OPTIMAL ACCOMMODATION OF FAULTS IN SENSORS AND ACTUATORS

Giacomo Tortora* 1 Basil Kouvaritakis*
David W. Clarke*

* Department of Engineering Science, University of Oxford, Oxford OX1 3PJ, UK

Abstract: A fault tolerant control scheme that accommodates sensor and actuator faults while maintaining optimal performance with respect to reference tracking is devised. Optimal fault-accommodation is achieved by solving a norm minimization problem which has been extensively studied in the literature. Here attention is focused on multiple-input single-output (MISO) systems. It is shown that, if the interest lies in the fault-tolerance of the output only, then the optimal tracking and fault-accommodation problems are totally decoupled. Motivated by the need to update in real-time the controller parameters as faults develop, an alternative scheme is presented which trades-off some optimality for computational simplicity. Copyright ©2002 IFAC

Keywords: Fault tolerance, intelligent instrumentation, sensor failures, actuators, multiobjective optimisation.

1. INTRODUCTION

Fault-tolerant control (FTC) is a relatively new field of research addressing the issue of designing controllers that are able to ensure the safe and efficient operation of the controlled system despite the occurrence of faults (Blanke et al., 2000).

Typically, fault-tolerance is achieved by implementing an automatic fault-detection and diagnosis scheme followed by a fault-accommodation or controller-reconfiguration algorithm. Tortora et al. (2002) and Tortora et al. (2001) present a different approach to fault-tolerance and apply it to the case of sensor-fault accommodation in SIMO systems where the control aim is to minimize the effect on a primary output of faults occurring on all available sensors while maintaining optimal performance in respect of reference tracking. This approach is based on the assumption that intelligent instrumentation (such as SEVA sensors) is deployed. These devices return to the user not only a working/faulty flag, but also a measure of the reliability of the instrumentation which can therefore be constantly and easily monitored. In Tortora et al. (2001) it is assumed that the sensor reading is corrupted at all times by additive noise, the size of the bounds on the noise giving a measure of the performance of the sensor. It is then possible to model sensor faults as an increase in the size of the bounds of the noise superimposed on the measurement signal, leading to a fault-accommodation algorithm that minimizes the effect on the controlled output of deterioration in sensor performance. The sensitivity of the controlled output to sensor faults is measured by the induced $l_{\infty}$ norm (i.e. the $l_1$ norm) of the transfer function from noise to output. Moreover, the class of controllers that give optimal tracking performance (measured by an $l_2$ norm) is defined so that restricting the choice of controllers to this class allows to accommodate faults without affecting the optimal tracking properties of the closed loop. In
Tortora et al. (2001) it is proved that the two optimizations of tracking and fault-accommodation performance are completely decoupled. This is a very desirable property, meaning that both optim can be obtained simultaneously.

The current work uses the same $l_1-l_2$ framework but extends it to consider the effect of both sensor and actuator faults on both the input and output variables. Similarly to sensor faults, actuator faults are modelled as an increase in an input disturbance. Again the disturbance is assumed to be additive and bounded. The systems under consideration are MISO, so that actuator redundancy is used to accommodate faults. The optimal tracking problem is set and solved in §2. The optimal fault-accommodation problem is solved in §3. It is furthermore established that, if we are only interested in the fault-tolerance of the output, the decoupling property of Tortora et al. (2001) can be extended to the case where fault-tolerance performance is measured with respect to both sensor and actuator faults. Moreover, the simultaneous optimization of reference tracking and output fault-tolerance still allows degrees of freedom in the controller. These can then be deployed to minimize input fault-tolerance. In §4 a computationally inexpensive and transparent (albeit suboptimal) alternative method of achieving fault-tolerance is presented. Numerical examples in §5 show that the suboptimality of this alternative method can be small.

Throughout this paper a discrete time system is assumed. The following convention will be used:

- no distinction between transfer functions and polynomials;
- lower case (l.c.) = scalar or scalar transfer function;
- l.c. underlined = vector of scalars;
- upper case (u.c.) = matrix;
- bold l.c. = vector of transfer functions;
- bold u.c. = matrix of transfer functions;
- $i,j$ denotes $i^{th}$ element of a vector;
- $i,j$ denotes $i,j^{th}$ element of a matrix;
- $i\rightarrow j$ = transfer function from $i$ to $j$;
- (c) denotes $t$-domain;
- $C_a$ denotes the Toeplitz convolution matrix of the polynomial $a(z^{-1})$. This a lower triangular matrix where the $i,j^{th}$ element is given by the coefficient of the $(i-j)^{th}$ power of $z^{-1}$ in the polynomial $a(z^{-1})$;
- $\|.|_2$ = induced $l_{\infty}$-norm for TFs or ratios of $z$-transforms and $l_1$-norms for $t$-domain.

2. THE TRACKING PROBLEM

Consider a MISO system with output $y$ and $l$ inputs denoted by the vector $u$. The system is subjected to measurement noise ($z$) and input disturbances ($d$) representing respectively sensor and actuator faults. The problem is to find a controller that minimizes the cost $J_{\text{track}}$ defined as:

$$J_{\text{track}} = \| \hat{r} - \hat{y} \|_2^2 + \sum_{i=1}^{l} \lambda_i \| \hat{u}_i \|_2^2,$$

assuming $x = 0$, $d = 0$,

where $\hat{u}_i$ denotes the $i^{th}$ input and $\| . \|_2$ denotes the $l_2$ norm. Let $b^T$, $a$ be the plant zero and pole polynomial respectively. Standard $l_2$ theory can be used to find the controller parameters $m, n$ that, deployed in the loop configuration of figure 1, minimize $J_{\text{track}}$. The closed-loop transfer function from reference to output is given by:

$$g_{r\rightarrow y} = \frac{b^T m}{am + b^T n}.$$  

Let $p$ denote the optimal closed loop pole polynomial:

$$p = am + b^T n, p(\infty) = a(\infty).$$

Considering eqn.2 it is apparent that the optimal $J_{\text{track}}$ can be achieved by any $m, n$ satisfying eqn.3. The general class of controllers satisfying eqn.3 can be obtained by simple transposition of the result in Tortora et al. (2002) which for brevity is given here without proof.

Lemma 2.1. (Tortora et al., 2002). The whole class of stabilising $m$ and $n$ that minimizes the tracking cost of (2) is given as:

$$\begin{bmatrix} m \\ n \end{bmatrix} = \begin{bmatrix} m_o \\ n_o \end{bmatrix} + \begin{bmatrix} e^T \\ E \end{bmatrix} q, \; q \in l_1$$

where:

$$am_o + b^T n_o = p, \; e = -b, \; E = -(aI + K_b E_1)$$

and $K_b$ is a matrix representation of the kernel of $b^T$, while $E_1$ is such that the zeros of $\det(aI + K_b E_1)$ are all within the unit circle.
3. OPTIMAL REFERENCE TRACKING WITH FAULT-ACCOMMODATION

3.1 Optimization of mixed output and input fault tolerance

As explained in the introduction, sensor and actuator faults are modelled as an increase in the bounds of (respectively) the measurement noise and the input disturbances. The reliability of the sensor and of the actuators can then be expressed by the scalar $s^x$ and the vector $d^x$ defined as:

$$ s^x = \|\hat{x}\|_\infty, \quad d^x = [\|\hat{d}_1\|_\infty \ldots \|\hat{d}_l\|_\infty]. $$

(6)

Fault-tolerance of the inputs and output is measured by the $\infty$-norms of $\hat{u}, \hat{y}$ for the worst case over the whole class of instrumentation faults (i.e. the whole class of $\hat{x}, \hat{d}$ satisfying eqn. 6). The indices of fault-tolerance $J_Y$ and $J_U$ of the output and inputs are then defined as:

$$ J_Y = \|\hat{y}\|_\infty, \quad J_U = \|\hat{u}\|_\infty \text{ (assumption } r = 0), \quad (7) $$

where superscript .* is used to denote the worst case over the whole class of faults. Assuming that the measures of sensor and actuator performance $s^x$ and $s^y$ are known, $J_Y$ and $J_U$ can be easily computed using the $l_1$ norms of the transfer functions from $x$ and $d$ to $y$ and $u$:

$$ J_Y = \sum_i \left( \|G_{i}^{d\rightarrow y}\|_{1} d_{i}^{d} + \|G_{i}^{x\rightarrow y}\|_{1} s^{x} \right), \quad (8) $$

$$ J_U = \max_j \left[ \sum_i \left( \|G_{i}^{d\rightarrow u}\|_{1} d_{i}^{d} + \|G_{i}^{x\rightarrow u}\|_{1} s^{x} \right) \right]. \quad (9) $$

For the loop configuration of Figure 1 these transfer functions are given by:

$$ g^{d\rightarrow y} = \frac{m b^T}{p}, \quad G^{d\rightarrow u} = -\frac{nb^T}{p}, \quad (10a) $$

$$ g^{x\rightarrow y} = \frac{b^T n}{p}, \quad g^{x\rightarrow u} = \frac{a m}{p}. \quad (10b) $$

Fault-tolerance can in general be measured by a "mixed fault-tolerance" index $J_{mft}$ defined as:

$$ J_{mft} = J_Y + \sigma J_U \quad (11) $$

where $\sigma$ is a constant. Mixed fault-tolerance can then be maximised subject to optimal tracking performance by minimizing $J_{mft}$ over the class of eqn.4 using eqns.8,9,10. In general this will define a "bad-rank" $l_1$ minimization problem (McDonald and Pearson, 1991) so that the optimal $q$ minimizing $J_{mft}$ will be a vector of transfer functions. This creates numerical problems since the minimization of $J_{mft}$ is given by an LP with an infinite number of variables and equality constraints. However, a variety of methods to get around this problem and obtain arbitrarily good approximations to the optimal have been devised. The most obvious approach is to optimize over a vector of polynomials $q$ and to truncate the transfer functions giving the equality constraints. As $n_q$ (the number of terms in each of the polynomial elements) tends to infinity then the solution to the LP will tend to the optimal $J_{mft}$. This is the Q-design method (Boyd and Barratt, 1991) and the resulting LP will not be discussed here. Recent research has concentrated on other methods of solving $l_1$ optimization problems which provide converging upper and lower bounds to the optimal solution (Khammash, 1997; Elia and Dahleh, 1998).

3.2 Optimization of output fault-tolerance only

An important special case arises when $J_{mft} = J_Y$. Optimal reference tracking with fault-accommodating then displays some important and desirable properties. The transfer functions affecting $J_Y$ are $g^{d\rightarrow y}$ and $g^{x\rightarrow y}$. Using eqn.3 we can rewrite:

$$ g^{x\rightarrow y} = -\left(1 - \frac{am}{p}\right), \quad (12) $$

so that the optimal fault accommodation problem is given by:

$$ \min_{m,n} J_Y $$

subject to: $am + b^T n = p, \quad m, n \in l_1, \quad (13b) $$

where:

$$ J_Y = \sum_i \left( \|\beta_i m\|_{1} \|\hat{d}\|_{1} \right) + \|1 - \frac{am}{p}\|_{1} s^x. \quad (14) $$

Note that $J_Y$ of eqn.14 is a function of $m$ only (a scalar transfer function) but we also have $n$ appearing in condition 13b. However, $n$ can be eliminated by noting that, given an $m \in l_1$, condition 13b is equivalent to requiring that the Bezout identity $b^T n = p - am$ has as a solution some $n \in l_1$. This is equivalent to requiring that $p - am$ has as zeros the non-minimum phase zeros of $b^T$ (where the zeros of the vector of polynomials $b^T$ are defined as the roots of the GCD of all its elements). Note that $b^T$ will have at least one non-minimum phase zero arising from a unit time delay. Assume that there exist $n_0$ such zeros and denote them by $\beta_1, \beta_2, \ldots, \beta_{n_0}$. Then the fault accommodation problem can be rewritten as:

$$ \min_{m} \left[ \sum_i \left( \|\beta_i m\|_{1} \|\hat{d}\|_{1} \right) + \|1 - \frac{am}{p}\|_{1} s^x \right] \quad (15a) $$

subject to: $m \in l_1, \quad m(\beta_i) = \frac{p(\beta_i)}{a(\beta_i)} \forall \beta_i. \quad (15b) $$

Let

$$ m = pm' \quad (16) $$

subject to: $m(\beta_i) = \frac{p(\beta_i)}{a(\beta_i)} \forall \beta_i. \quad (15b) $
where \( m' \) will in general be a transfer function. Then problem 15 becomes:

\[
\min_{m'} \left[ \sum_{i} \left( ||b_i m'||11 ||s'\right) + ||1 - am'||1s'\right] \tag{17a}
\]

subject to: \( m' \in l_1, m'(\beta_i) = \frac{1}{a(\beta_i)} \forall \beta_i \). \( \tag{17b} \)

Condition 17b, together with eqns.3 and 16 define an alternative controller parametrization for the minimization of \( J_y \) subject to optimal \( J_{\text{track}} \). Instead of minimizing over the vector of transfer functions \( q \) we minimize over the scalar transfer function \( m' \) subject to (17b) and recover \( m \) from eqn.16 and \( n \) as a solution to the Bezout identity \( b^T n = p - am \). The solution to the latter is non-unique but, as explained earlier, this has no bearing in the minimization of \( J_y \).

Moreover, note that \( p \) and \( h \) which minimize \( J_{\text{track}} \) (see eqns.2,3) do not appear in problem 17. This implies that it is possible to simultaneously optimize both \( J_{\text{track}} \) and \( J_y \). That is, the pair \( m_{\text{opt}}, n_{\text{opt}} \) minimizing \( J_y \) subject to the optimal \( J_{\text{track}} \) actually attain the minimum \( J_y \) over the whole class of stabilizing controllers. This fact is summarized in the theorem below.

**Theorem 3.1.** Let \( m_{\text{opt}} \) be the minimizer of problem 15 and \( n_{\text{opt}} \) be such that:

\[
n_{\text{opt}} \in l_1, \quad am_{\text{opt}} + b^T n_{\text{opt}} = p. \tag{18}
\]

Then the pair \( m_{\text{opt}}, n_{\text{opt}} \) not only minimizes \( J_{\text{track}} \) but also achieves the minimum \( J_y \) over the whole class of stabilizing controllers.

In general \( m_{\text{opt}} \) and \( n_{\text{opt}} \) will not be polynomial. However, an arbitrarily good approximation to the optimal can be obtained by taking \( m' \) to be a polynomial of arbitrarily high order. Let \( \gamma \) be a vector of length \( n_\gamma \), \( V_\gamma \) be the \( n_\gamma \times n_\beta \) Vandermonde matrix for the set \( \{\beta_i\} \), and

\[
\psi_\beta = \begin{bmatrix} \frac{1}{a(\beta_1)} \\ \vdots \\ \frac{1}{a(\beta_{n_\beta})} \end{bmatrix}.
\]

Moreover, let \( n_a \) be the number of elements of the polynomial \( a \), and \( C_{b1}, C_a \) denote the first \( n_\gamma \) columns and \( n_\gamma + n_a - 1 \) rows of the Toeplitz convolution matrix for \( b_1 \) and \( a \) respectively (see convention in §1). Then Problem 17 can be solved within any desired accuracy using the following LP:

\[
\mu_{\text{opt}} = \min_{\gamma, \mathbb{I}_+, \mathbb{I}_-} \begin{bmatrix} V_\beta^T & 0^T & 0^T \\ C_{b1} & \mathbb{I} & \mathbb{I}_- \\ \vdots & -I & I \\ C_{b\gamma} & \mathbb{I}_+ & \mathbb{I}_- \end{bmatrix} \begin{bmatrix} \gamma \\ \mathbb{I}_+ \\ \mathbb{I}_- \end{bmatrix} \leq 0, \quad \mathbb{I}_+, \mathbb{I}_- \geq 0, \tag{19a}
\]

where:

\[
s_\gamma^T = [s_0^T 1^T \ldots s_{n_\beta}^T 1^T s^T 1^T]. \tag{19c}
\]

As \( n_\gamma \to \infty, \mu_{\text{opt}} \) tends to the solution of Problem 17 and the minimizer \( \gamma_{\text{opt}} \) tends to the vector of the coefficients of \( m_{\text{opt}} \) such that \( m_{\text{opt}} = pm_{\text{opt}} \). The computational advantage of using the LP of (19) over the equivalent LP using truncated \( q \) is two-fold. First of all for a certain order of the closed loop polynomials (given by \( \mathbb{I}_+ - \mathbb{I}_- \)) there are fewer variables since we are effectively minimizing over the polynomial \( m' \) rather than the vector of polynomials \( q \). Moreover, the equality constraints defined by the Toeplitz matrices are finite and need not be truncated like they would be if we used truncated \( q \). The reduction in the number of equality constraints and the fact that they are exact makes the LP of (19) faster and more robust to numerical errors.

### 3.3 Minimizing \( J_y \) subject to optimal \( J_{\text{track}} \) and optimal \( J_y \)

The solution for the controller minimizing simultaneously \( J_y \) and \( J_{\text{track}} \) is not unique since given an \( m_{\text{opt}} \) there is a class of \( n_{\text{opt}} \) satisfying the Bezout identity of (18). This class is easily obtainable as:

\[
n_{\text{opt}} = n_p + K_q q_p, \quad am_{\text{opt}} + b^T n_{\text{opt}} = p, \quad n_p, q_p \in l_1 \tag{20}
\]

where \( K_q \) has already been defined in Lemma 2.1 as the kernel of \( b^T \). Moreover from eqns.9 and 10 it is clear that \( J_0 \) depends on \( p \). Having minimized \( J_y \) and \( J_{\text{track}} \) it would therefore be sensible to use the remaining degrees of freedom in the choice of \( n \) by minimizing \( J_0 \) over the class of (20). Again this leads to a "bad-rank" \( l_1 \) problem that can be solved by any of the methods quoted in §3.1. Using the \( Q \)-design method to minimize \( J_0 \) leads to an LP with truncated constraints. Numerical examples show that the minimum \( J_0 \) achievable is affected by the minimization of \( J_{\text{track}} \) and \( J_y \). Therefore performing the successive minimization of the three indices \( J_{\text{track}}, J_y, J_0 \) gives the global minimum for the first two but not for the third. On the other hand, minimizing \( J_y \) and then \( J_0 \) in succession is computationally more efficient than minimizing a linear combination of them.
4. FAULT ACCOMMODATION: REDUCING THE COMPUTATIONAL BURDEN

The fault-accommodation algorithm works by monitoring the indices of sensor and actuator performance $s^x$ and $d^d$. Each time these change the optimal controller parameters need to be recomputed. Therefore the LP’s optimizing fault-tolerance need to be performed on-line, as the faults develop. The computational cost of the fault-accommodation algorithm then becomes of paramount importance. As explained in §3 one way to reduce the computational burden is to minimize $J_Y$ and $J_Q$ in succession. Moreover, the optimization can be made simpler by reducing the number of terms in the controllers. However both these measures are likely to yield heavily suboptimal solutions.

An alternative is to constrain the polynomial vector $q$ to be given by a mix of $n_p$ stable predefined polynomial vectors $q_i$ such that:

$$q = \sum_{i=1}^{n_p} \rho_i q_i \quad q_i \in l_1, \quad (21)$$

and defining the controller $m$, $n$ from the parametrization of eqn.4. Fault-tolerance can then be achieved by optimizing $J_{m|n}$ over the coefficients $\rho_i$ rather than the polynomial vector $q$. For transparency and efficiency purposes it would be desirable to deal optimally with the complete failure of any actuator or of the sensor by setting one of the variables $\rho_i$ equal to 1 and all the others equal to zero. This can be done by choosing the polynomial vectors $q_i$ as the solutions to the optimal fault-accommodation problem (§3) for $s^x$ and for each one of the elements of $d^d$ going to infinity in turn. Different criteria for the choice of the set of $q_i$ can of course be used, so that optimality is recovered for a variety of fault scenarios. This is illustrated through an example in §5. Using more functions $q_i$ would obviously improve fault-tolerance as well as the number of variables of the on-line optimization, leading to a trade-off between optimality and computation time. Note that the $q_i$ can be computed off-line.

The foremost advantage of the “mixed” scheme proposed in this section compared to the optimal methods of §3 is that, for a given order of controller, we have fewer variables in the optimization and hence a faster algorithm. For the minimization over all stabilizing controllers to have a comparable computational cost we would need to keep the controller order much lower, leading to suboptimality for all operating conditions. On the other hand, the scheme proposed in this section can provide optimal controllers in the case of complete failure of the sensor or of one of the actuators, while being far more transparent to the operator than the schemes of §3.

5. NUMERICAL EXAMPLES

In this section the benefits of fault-accommodation will be illustrated using numerical examples. The plant under consideration has two inputs and one output. The open loop pole and zero polynomials are taken to be:

$$a = (1 - 1.5z^{-1})(1 - 0.99z^{-1})(1 + 1.1z^{-1}) \quad (22)$$

$$b = \begin{bmatrix} 1.5z^{-1}(1 - 1.3z^{-1})(1 + 0.5z^{-2}) \\ 2z^{-1}(1 - 1.2z^{-1})(1 + 0.6z^{-2}) \end{bmatrix} \quad (23)$$

For simplicity throughout this section we assume that $J_{m|n} = J_Y$.

![Fig. 2. Variation of $J_Y$, optimal fault-accommodation.](image)

Figure 2 shows the variation of $J_Y$ as fault-accommodation is performed using the optimal LP of (19). The values shown are for $s^d = 1$.

![Fig. 3. Faulty actuator 1. Comparing % suboptimality of sets 1 (continuous line) and 2 (dashed line).](image)

The plots of figure 3 show the suboptimality of the "mixed" strategy (§4) as the first actuator fails, namely as $g^d_1$ increases and $g^d_2$, $s^x$ are kept fixed at 1. In order to show the flexibility of this strategy the results obtained using two different sets of $q_i$ are displayed. The continuous line is given by using “set 1” and the dashed line by using “set 2”. These are defined in Tables 1 and 2 respectively.
Fig. 4. Faulty sensor. Comparing % suboptimality of sets 1 (continuous line) and 2 (dashed line).

<table>
<thead>
<tr>
<th>Table 1. Set 1</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
<tr>
<td>$q_1$ Faulty act. 1</td>
</tr>
<tr>
<td>$q_2$ Faulty act. 2</td>
</tr>
<tr>
<td>$q_3$ Faulty sensor</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 2. Set 2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
<tr>
<td>$q_1$ Faulty act. 1</td>
</tr>
<tr>
<td>$q_2$ Faulty act. 2</td>
</tr>
<tr>
<td>$q_3$ All working equally well</td>
</tr>
</tbody>
</table>

As expected both sets are optimal as $s_1^d \to \infty$ (of course the same property will pertain with $s_2^d$). Using set 1 will allow to recover the optimal controller in the case of complete failure of one of the actuators or of the sensor. Set 2 is unable to deal efficiently with sensor faults but on the other hand is also optimal when all instrumentation is working equally well. This is shown in figure 4 where the suboptimality of the two sets are compared for $s^e$ varying and $s_1^d = s_2^d = 1$ (sensor fault scenario).

6. CONCLUSIONS

This paper has treated the problem of accommodating sensor and actuator faults. The faults are modelled as an increase in the measurement noises and input disturbances respectively. The bounds on these are assumed to be known thanks to the use of intelligent instrumentation.

The control design objective is to accommodate optimally faults as they develop, in real-time while maintaining optimal performance with respect to reference tracking (measured by the $l_2$ cost index $J_{\text{track}}$). $l_1$ norms are used to define a suitable measure of fault-tolerance ($J_{\text{inf}}$) incorporating the effect that faults have on both the output ($J_Y$) and the inputs ($J_U$). It is shown that this problem can be solved using norm minimization methods which are extensively treated in the literature. Moreover, it is shown that the minimizations of $J_{\text{track}}$ and $J_Y$ alone have the property of being independent of each other, i.e. that there exist controllers that simultaneously achieve the optimal $J_Y$ and the optimal $J_{\text{track}}$. Also, this optimization can be performed using a controller parametrization alternative to the Youla, leading to a simpler algorithm. Furthermore, the optimal controller is not unique so that the remaining degrees of freedom can be used to minimize $J_U$.

The optimal controller parameters have to be computed on-line as faults develop prompting the need for a computationally inexpensive fault-accommodation algorithm. This motivates the alternative fault-accommodation scheme described in §4. This scheme reduces the on-line computational burden and still yields optimal reference tracking while optimally accommodating faults in given fault scenarios. The reduction in on-line computation is achieved at the cost of some suboptimality in fault-tolerance for other reliability levels of the instruments. This alternative scheme has the added advantage of great transparency to the user since it can deal with the failure of a single instrument by setting one coefficient equal to 1 and all the others equal to 0. Finally, the numerical examples of §§5 suggest that the degree of suboptimality can be rather small.

7. REFERENCES


