (tol=1.1 e-3)
0.0052	3.1052	PLS-35
0.0000	2.9033	min.loss
0.0000	2.9033	min.loss + noise=1.e-8
0.0069	3.1099	min.loss + noise=1.e-4

Again, smoothening of y_1 has no effect, which again is not surprising since the fit was perfect.

2.9.7 Test 3. Our Own Example

The data contains 2 $\boldsymbol{y}_1$, 7 $\boldsymbol{x}$, 2 $\boldsymbol{d}$.

$\boldsymbol{G}_1 = \begin{bmatrix} 1\,0 \\ 0\,1 \end{bmatrix}$ $\boldsymbol{G}_{d_1} = \begin{bmatrix} 0\,0 \\ 0\,0 \end{bmatrix}$ $g_{y_1} = [0.2034563]$ $g_{y_2} = [00.213489]$ $\boldsymbol{G}_y = [g_{y_1}^T, g_{y_2}^T]$ $g_{d_1} = [004568 - 9]$ $g_{d_2} = [00 - 3 - 55918]$ $\boldsymbol{G}_{y_d} = [g_{d_1}^T; g_{d_2}^T]$

8 (or 32) noisy data set generated from ideal data $[\boldsymbol{G}^y \boldsymbol{G}_d^y]$ (4 data sets) with 2 (or 8) different noise sets to get a total of 8 data sets. We here generate the data by adding noise to data from a model ("calibration set") and the "validation set" is the noise-free data. To center the data I used opposite sign when the data was "repeated". The noise was generated using randn command in Matlab.

It was found that when tested on a single example then almost any of the methods could be the winner. To avoid this effect, the comparison was run many times (with different random noise).

Table 2.5 shows the average value of $||\boldsymbol{Y}_1 - \boldsymbol{H}\boldsymbol{X}||_F$ for 8 data sets after running 250 times. The best method is PCR (val=0.4931), PLS with 4 components (val=0.5137) is the best PLS. This is not surprising since we know that the data contains 4 directions. The new loss method is not doing so well in this case (val=1.055), which is not so surprising since with only 8 data sets there is limited information about the noise. Note that it is even worse than least squares (val=0.9985). As expected, the improvement by adding noise was significant (val=0.5850), but it is still not quite as good as PCR and PLS.

Surprisingly, smoothening of y_1 had absolutely no effect in this case, even when I added noise on the $\boldsymbol{y}_1$-data (results not shown).

Table 2.5 Our own example: 8 data sets in each run (average of 500 runs)

w/noise(cal)	no noise(val)	Method
0.2934	0.9985	LS (=PCR-8)
0.5496	0.4931	PCR (tol=1, most cases PCR-5)
0.5102	0.5137	PLS-4
0.3150	1.0552	loss method (no noise)
0.3150	1.0552	loss method (smooth y1)
0.4205	0.5850	loss method (noise=0.5)

Now, to show that the loss method does better when there is more data, Table 2.6 shows the average value of $||\boldsymbol{Y}_1 - \boldsymbol{HX}||_F$ for 32 data sets after running 300 times.

Table 2.6 Our own example: 32 data sets in each run (average of 300 runs)

w/noise(cal)	no noise(val)	Method
1.4826	0.3560	LS (=PCR-8)
1.4970	0.3560	PCR (most cases PCR-6)
1.5256	0.3698	PLS-4
1.7700	0.2690	loss method (no noise)
1.7700	0.2690	loss method (smooth y1)
1.7703	0.2687	loss method (noise=0.5)

The loss method is the winner (val = 0.269) in spite of the fact that it has no “tuning” parameters.

Here there is little effect of adding artificial noise with the loss method, presumable because we have enough data.

2.9.8 Comparison with Normal Least Squares

Normal least square solution. Problem: Find $\boldsymbol{H}$ such that the magnitude of $||\boldsymbol{Y}_1 - \boldsymbol{HY}||_2$ is minimized for the given set of data for $\boldsymbol{Y}_1$ and $\boldsymbol{Y}$.

Solution: $\boldsymbol{H} = \boldsymbol{Y}_1\text{pinv}(\boldsymbol{Y})$. This minimizes $||\boldsymbol{Y}_1 - \boldsymbol{HY}||_2$ and, for cases where this $\boldsymbol{H}$ is not unique, minimizes $||\boldsymbol{H}||_2$.

This is the same as finding $\boldsymbol{H}$ to minimize $||[I \quad \boldsymbol{H}]\boldsymbol{Y}_{all}||$, which we know is not the optimal solution

Proof. The data matrix is $\boldsymbol{Y}_{all} = [\boldsymbol{Y}_1; \boldsymbol{Y}]$ Assume that we seek $\boldsymbol{H}_{all}$ to minimize $||\boldsymbol{H}_{all}\boldsymbol{Y}_{all}||_2$. We have degrees of freedom in $\boldsymbol{H}_{all}$, so we set $\boldsymbol{H}_{all} = [I \quad -\boldsymbol{H}]$. Then we want to minimize $||\boldsymbol{H}_{all}[\boldsymbol{Y}_1; \boldsymbol{Y}]||_2 = ||-\boldsymbol{Y}_1 + \boldsymbol{HY}||$ the best solution is given by the pseudo inverse, $\boldsymbol{H} = \boldsymbol{Y}_1\text{pinv}(\boldsymbol{Y})$ which is the usual least square solution. □

So why is least squares not optimal?

The “problem” (objectives function) for normal least squares is to get the best match of the available data, ie., minimize $||\boldsymbol{Y}_1 - \boldsymbol{H}\boldsymbol{X}||$, and it does not consider how the estimate $\boldsymbol{y}_1 = \boldsymbol{H}\boldsymbol{x}$ is going to be used in the future.

So why is the loss approach expected to be better?

In the loss approach, the problem is:

Given that $\hat{\boldsymbol{y}}_1 = \boldsymbol{H}\boldsymbol{x}$, find an optimal $\boldsymbol{H}$ such that the average magnitude of $||\boldsymbol{y}_1 - \hat{\boldsymbol{y}}_1||_2$ is minimized for the expected future $\boldsymbol{d}s$ and $\boldsymbol{n}^y s$ (which are assumed 2-norm bounded).

Here, we use the data to obtain $\boldsymbol{Y} = \boldsymbol{Y}_{opt}$, $\boldsymbol{G}^y$ and $\boldsymbol{G}_1$ (step 1). This step may possibly be improved but it seems reasonable.

The main advantage is that in step 2, we obtain the estimate $\boldsymbol{y}_1 = \boldsymbol{H}\boldsymbol{y}$ that will work best “on average” for the expected disturbances and measurement noise (as are indirectly given by the data in $\boldsymbol{Y}_{opt}$, that is, we consider the future use of the estimator and not just fitting of the data).

2.10 Discussion

2.10.1 Gradient Information

How can we use the proposed approach in practice, for example, to find the optimal policy for a marathon runner. We need to be able to distinguish between “optimal data” ($\boldsymbol{Y}^{opt}$) and “nonoptimal data” to find $\boldsymbol{G}^y$ and $\boldsymbol{G}_1$. A simple approach is to set $\boldsymbol{y}_1 = \boldsymbol{J}_u$ (that is, we want to estimate the gradient). Then we know that optimal data corresponds to $\boldsymbol{y}_1 = \boldsymbol{J}_u = 0$ and we can do exactly as above (Data approach 1 or 2). However, note that this means that we need some non-optimal data where we know the value of $\boldsymbol{J}_u$.

2.10.2 Relationship to NCO tracking

Finally, some ideas related to NCO tracking.

The NCO-idea is to set the Gradient= 0. An important difference compared to the proposed loss approach, is that in NCO-tracking one tries to find an expression for $\boldsymbol{u}$ (“open-loop implementation” similar to deadbeat control).

On the other hand in the loss method (self-optimizing control), we “stop” when we have the expression for the gradient $\boldsymbol{c} = \boldsymbol{J}_u$. the implementation to find $\boldsymbol{u}$ that gives $\boldsymbol{J}_u = 0$ is by feedback control!

From (2.10) and (2.11) we see that we want to minimize $(\boldsymbol{u} - \boldsymbol{u}_{opt})$ or $\boldsymbol{J}_u$ (but weighted by $\boldsymbol{J}_{uu}$). Recall here that in the full-$\boldsymbol{H}$ case, $\boldsymbol{J}_{uu}$ is not needed. Still, it remains unclear if this means that we can just minimize $||\boldsymbol{J}_u||$?

Another problem with NCO idea to “Control gradient to zero” is that this is not really possible since gradient can not be measured. Thus, it needs to be estimated. For the case with no noise the estimate $\boldsymbol{J}_u$ is same as “nullspace method”, so $\boldsymbol{c} = \boldsymbol{H}\boldsymbol{y} = \boldsymbol{J}_u$!

For noisy case not so clear, but may as well use $\boldsymbol{c} = \boldsymbol{H}\boldsymbol{y}$.

2.11 Appendix

```

% This is file matlab-test-cases.m
% Load data from C:\Documents and Settings\skoge\My Documents\
    MATLAB

% -------------------------------------------------
% Test case 1. Martens data:
load GlutenStarchNIR
%    X: Variablene 13-112 er NIR absorbanser (log(1/T) fra
    850-1048 nm).
%    y1: Variabel 4 er kvantitiv analytt-konsentrasjon (gluten),
% Matrix: 100 rows 112 cols: regular MATLAB matrix
% VarLabels: 112 rows 17 cols: MATLAB character string
% ObjLabels: 100 rows 7 cols: MATLAB character string
Yall=Matrix';
XX = Yall(13:112,:);
YY = Yall(4,:);
X = XX(:,1:50);
X0= XX(:,51:100);
Y1= YY(:,1:50);
Y10=YY(:,51:100);

% repeated Martens
% 100 times where I take out data each time
%for nsis=1:98
%X = XX(:,[1:nsis nsis+2:100]); X0 = XX(:,nsis+1);
%Y1 = YY(:,[1:nsis nsis+2:100]); Y10 = YY(:,nsis+1);
% Two end cases are handled separately
%nsis=0
%X = XX(:,[nsis+2:100]); X0 = XX(:,nsis+1);
%Y1 = YY(:,[nsis+2:100]); Y10 = YY(:,nsis+1);
%nsis=99
%X = XX(:,[1:nsis]); X0 = XX(:,nsis+1);
%Y1 = YY(:,[1:nsis]); Y10 = YY(:,nsis+1)

```

```
% Jump to.. MATLAB code starts here

% ---------------------------------------------------
% Test case 2.
% Bjørn Alsberg data (epost 15.11 2010)
load wheat_spectra
%Your variables are:
%X       Y       idxcal  idxval  moist   protein
Xwheat=X'; Ywheat=Y'; % 2 y's, 701 x's, 100 data set
X =Xwheat(:,idxcal); Y1 =Ywheat(:,idxcal); % 50 calibration sets
X0=Xwheat(:,idxval); Y10=Ywheat(:,idxval); % 50 validation sets
%X =Xwheat(:,idxval); Y1 =Ywheat(:,idxval); % 50 calibration sets - switched
%X0=Xwheat(:,idxcal); Y10=Ywheat(:,idxcal); % 50 validation sets - switched


% Jump to.. MATLAB code starts here

% ---------------------------------------------------
% Test case 3. Own example

G1 = [1 0; 0 1]
Gd1= [0 0; 0 0]
gy1 =[0.2 0  3  4  5   6 3]
gy2 =[0 0.2  1  3  4   8 9]
Gy = [gy1',gy2']
gd1 =[0 0  4  5   6  8  -9]
gd2 =[0 0  -3  -5   5  9 18]
Gyd = [gd1',gd2']

Y10 = [G1 Gd1]
X0 = [Gy Gyd]

Y100=Y10;
na=0; aa=a*0;

% Run repeatedly from here for several cases

Y10=Y100;

% 8 data sets
Noise = 0.5*randn(7,8)
X =  [X0 -X0] + Noise
```

```
Y1 = [Y10 -Y10] % use Y10 and -Y10 to get centered data

%%32 data sets
%X =  [X0 -X0]
%Noise = 0.5*randn(7,32)
%X = [X X X X] + Noise
%Y1 = [Y10 -Y10] % use Y10 and -Y10 to get centered data
%Y1 = [Y1 Y1 Y1 Y1]
%%NoiseY= 0.0*randn(2,32) % with 2 y1's
%%Y1 = [Y1 Y1 Y1 Y1] + NoiseY
% 100 times where I take out data each time

%---------------------------------------------------
% MATLAB code starts here

% Least squares (= PCR with default tol)
method1='␣␣␣LS';
Hls = Y1*pinv(X);
res=Hls*X;
res0=Hls*X0;
a11=norm(res-Y1);
a12=norm(res0-Y10);

% PCR (vary tol)
% PCR: To find cut-off to find no. of components, semilogy(svd(X))
method6='␣␣␣PCR';
tol=1.e-4
Hpcr = Y1*pinv(X,tol);
npcr=rank(pinv(X,tol)) % no. of components used in PCR
res=Hpcr*X;
res0=Hpcr*X0;
a61=norm(res-Y1);
a62=norm(res0-Y10);

% Weighted least squares (to get relative noise assumption rather than additive noise)
method5='␣␣␣weightedLS';
[rx,cx]=size(X);
mag=[];
for i = 1:rx
mag=[mag norm(X(i,:))]; % the magnitudes are about 0.8
end
Xs = inv(diag(mag))*X;
Hlss1 = Y1*pinv(Xs);
```

```
Hlss = Hlss1*inv(diag(mag));
ress=Hlss*X;
ress0=Hlss*X0;
a51=norm(ress-Y1) ;
a52=norm(ress0-Y10) ;

% pls
npls=35
%npls=npcr
[XL,yl,XS,YS,beta,PCTVAR] = plsregress(X',Y1',npls);
% Note that PLS has an additional bias/centering parameter.
yfit = [ones(size(X',1),1) X']*beta;
yfit0 = [ones(size(X0',1),1) X0']*beta;
a71=norm(yfit-Y1');
a72=norm(yfit0-Y10');
method7='␣␣␣␣PLS';

% Loss method
%  step 1
Yall = [Y1; X];
[u,s,v]=svd(Y1);
Yall1 = Yall*v;
[r1,c1]=size(Yall);
[r2,c2]=size(Y1);
ny=r2;
G1=Yall1(1:ny,1:ny);
Gy=Yall1(ny+1:r1,1:ny);
Xopt = Yall1(ny+1:r1,ny+1:c1);

% step 2,
%Hopt = (pinv(Xopt*Xopt')*Gy)'; %analytical expression
[Hopt,loss]=soc_avg(Gy,Xopt);
D=Hopt*Gy*inv(G1); %
Hopt4=inv(D)*Hopt;
resb=Hopt4*X;
resb0=Hopt4*X0 ;
a41=norm(resb-Y1) ;
a42=norm(resb0-Y10) ;
method4='␣␣␣␣loss␣';

% NEW modified loss method for case with noise on Y: redo step
    1.
% New estimate of Y1.
Y1smooth = Hopt4*X;
% then redo step 1 and 2
```

```
Yallsmooth = [Y1smooth; X];
[u,s,v]=svd(Y1smooth);
Yallsmooth = Yall*v;
G1smooth=Yallsmooth(1:ny,1:ny);
Gysmooth=Yallsmooth(ny+1:r1,1:ny);
Xsmooth = Yallsmooth(ny+1:r1,ny+1:c1);
% step 2
[Hopt,loss]=soc_avg(Gysmooth,Xsmooth);
D=Hopt*Gysmooth*inv(G1smooth); %
Hopt10=inv(D)*Hopt;
resa=Hopt10*X;
resa0=Hopt10*X0 ;
a101=norm(resa-Y1) ;
a102=norm(resa0-Y10) ;
method10='␣␣␣loss␣w/smooth';

% loss method: add artificial noise weights
noisemag=2.e-4
%noisemag=tol
% Noise Alt.1 just additive noise:
noise = noisemag*eye(rx);
Xoptnoise = [Xopt noise];
% Noise Alt2. Add noise proportional to variation in each output
%[rx,cx]=size(Xopt);
%mag=[];
%for i = 1:rx
%mag=[mag norm(Xopt(i,:))]; % the magnitudes are about 0.8
%end
%noise = noisemag*diag(mag);
%Xoptnoise = [Xopt noise];

% step 2- with artificial noise
% semilogy(svd(X))
% semilogy(svd([X X0])
[Hopt,loss]=soc_avg(Gy,Xoptnoise);
D=Hopt*Gy*inv(G1); %
Hopt9=inv(D)*Hopt;
resb=Hopt9*X;
resb0=Hopt9*X0 ;
a91=norm(resb-Y1) ;
a92=norm(resb0-Y10) ;
method9='␣␣␣␣loss␣w/noise';

% Summary of results
```

```
methods=[method1 method5 method6 method7 method4 method10
    method9]
a =[a11 a12; a51 a52; a61 a62; a71 a72; ; a41 a42; a101 a102;
    a91 a92]'

% For repeated case 1 and case 3:
na=na+1
aa=aa+a;
aaa=aa/na
% For repeated case 1
end
```

Acknowledgements The authors are pleased to acknowledge the financial support from the NIL-I-007-d project and the Norwegian Research Council

References

Alstad V, Skogestad S (2007) Null space method for selecting optimal measurement combinations as controlled variables. Ind Eng Chem Res 46:846–853

Alstad V, Skogestad S, Hori E (2009) Optimal measurement combinations as controlled variables. Journal of Process Control 19(1):138–148

Halvorsen IJ, Skogestad S, Morud JC, Alstad V (2003) Optimal selection of controlled variables. Ind Eng Chem Res 42

Hori ES, Skogestad S, Alstad V (2005) Perfect steady-state indirect control. Industrial & Engineering Chemistry Research 44(4):863–867

Kariwala V, Cao Y (2009) Bidirectional branch and bound for controlled vaiable selection. part ii: Exact local method for self-optimizing control. Computers and Chemical Engineering 33:1402–1414

Kariwala V, Cao Y, Janardhanan S (2008) Local self-optimizing control with average loss minimization. Ind Eng Chem Res 47:1150–1158

Yelchuru R, Skogestad S (2010) Miqp formulation for optimal controlled variable selection in self optimizing control. In: PSE Asia, 2010. July 25-28, Singapore

Comments – Remarks

Comments – Remarks

Comments – Remarks

Comments – Remarks