Defragilization in Optimal Design and its Application to Fixed Structure LQ Controller Design

Torbjörn Norlander and Pertti M. Mäkilä, Senior Member, IEEE

Abstract—In this work, we present a general methodology to address the recently debated fragility issue in optimization-based modern control systems design. Based on this methodology we have developed a defragilization procedure which is applied to fixed structure linear quadratic (LQ) control design. Issues regarding computational efficiency of the procedure and the relationship between controller fragility and controller parameterization are discussed via two examples taken from practical applications. The first example is based on a rotary cement kiln control application and the latter example on a wing flutter control application. The procedure could also be used in design problems with specifications which are related to controller nonfragility requirements, e.g., so as to allow some degree of controller on-line tuning.

I. INTRODUCTION

RECENTLY there has been a lively debate about claimed fragility of controllers designed by modern control design methods [such as linear quadratic (LQ), $H_2$, $H_\infty$, or $L_1$ control]. This debate was initiated in [7] and several papers discussing the subject have followed, see, e.g., [2], [11], [17], and references therein.

In the present work fragility in optimization is discussed from a more general point of view, i.e., controller fragility is interpreted as an example of a general phenomenon in minimization of some, say, continuous function $f(x)$ for $x$ in some set $X$. Let $x^*$ denote a minimizer of $f(x)$ in $X$ and $\hat{x} \in X$ is some point close to $x^*$, yet $f(\hat{x})$ is considerably larger than $f(x^*)$, while there exists some point $x_p \in X$ such that $f(x)$ attains values only slightly larger than $f(x^*)$ for any point close to $x_p$. Then $x_p$ might well be preferred over $x^*$, say as an optimal design as one can never rule out inaccuracies during manufacturing, implementation, etc. of the optimal design. Hereafter we will call the optimization problem fragile when the above phenomenon occurs.

It is important to understand that controller fragility should not be regarded as an isolated problem but should instead be discussed together with the topic of controller parameterization (controller structure). It is important to choose a good structure for the controller as illustrated in [17]. Furthermore the choice of structure is not only important to consider in the design stage but also in the implementation step. See the survey by Hanselmann [5] for a thorough discussion on this subject, and the book by Moroney [14] where both design and implementation issues are discussed.

In this paper, we will describe how nonfragile optimal solutions can be accomplished by introducing a defragilization procedure. We will further discuss how this procedure can be applied on fixed structure LQ controller design. A brief discussion of controller parameterizations is included in the paper as this is closely related to fixed structure controller design. Interesting related recent work on nonfragile fixed structure LQ control design has been described in [3] and [6], where the term resilient controller design is used for nonfragile controller design. The procedure presented in this paper could also be applied with design methods which take into account other performance criteria, such as fixed structure $H_\infty$ design. This is, however, not considered in the present work.

The rest of this paper is organized as follows. In Section II, defragilization in optimal design is discussed. In Section III, defragilization is applied on controller design, specifically fixed structure LQ control design. An algorithm is given illustrating how to compute a defragilized controller using a modified descent Anderson–Moore method. Two examples are included in Section IV to illustrate the ideas of this work in practical applications. The first example is based on a digital control application for an industrial dry process rotary cement kiln and the second example on a wing flutter control problem.

II. DEFRACTILIZATION IN OPTIMAL DESIGN

An optimal design problem can be formulated as that of minimizing some real-valued criterion function $f(x)$ over $x$ in some set $X$. The set $X$ could be, e.g., the set of all $n$-tuples of real numbers. Let $x^* \in X$ denote a minimizer of $f(x)$ in $X$. Let $x^*$ be called an optimal design for the pair $(f,X)$.

However, the above standard optimal design problem and its solution leaves many questions open. One of these questions is related to how accurately the optimal design is actually realized, manufactured or implemented in the real world, e.g., if $x$ denotes the size and geometry variables for some engineering component, then clearly $x^*$ can be only realized within some finite tolerance. Hence if the optimal design $x^*$ is fragile in the sense that a small change in $x^*$ can result in a design criterion $(f)$ value that is much larger than $f(x^*)$, then one is left with the impression that $x^*$ is not really the true optimal design from an engineering point of view. That is, it is in general important to understand the behavior of the criterion function $f$ in more detail: it does not suffice to determine a minimizer of $f$!
Another open question has to do with computational issues. Let \( \mathcal{X} \) denote some nonempty open set in the Euclidean space of \( n \)-tuples of real numbers. What if \( f \) is not very smooth, say, continuous but not differentiable at a finite number of points in \( \mathcal{X} \). Then minimizing \( f \) in \( \mathcal{X} \) would require nondifferentiable optimization (NDO) theory. In this situation first-order sensitivity analysis would be difficult to do, a fact that restricts, e.g., the generality of pole-zero sensitivity minimization for addressing controller fragility [4]. Similarly, numerical NDO is typically much trickier than solving differentiable optimization problems. An interesting idea is that addressing fragility in the optimization problem directly could result in smooth optimization problems. Let us consider a simple motivating example.

**Example 2.1:** Define the function \( f(x) = ax \) for \( x \geq 0 \) and \( f(x) = bx \) for \( x < 0 \), where \( b < 0 \), \( a > 0 \) and \( -b \gg a \). That is \( f(x) \) is asymmetric around the origin so that it increases much steeper for negative \( x \) than for positive \( x \). Let \( \delta > 0 \) denote the maximum size of the uncertainty in implementing \( x \). Now \( x^* = 0 \) is the minimizer of \( f(x) \) on the real axis. Let \( f_\delta(x) \) denote the mean value of \( f \) in the interval \( I_\delta(x) = [x-\delta, x+\delta] \). It clearly makes sense to minimize \( f_\delta(x) \) rather than \( f(x) \) to find a defragilized optimal design for \( x \). This is also illustrated in Fig. 1. We can define the defragilization operator \( T_\delta \) as the mean value operator, that is \( T_\delta f = f_\delta \). The defragilized optimal design is then given as the minimizer of \( T_\delta f \). Note that \( T_\delta f \) is a differentiable function, although \( f \) is not differentiable at \( x = 0 \).

It is easy to see that the gradient of \( (T_\delta f)(x) \) has a unique zero at \( x = x(\delta) \equiv -\delta(a + b)/(a - b) > 0 \), which is the unique minimizer of \( T_\delta f \).

**Definition 2.1:** Define the defragilization operator as

\[
T_\delta f = f_\delta
\]

with the mean value function \( f_\delta \) of \( f \) defined as

\[
f_\delta(x) = \frac{\int_{V_\delta(x)} f \, dV}{\text{vol}(V_\delta(x))}
\]

of the minimizer \( x^* = 0 \) of \( f(x) \). Clearly there are other ways to choose the defragilization operator \( T_\delta \). The main point here is that this operator will depend on some parameters (here \( \delta \)), which can be used to select the defragilized optimal design in a sensible manner.

**A. A Defragilized Optimization Problem**

Let \( T_\delta \) denote a defragilization operator for the pair \((f, \mathcal{X})\). The defragilized optimal design problem is then as follows.

**Problem 2.1:** The Defragilized Optimal Design Problem: Minimize \( (T_\delta f)(x) \) in \( \mathcal{X} \).

Let \( \mathcal{X} \) be in the sequel the Euclidean space of all \( n \)-tuples of real numbers. (This choice is for illustration purposes only. Much more general situations can be considered in a similar manner.) We shall assume (for notational convenience) that the variables \( x \) are normalized so that each variable can be assumed to have equal uncertainty from the point of view of realization of the computed designs. Let \( \delta > 0 \) denote the maximum (additive) uncertainty in each of the components \( x_i \) of \( x \). (It would also be possible to consider relative, or multiplicative, uncertainties.) Let \( V_\delta(x) \) denote the \( n \)-dimensional cube centered at \( x \in \mathcal{X} \) with side length \( 2\delta \). So the volume, \( \text{vol}(V_\delta(x)) \), of \( V_\delta(x) \) is \((2\delta)^n\).

**Definition 2.1:** Define the defragilization operator as

\[
T_\delta f = f_\delta
\]
where the integral denotes a multiple Riemann integral over the cube $V_{\delta}(x)$.

We have the following result.

**Theorem 2.1:** Let $f$ be a continuous function in $X$. Then $T_{\delta}f$ is a continuously differentiable function in $X$. Furthermore, the gradient of $T_{\delta}f$ is given as

$$
\frac{\partial (T_{\delta}f)}{\partial x_i} = -\frac{1}{\text{vol}(V_{\delta}(x))} \int_{x_j - \delta \leq x_j \leq x_j + \delta} \left\{ \begin{array}{l}
y_1 \\
y_i - 1 \\
x_i + \delta \\
y_{i+1} \\
\vdots \\
y_n 
\end{array} \right\} \prod_{j=1, j \neq i}^{n} dy_j
$$

where $i = 1, \ldots, n$ and the integration is done over an $(n-1)$-dimensional cube.

In the one-dimensional (1-D) case the cube $V_{\delta}(x)$ reduces to the interval $I_{\delta}(x) = [x - \delta, x + \delta]$. The derivative of $(T_{\delta}f)(x)$ is then obtained from the definition of the derivative as

$$
d(T_{\delta}f) = \frac{f(x + \delta) - f(x - \delta)}{2\delta}
$$

The above derivative is a difference quotient of $f$ and is thus clearly a continuous function of $x$ for a continuous function $f$. The proof of Theorem 2.1 now follows from the 1-D case but due to lack of space the details are omitted here.

The defragilized optimal design problem invites a very interesting definition of a strong form of optimality.

**Definition 2.2:** A point $x_0 \in X$ such that $x_0$ minimizes $f$ in $X$ for any $\delta > 0$ is called a strongly optimal solution, or design, to the design problem: minimize $f$ in $X$.

Consider the 1-D case.

**Proposition 2.1:** Let $x_0$ be the set of all real numbers. Let $f$ be a continuous function on $x_0$. Let $f$ be symmetric with respect to the point $x_0$, that is $f(x_0 - x) = f(x_0 + x)$ for any $x$. Furthermore, let $f$ be strictly increasing for $x > x_0$. Then $x_0$ is a strongly optimal solution for the pair $(f, x_0)$.

**Proof:** First note that by the symmetry property, $f$ is strictly decreasing for $x < x_0$, that is $f(x_0 - y_0) > f(x_0 - y_1)$ for any $y_0 > y_1 \geq 0$. Note also that by (4), using symmetry and the strictly increasing property of $f$, $(T_{\delta}f)(x) > 0$ for $x < x_0$ and $(T_{\delta}f)(x) < 0$ for $x > x_0$, and $(T_{\delta}f)(x_0) = 0$. Hence $x_0$ is the unique minimizer of $T_{\delta}f$ for any $\delta > 0$. (Note that $x_0$ is also the unique minimizer of $f$.) This proves the result.

This agrees well with the intuition that in a symmetric optimization situation $(f, X)$, the minimizer, $x^*$, of $f$ is somehow stronger as there is no reason to prefer any other point $x \in X$ due to the lack of any asymmetry in the behavior of $f$ with respect to $x^*$. Hence in such a situation defragilization does not change anything.

Defragilized optimal design problems have thus many interesting properties. It is also possible to study the defragilization operator $T_{\delta}$ of Definition 2.1 as an operator, say on the linear normed space $C_b(X)$ of bounded real-valued continuous functions on the set $X$, of all $n$-tuples of real numbers. (This linear space has the norm $\|f\|_{\infty} = \sup_{x \in X} |f(x)|$.) The induced norm of $T_{\delta}$ is then defined as

$$
\|T_{\delta}\| = \sup_{f \neq 0, f \in C_b(X)} \frac{\|T_{\delta}f\|_{\infty}}{\|f\|_{\infty}}.
$$

The following result is useful.

**Proposition 2.2:**

**Proof:** It follows from the definition of the operator $T_{\delta}$ just before Theorem 2.1 that $\|T_{\delta}f\|_{\infty} \leq \|f\|_{\infty}$ for any $f \in C_b(X)$. Take $f$ to be any nonzero real constant. For such a function $\|T_{\delta}f\|_{\infty} = \|f\|_{\infty}$. This proves the result.

That is, $T_{\delta}$ has unit operator norm. This is another indication that the defragilization operator $T_{\delta}$ has nice properties and provides typically a regularization and smoothing effect.

It is somewhat surprising that standard treatises of optimization theory do not consider fragility issues, although they are clearly most interesting in optimal design theory.

Let us next derive a result that gives further insight into defragilized optimization problems.

**Theorem 2.2:** Let $f : \mathbb{R}^n \rightarrow \mathbb{R}$ be the quadratic function

$$
f(x) = c + b^T x + x^T Q x
$$

where

- $c$ real number;
- $b$ vector in $\mathbb{R}^n$;
- $Q$ an $n \times n$ positive definite (symmetric) matrix.

Consider the defragilized function $T_{\delta}f$, where $T_{\delta}$ is defined as earlier and $\delta > 0$ is a positive number. Then the unique minimizer, $x^*$, of $T_{\delta}f$ is given by

$$
x^* = -\frac{1}{2} Q^{-1} b
$$

and the minimum (value) of $T_{\delta}f$ by

$$
(T_{\delta}f)(x^*) = f(x^*) + \frac{\delta^2}{3} \text{tr} Q
$$

where $\text{tr}$ denotes the trace of a square matrix. That is, the point $x^*$ is a strongly optimal solution to the problem of minimizing $f$ in $\mathbb{R}^n$.

**Proof:** By direct integration one variable at time, it is seen that

$$(T_{\delta}f)(x) = f(x) + \frac{\delta^2}{3} \text{tr} Q.$$
of $T_δf$, as given above, follows then automatically. It is further observed that as the unique minimizer of $T_δf$ does not depend on the value of $δ > 0$, $x^*$ is also a strongly optimal solution to the problem of minimizing $f$. This completes the proof.

This result means also that for a quadratic function $f$ the gradient satisfies

$$\frac{\partial (T_δf)}{\partial x} = \frac{\partial f}{\partial x}$$

(11)

and the Hessian matrix (matrix of second-order partial derivatives) similarly

$$H(T_δf) = H(f).$$

(12)

Hence, if we were to minimize $T_δf$ iteratively, when $f$ is a quadratic function, by any method that uses only function values, the gradient, and the Hessian matrix, then this would lead to the same sequence of $x$-values as that of minimizing $f$ itself iteratively. This means that in the general case when $f$ is not quadratic, we cannot use the computationally simple idea of approximating $f$ locally by a quadratic function and to use the so obtained minimizer of the defragilized version of this quadratic approximation as the next iterate of $x$ and so on. This is so because such a sequence would at best converge so as to minimize $f$ but not $T_δf$ if the minimizers of $f$ and $T_δf$ are not the same.

The above result shows also that third- and higher order properties of a general smooth function have significance in defragilized optimization problems.

### III. Defragilization in Controller Design

Work on certain optimal controller realization problems with regard to finite word length (FWL) effects in digital control systems has been documented in [4], [9], [14], and [21]. Most of this work either considers first-order pole-zero sensitivity minimization or probabilistically modeled FWL effects in control signal computations and so this work does not provide a direct solution of the fragility issue raised in [7]. It seems that defragilization operator techniques as discussed in the present work provide a natural tool to address controller fragility.

It is very common to model systems from an input–output point of view using expanded transfer function models (in fact standard books on system identification deal mostly with such models). In the single-input single-output (SISO) case the expanded transfer function model $G(z)$ is given by

$$G(z) = \frac{P(z)}{Q(z)}$$

(13)

where $P(z) = \sum_{i=0}^{m} p_i z^i$ and $Q(z) = \sum_{j=0}^{n} q_j z^j$ are coprime polynomials (i.e., they have no common factors). Similarly controllers are often designed and expressed in expanded transfer function form.

It is, however, important to understand that the expanded transfer function form is problematic for medium- and high-order controllers (medium order meaning here 4–6 and high-order > 6). A dramatic way of illustrating this problem is provided by the following example.

**Example 3.1:** Consider the fragility of the roots of the polynomial $Q(z) = (z + a)^n$, where $0 < a < 1$, as a function of the order (degree) $n$. $Q(z)$ has a multiple root at $-a$. In expanded form the polynomial $Q(z) = z^n + na^2 z^{n-1} + \cdots + a^n$. Consider the perturbed polynomial $Q_δ(z) = Q(z) + \epsilon z^n$. Thus $Q_δ(z)$ is otherwise the same as $Q(z)$ except for a relative error of $\epsilon > 0$ in the zero order coefficient of $Q(z)$. The perturbed polynomial $Q_δ(z)$ has the roots $z = -a + \epsilon^{1/n} e^{\delta n (2k-1)/n}$, $k = 1, \ldots, n$. Take $n!$ to be an odd integer. The root with largest absolute value is then given by $-a(1 + \epsilon^{1/n})$. Hence the smallest relative error to make the perturbed polynomial unstable is determined from the condition $a(1 + \epsilon^{1/n}) = 1$. This gives

$$\epsilon = \left(\frac{1}{a} - 1\right)^n$$

(14)

that is for $1/2 < a < 1$ the required relative error tends to zero exponentially in $n!$.

Other similar examples (without repeated roots) can be readily constructed leading to the conclusion that the roots of extended form polynomials tend to become extremely sensitive or fragile to small changes or inaccuracies in the polynomial coefficients with increasing order of the polynomials. This gives a nice motivation for defragilized optimal fixed structure control problems. Note also that in general different realizations will perform differently in a defragilized optimal control problem, so this gives an interesting technique to choose a good realization.

Note that the dynamic control law

$$x_c(k + 1) = A_c x_c(k) + B_c y(k)$$

(15)

$$u(k) = C_c x_c(k) + D_c y(k)$$

(16)

corresponds just to one particular realization from a set of realizations that can be obtained from the above law by applying a similarity transformation $S$ on the state $x_c$. The above dynamic control law can then be written as, denoting $\xi_c = S x_c$:

$$\xi_c(k + 1) = S A_c S^{-1} \xi_c(k) + S B_c y(k)$$

(17)

$$u(k) = C_c S^{-1} \xi_c(k) + D_c y(k)$$

(18)

Now the expanded transfer function form corresponds to choosing $A_c$ in a companion matrix form. Often better realizations from the point of view of pole-zero fragility are obtained by transforming the expanded transfer function form so that $S A_c S^{-1}$ becomes block diagonal (small block sizes suffice), e.g., so that the transformed realization corresponds to a pole-zero factored transfer function form (that is to a cascade, or serial, connection of first– and second-order transfer function blocks).

This is further illustrated in example 4.2. On the importance of good controller realizations see also [1], [8] and [20].

#### A. Introduction to Fixed Structure LQ Control

Fixed structure LQ control is a design method where the order and the structure (or parameterization) of the controller are defined by the user. The method can be described as a tuning technique to optimally tune the parameters of the controller so that a discrete LQ criterion is minimized. A good starting point for the interested reader is [12]. In the present paper we will only give a brief introduction to fixed structure control.
Consider the discrete state-space system
\begin{align}
    x(k+1) &= Ax(k) + Bu(k) + v(k) \\
    y(k) &= Cx(k) + c(k) \\
    u(k) &= Fy(k)
\end{align}
(19) (20) (21)
where
\begin{align*}
    x &\text{ state; } \\
    u &\text{ input; } \\
    y &\text{ output; } \\
    v \text{ and } c &\text{ are zero mean white noise processes with covariance matrices } R_v \text{ and } R_c \text{, respectively, and with cross-covariance } R_{vc} = E[v(k)c(k)']. \text{(Here } E \text{ denotes expectation.)}
\end{align*}
The main problem in fixed structure LQ control design can be formulated as follows.

**Problem 3.1:** Find a discrete fixed structure controller gain matrix \( F \), such that the static output feedback control law (21) minimizes the discrete loss function
\begin{equation}
J(F) = E\left\{ x(k)'Q_1x(k) + 2x(k)'Q_2u(k) + u(k)'Q_2u(k) \right\}
\end{equation}
(22)
where \( \{x(k)\} \) and \( \{u(k)\} \) are to be interpreted as (second-order/weakly) stationary processes and \( S_F = \{ F \in \mathbb{R}^{p \times q} | \rho(A + BFC) < 1 \} \) and \( \rho(\cdot) \) denotes the spectral radius.

Alternatively, \( J(F) \) can be expressed as an infinite summation as follows:
\begin{equation}
J(F) = \lim_{N \to \infty} \frac{1}{N} E\left\{ \sum_{k=0}^{N-1} \left[ x(k)'Q_1x(k) + 2x(k)'Q_2u(k) + u(k)'Q_2u(k) \right] \right\}
\end{equation}
(23)
(This latter expression allows the stochastic process \( \{x(k)\} \) to evolve from an arbitrary initial condition for \( x(k) \) at \( k = 0 \).)

**Remark:** In a similar manner it is possible to formulate the sampled-data control problem, see [16].

A popular way to solve problem 3.1 is to introduce the loss increment technique, with the loss increment defined as
\begin{equation}
dJ = 2tr\{dF'[B'S(F + dF)B + Q_2dF(CP(F)C' + R_e)] + B'S(F + dF)(AP(F)C' + R_{ve}) + Q_{12}P(F)C' \} \\
+ tr\{dF'[B'S(F + dF)B + Q_2dF(CP(F)C' + R_e)] \}
\end{equation}
(26)
and the gradient is thus
\begin{equation}
\frac{\partial J}{\partial F} = 2SF'P + B'S(APC' + R_{ve}) + Q_{12}PC' \end{equation}
(27)
where \( S = B'SB + Q_2 \) and \( P = CPC' + R_e \). Here \( P(F) \) and \( S(F) \) are solutions to the algebraic Lyapunov equations
\begin{align}
P(F) &= (A + BFC)P(F)(A + BFC)' + BFR_eB' \\
&+ BFR_{ve} + R_{ve}(BF)' + R_v \\
S(F) &= (A + BFC)'S(F)(A + BFC) + Q_{12}FC' \\
&+ (Q_{12}FC)' + C'F'Q_2FC + Q_1.
\end{align}
(28) (29)
For a detailed description on how to determine the loss increment see [15].

By utilizing the structure of the loss increment in (24) it is possible to develop convergent methods to find the optimum (satisfying \( \partial J/\partial F = 0 \)) if there exists any, e.g., the descent Anderson–Moore method. The loss increment concept has been instrumental in rigorous proofs of convergence for many important algorithms for solving the necessary optimality conditions of the linear quadratic instantaneous output feedback problem, see [12] for a survey. It is also true that numerically efficient methods for solving the LQ instantaneous output feedback problem provide the most general and robust approach for solving the optimal state feedback matrix in linear quadratic control. This is especially important in multiple criteria LQ control as then the standard assumptions for LQ problems are somewhat restrictive, see [10] and [18].

### B. Defragilization in Fixed Structure LQ Control Design

Let us approximate the defragilization operator \( T_\delta \) by a simpler average by partitioning the cube \( V_\delta(x) \) into a finite number, \( N \), of equal sized parts and picking from each part a point \( x + \Delta_i, \ i = 1, \ldots, N \). That is by approximating the Riemann integral defining \( T_\delta \) by a certain Riemann sum. We can define this operation as a new defragilization operator \( R_\delta \)
\begin{equation}
(R_\delta f)(x) = \frac{1}{N} \sum_{i=1}^{N} f(x + \Delta_i).
\end{equation}
(30)
The gradient of \( R_\delta f \) can now be computed easily from that of \( f \) (so it is assumed now that \( f \) is a differentiable function) as
\begin{equation}
\frac{\partial R_\delta f}{\partial x} = \frac{1}{N} \sum_{i=1}^{N} \frac{\partial f(x + \Delta_i)}{\partial x}.
\end{equation}
(31)
That is to evaluate the gradient of \( R_\delta f \) at \( x \) we need to evaluate the gradient of \( f \) at \( N \) different points in the cube \( V_\delta(x) \). Obviously, the quantities \( \Delta_i, \ i = 1, \ldots, N \), can be chosen in many ways.
To apply the defragilization procedure to the fixed structure LQ control problem, define a defragilized fixed structure LQ control problem as follows.

**Problem 3.2:** Find a fixed structure controller \( F \in S_F \), such that the discrete static output feedback control law \( u(k) = Fy(k) \) minimizes a defragilized form of the loss function \( J \) in (22), defined as \( R_\delta J \), with respect to the controller feedback matrix \( F \), where the definition of \( R_\delta \) is adapted from (30) and \( S_F \) is defined in problem 3.1.

The loss increment expression suggests several methods for finding a minimizer of the defragilized loss function \( R_\delta J \). The simplest of these methods is a direct generalization of the descent Anderson–Moore method using a simple step length rule as follows. The algorithm is called here the defragilized descent Anderson–Moore (DDAM) method.

**Algorithm DDAM:**
1. Let \( F_0 \in S_F \) be an initializing feedback matrix. Let \( (R_\delta J)(F_0) \) be well defined. Choose \( \epsilon > 0 \) and an integer \( iterm > 0 \). Compute \( (R_\delta J)(F_0) \) and \( (\partial (R_\delta J) / \partial F)_0 \). Put \( k = 1 \).
2. If \( \| (\partial (R_\delta J) / \partial F)_k \| < \epsilon \) then stop with success flag up else go to step 3.
3. Set \( \alpha = 1 \). If \( k \geq iterm \) stop with nonconvergence flag up else continue to step 4.
4. Compute
   \[
   D_k = \frac{1}{2} S(F_k)^{-1} \left( \frac{\partial (R_\delta J)}{\partial F} \right)_k P(F_k)^{-1}.
   \] (32)
5. Set \( F_{k+1} = F_k + \alpha D_k \) and compute \( (R_\delta J)(F_{k+1}) \).
6. If \( F_{k+1} \) and \( F_{k+1} + \Delta i \in S_F, i = 1, \ldots, N \), and \( (R_\delta J)(F_{k+1}) < (R_\delta J)(F_k) \) then set \( k \leftarrow k+1 \) and go to step 2 else set \( \alpha \leftarrow 0.5 * \alpha \) and go to step 5.

End of Algorithm DDAM.

To prove global convergence of the DDAM algorithm to a stationary point of the loss function \( (R_\delta J)(F) \), it is in fact better to refine the step length rule in the DDAM somewhat, see [12] for details, but the above step length rule is usually satisfactory.

### IV. DeFRAGILIZATION EXAMPLES

Below we give two illustrative examples. The first example illustrates some computational aspects of defragilized fixed structure LQ control. The second example illustrates that different controller parameterizations behave differently against coefficient variations, even if the same design criterion has been minimized in the design. We have used MATLAB version 5.2 in our computations and the dlyap function of the Control Systems Toolbox has been employed to solve standard Lyapunov equations.

#### Example 4.1: This example is from [13] and [19], where an application on digital control of a dry process rotary cement kiln is described. The identified model was

\[
\begin{bmatrix}
0.914 & 0.0800 \\
-0.126 & 0.917
\end{bmatrix} y(k) +
\begin{bmatrix}
2.091 & -0.0744 \\
-0.211 & -0.0156
\end{bmatrix} u(k) +
\begin{bmatrix}
0 & 0 \\
0 & 0.715
\end{bmatrix} \epsilon(k)
\] (33)

where \( \epsilon \) is white noise with covariance matrix

\[
R_\epsilon = \begin{bmatrix} 0.0044 & 0.000257 \\ 0.000257 & 0.0214 \end{bmatrix}.
\] (34)

The control criterion is from [13]

\[
J = E[y(k)'Qy(k) + u(k)'Ru(k)]
\] (35)

where

\[
Q = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, R = \begin{bmatrix} 10.7 & 0 \\ 0 & 0.0220 \end{bmatrix}.
\] (36)

Let the control law be

\[
u(k) = F y(k).
\] (37)

By defining the state vector \( x(k) = [y(k)'c_2(k)'] \) we can write the system (33) as a standard state-space system with three states. Consider then the minimization of a defragilized loss function \( (R_\delta J)(F) \).

We computed the optimal feedback gain matrix \( F \) using the DDAM algorithm for

\[
(R_\delta J)(F) = \frac{1}{N} \sum_{i=1}^{N} J(F_i)
\] (38)

where \( \{F_i\}_{i=1}^{N} \) contains the center point \( F \) and the corner points \( F + \delta [\pm 1], F + \delta [\pm 1], \) of two hypercubes. (Here \( \pm 1 \) denotes any matrix of the same dimensions as \( F \) with elements \( \pm 1 \).) Let \( m = \dim u \times \dim y \). Hence \( N = 2^m+3 + 1 = 33 \). Put \( \delta = 0.03 \).

The algorithm was initialized with a zero matrix \( F_0 = 0 \), and \( \epsilon = 0.000001 \) as the stopping criterion for the 2-norm of the gradient of \( (R_\delta J)(F) \). This resulted in

\[
F_{op} = \begin{bmatrix} -0.2248 & 0.1068 \\ 2.0732 & 2.0321 \end{bmatrix}
\] (39)

with \( (R_\delta J)(F_{op}) = 0.3800 \). It took only 12 iteration steps (that is, 12 gradient evaluations of \( R_\delta J \) were required) and 1.25 \( R_\delta J \) evaluations per iteration step to reach the result!

Note that the LQG controller computed for the system (33) in [13] has 12 coefficients but only marginally better nominal performance than the best proportional controller with four parameters only. A four-parameter controller was implemented successfully [13], [19].

The next example illustrates that the controller structure should be chosen with care in the design in order to minimize
risks with controller fragility. It also demonstrates how fixed structure LQ control design can be utilized to compare, not only different orders of the controller, but also different choices of controller parameterizations still using the same design criterion.

**Example 4.2:** The system model is taken from a research project at Volvo Aero Corporation where wing flutter control is studied. The main issue is to study if active control of a wing’s trailing edge flap could be used to minimize flutter and in this way increase the flight envelope. We will use this system in the present work to discuss some practical properties with defrag-lized fixed structure LQ control.

The identified model of the system is

\[
x(k+1) = Ax(k) + Bu(k) + Gw(k)
\]

\[
y(k) = Cx(k) + v(k)
\]

with the matrices shown at the bottom of the page and \(w\) and \(v\) are uncorrelated white noise sequences with \(E\{w'w\} = 0.0198\) and \(E\{v'v\} = 8.2655 \cdot 10^{-4}\).

As an initial attempt we designed an LQG controller, using standard analytical methods found in MATLAB, for the system minimizing the loss function (22). The realization that were chosen for the LQG controller is given by (15) and (16) with \(D_c = 0\). After some tuning of the weight matrices we received good behavior with

\[
Q_1 = \begin{bmatrix}
500 & 0 & 0 & 0 & 0 \\
0 & 1000 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 1 \\
\end{bmatrix}, Q_{12} = 0, Q_{2} = 10.
\]

As the open-loop loss for the system is \(2.4687 \cdot 10^7\) and the achieved closed-loop loss using the LQG controller was 158.3274, a considerable reduction of the loss was achieved. However the designed LQG controller of order 5 has 35 parameters to implement which often is far too many for practical applications. If we instead define a state vector, output signal and control signal as \(\mathbf{x}(k) = [x(k), y(k)]\), \(\mathbf{y}(k) = [x_c(k), y_c(k)]\) and \(\mathbf{u} = [x_c(k+1), u(k)]\), where \(x_c(k)\) is the state vector of the controller. Then it is possible to use fixed structure LQ control design (e.g., a simpler form of DDAM with \(N = 1\)) to design optimal LQ controllers of any order for the system. The achieved optimal closed-loop loss values for controller orders from 1 to 5 are given in Table I. It is seen that a third-order controller has almost identical performance as the LQG controller designed with analytical methods, but with 20 parameters LESS! Furthermore, the first-order controller with three parameters has only slightly higher loss compared to the LQG controller.

A more attractive parameterization for practical controllers is the transfer function form (or input–output form) as these controllers often are more intuitive and easier to implement (less computations) than state-space controllers. We have therefore designed two second-order controllers in transfer function form, one in expanded transfer function form (\(K_{xtf}(z)\)) and the other in pole-zero factored transfer function form (\(K_{pzf}(z)\))

\[
K_{xtf}(z) = \frac{b_{11}z + b_{12}}{z^2 + a_{11}z + a_{12}}
\]

\[
K_{pzf}(z) = \frac{b_{21}(z + b_{22})}{(z + a_{21})(z + a_{22})}
\]

both controllers minimizing the same control criterion as the LQG controller. Note that \(K_{pzf}(z)\) only has real poles. We have for this case computed two optimal feedback gain matrices \(K_1\) and \(K_2\) with the DDAM algorithm for (22) using for \(K_1\) the augmented state- and output vector \(x_{K_1}(k) = [x(k), y(k), u(k-1), u(k-2), u(k-3)]\) and \(y_{K_1}(k) = [y(k), y(k-1), y(k-2), u(k-1), u(k-2)]\), with \(K_1 = [b_{11}, b_{12}, a_{11}, a_{12}]\). For \(K_2\) we have used the state-space representation defined above with \(D_c = 0\) and \(A_c\) in modal form, i.e., letting the coefficients \(A_c\{1,2\} = A_c\{2,1\} = 0\). and after the optimum is found transformed \(K_2\) into \(K_{pzf}(z)\). Parameters for the designed expanded transfer function controller (\(K_1\)) and the pole-zero factored transfer function (\(K_2\)), are given in Tables II and III.

<table>
<thead>
<tr>
<th>Order ((n_c))</th>
<th>(J_{opt})</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>169.128</td>
</tr>
<tr>
<td>2</td>
<td>168.674</td>
</tr>
<tr>
<td>3</td>
<td>158.887</td>
</tr>
<tr>
<td>4</td>
<td>158.3274</td>
</tr>
<tr>
<td>5</td>
<td>158.3274</td>
</tr>
</tbody>
</table>

The closed-loop performance is the same for both of these controllers (see also the second-order state-space controller), i.e., 

\[
J(K_1) = J(K_2) = 168.674\text{ (with a slight abuse of notation, as it should (if we expand K2 we get K1 as expected).}
\]
But let us now return to the issue of controller fragility. We examine the sensitivity of the closed loop with respect to adjustments in the controller, e.g., in the denominator parameters ($a_{11}$, $a_{12}$, $a_{21}$ and $a_{22}$). Then it is seen that for $K_2(z)$ we can allow any adjustments in $a_{21}$ and $a_{22}$ up to $+13\%$ (from their nominal values) without risking instability. This is not possible for $K_1(z)$ where the maximum deviation is only $+3\%$ in $a_{11}$ and $a_{12}$. Clearly the two controllers are not identical in the sense of controller fragility.

Consider now that we nevertheless choose to use the parameterization used in $K_{2tf}(z)$ and that we want to increase the tolerance against variations in $a_{11}$ and $a_{12}$, e.g., so that changes up to $+13\%$ can be allowed without risking instability (same as for $K_2(z)$). We have for this case computed an optimal feedback gain matrix $K_3$ with the DDAM algorithm for (38) using the augmented state- and output vector $x_{K1}$ and $y_{K1}$, with $K_{3d} = [b_{11}, b_{12}, a_{11} + \delta, a_{12} + \epsilon]$, where the min and max value of $\delta$ and $\epsilon$ are given as $+0.13 \cdot a_{11}$ and $+0.13 \cdot a_{12}$ respectively. We have used $N = 121$ so that we could get two equally spaced intervals (with five indexes each) around both $a_{11}$ and $a_{12}$. The parameters for this controller ($K_3$) are given in Table II and the optimal averaged loss is $J(K_3) = 109,716.5$. As the (nominal) loss is $J(K_2) = 109,192.4$, it is clearly only a small degradation in (nominal) performance that we have to allow when using defragilization in this case (compare with $J(K_1)$). We could also mention that ($\bar{R}_0, J)(K_2) = 187,358.6$, so a substantial reduction of averaged loss has been achieved.

To illustrate this further we have designed a controller ($K_4$) of the form $K_{2g}(z)$ with the same defragilization criterion as $K_3(z)$ using $A_{k2} = [1,1] = \alpha_{21} + \delta$ and $A_{k2} = [2,2] = \alpha_{22} + \epsilon$ (min and max value of $\delta$ and $\epsilon$ dependent on $a_{21}$ and $a_{22}$ for this case). We then got ($\bar{R}_0, J)(K_4) = 170,588.6$ and $J(K_4) = 109,357.5$, i.e., a large reduction in the averaged loss but only a minor increase in the nominal loss. Notable here is that in this case we have a pole-zero cancellation in the controller so the resulting controller is actually a first-order controller! Finally we can mention that the parameters $a_{11}$ and $a_{12}$ in $K_3(z)$ can be adjusted $+43\%$ from their nominal values without risking instability. Any of the parameters in the first-order controller $K_4(z)$ can be adjusted $+21\%$ before instability occurs in the closed-loop system.

V. Conclusion

We have presented a general methodology to address the recently debated fragility issue in optimization based modern control systems design. Based on this methodology we have derived a defragilization procedure for fixed structure LQ control design. This procedure has then been used in two realistic examples based on an industrial dry process rotary cement kiln control application and on a wing flutter control problem. In the examples the controller sensitivity could be reduced substantially without noticeable loss in performance. This procedure would also allow for some on-line controller tuning which is needed in many process industry applications (due to demands from the process operators).

<table>
<thead>
<tr>
<th>TABLE II</th>
<th>PARAMETER VALUES FOR CONTROLLERS $K_1$ AND $K_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a_{11}$</td>
<td>-1.3108</td>
</tr>
<tr>
<td>$a_{12}$</td>
<td>0.3948</td>
</tr>
<tr>
<td>$b_{11}$</td>
<td>9.8561</td>
</tr>
<tr>
<td>$b_{12}$</td>
<td>-6.2185</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>TABLE III</th>
<th>PARAMETER VALUES FOR CONTROLLERS $K_2$ AND $K_4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a_{21}$</td>
<td>-0.8419</td>
</tr>
<tr>
<td>$a_{22}$</td>
<td>-0.4689</td>
</tr>
<tr>
<td>$b_{21}$</td>
<td>9.8561</td>
</tr>
<tr>
<td>$b_{22}$</td>
<td>-0.6309</td>
</tr>
</tbody>
</table>

REFERENCES


**Torbjorn Norlander** was born in Motala, Sweden, in 1967. He received the M.Sc. and Ph.D. degrees in control engineering from Lulea University of Technology in 1995 and 2000, respectively. Since 1998, he has been employed at the Performance and Control Systems department at Volvo Aero Corporation. His current research interests include robust and optimal control, system modeling, and the application of advanced techniques on jet engine control and active wing flutter suppression.

**Pertti M. Mäkilä** (SM’83) was born in Turka, Finland, in 1954. He received the M.Sc. and Ph.D. degrees in chemical engineering from Abo Akademi University, Turku, Finland, in 1978 and 1983, respectively. He has been a Senior Fulbright Scholar at the University of California, Berkeley, a Gendron Fellow at the Pulp and Paper Research Institute of Canada in Montreal and Vancouver. He was Professor in control engineering at Lulea University of Technology, Lulea, Sweden, from 1994 to 1996. Since 1996, he has been Professor in automation engineering at Tampere University of Technology, Tampere, Finland. His current research interests include robust control, system modeling and identification, and signal processing.