Zsolt HORVÁTH

Széchenyi István University Győr , Hungary

doi:10.15199/48.2019.04.02

Numerical methods for solving the Modified Filter Algebraic Riccati Equation for H-infinity filtering

Abstract: *This paper presents numerical methods for solving the Modified Filter Algebraic Riccati Equation (MFARE) for synthesis of H-infinity fault detection filters. Two methods are presented, namely the gamma-iteration and then rewriting the MFARE in Linear Matrix Inequalities (LMIs) and casting it as a convex optimization problem. Each algorithm has to ensure the condition for a global convergence and also has to deliver an optimal solution. Not at least the computational cost has to be as small as possible.*

Streszczenie*. Zaprezentowano metodę numeryczną rozwiązywania równania MFARE (Modified Filter Algebraic Riccati Equation). Badano dwie metody – iterację gamma i przepisywanie równania w postać Linear Matrix Inequantities.* ***Metody numeryczne rozwiązywania równania MFARE do filtrów typu H-infinity***

Keywords: Modified Filter Algebraic Riccati Equation, linear-quadratic optimization problem, H-infinity optimization, gamma-iteration, LMI.
Słowa kluczowe: MFARE - Modified Filter Algebraic Riccati Equation, optymalizacja H-infinity

Introduction

Diesel engines have become more complex and powerful in the past decade, moreover lots of the mechanical functions are being replaced by electric and electronic devices, which are controlled by the ECU. In order to ensure the strict environment policies, these devices and the ECU as well have to make sure the reducing of fuel consumption and the emission of pollutant species. On the top of this the ECU is also equipped with reliable fault diagnose system to detect possible actuator, sensor and component failures in the engine. The subject of our investigation is a robust model-based fault detection filtering of faults in the air-path of diesel engines. When designing a H-infinity filter, the filter gain can be obtained by solution of a Modified Filter Algebraic Riccati Equation (MFARE), which is one of the central and most difficult tasks in the synthesis, see e.g. in [1], [2], [3] and [4]. One way to get there is an applying gamma-iteration, another one, which is more state of the art, is using LMIs. Several investigations of robust control have been carried out in the past two decades using LMIs, see e.g. [5], [6], [7]. As a result, it has been stated, that LMIs are effective and powerful tools for handling complex, but standard problems, such as a fast computing of global optimum within some pre-specified accuracy. As even it is to be done in our case, solving the H-infinity optimization problem to specify the filter.

This paper is organized as follows: after the introduction, in Section II we shortly revisit the problem of H-infinity optimization and describe the MFARE. In Section III MFARE is converted to an LMI as an optimization problem. In Section IV an algorithm called gamma-iteration is implemented to solve the MFARE, then it is formulated as a linear objective minimization problem using LMI.

Deriving the Modified Filter Algebraic Riccati Equation for robust H-infinity detection filtering

<u>The optimal H-infinity detection filtering problem</u>

According to [8], the linear time-invariant system (LTI-system) subjected to worst-case effect of modelling uncertainties, external disturbances and the effect of unknown faults, can be represented in state space form as follows:

$$\dot{x}(t) = Ax(t) + Bu(t) + B_\kappa \kappa(t) + \sum_{i=1}^{k} L_i \nu_i(t), \quad y(t) = Cx(t). \tag{1}$$

In (1) $x \in \mathbb{R}^n$, $y \in \mathbb{R}^p$, $u \in \mathbb{R}^m$. *A, B, and C* are appropriate constant matrices. It is assumed, that *(A, C)* is an observable pair. $B_\kappa = [B_w, L_\Delta]$ is the worst-case input direction and $\kappa(t) \in L_2[0,T]$ is the input function for all $t \in \mathbb{R}_+$ representing the worst–case effects of modelling uncertainties and external disturbances. It is to note, that the equation does not include parametric uncertainty [8]. The cumulative effect of a number of *k* faults appearing in known directions L_i of the state space and is modelled by an additive linear term, $\sum L_i \nu_i(t)$. $L_i \in \mathbb{R}^{nxs}$ and $\nu_i(t)$ are the fault signatures and failure modes respectively. $\nu_i(t)$ are arbitrary unknown time functions for $t \geq t_{ji}$, $0 \leq t \leq T$, where t_{ji} is the time instant when the *i*-th fault appears and $\nu_i = 0$, if $t < t_{ji}$. If $\nu_i(t) = 0$, for every *i*, then the plant is assumed to be fault free. Assume, however, that only one fault appears in the system at a time [8].

Based on the LTI-system model in (1), the state estimate can be obtained as

$$\dot{\hat{x}}(t) = A\hat{x}(t) + K(C(x(t) - \hat{x}(t))) + Bu(t), \quad \hat{y}(t) = C\hat{x}(t), \quad \hat{z}(t) = C_z \hat{x}(t), \tag{2}$$

where $\hat{x} \in \mathbb{R}^n$ represents the observer state, $\hat{y} \in \mathbb{R}^p$ represents the output estimate, $z \in \mathbb{R}^p$ denotes the output signal and $\hat{z} \in \mathbb{R}^p$ represents the weighted output estimate. *K* is the observer gain matrix and C_z is the constant estimation weighting.

The filter error system for (2) can be formulated as

$$\dot{\tilde{x}}(t) = (A - KC)\tilde{x}(t) + B_\kappa \kappa(t) + \sum_{i=1}^{k} L_i \nu_i(t), \quad \varepsilon(t) = C_z \tilde{x}(t). \tag{3}$$

In (3), $\tilde{x}$(t) and $\mathcal{E}$(t) are the state error and weighted output error, respectively, defined as

$$\tilde{x}(t) = x(t) - \hat{x}(t), \quad \varepsilon(t) = z(t) - \hat{z}(t) \tag{4}$$

Note that in presence of faults, the estimation error does not converge asymptotically to zero, but converges asymptotically to a subspace which is different from zero [8].

In the following we have to choose the filter gain, by minimizing the magnitude of the effects of perturbations on the output of the filter, which has to maximize the magnitude of the transfer function from failure modes to the filter error.

Solution to a H-infinity filtering

The performance measure for H-infinity filtering considered as a quadratic cost function is defined as

$$J(w,\nu,\hat{z})=\frac{1}{2}\left[\|z-\hat{z}\|_2^2-\gamma^2\left(\|w\|_2^2+\|\nu\|_2^2\right)\right] \quad (5)$$

where $\omega \in \mathbb{R}^p$ denotes the process disturbance in $L_2\,[0,T]$ and $\gamma > 0$ is a positive rational constant.

The quadratic cost function has to be minimized as

$$\sup_{w,\nu,a_i} J(w,\nu,\hat{z}) \quad (6)$$

The performance can be formulated as a min-max problem, which is a trade-off between the worst-case disturbance $\kappa(t)$and the L_2 norm of the filter error $\mathcal{E}(t) = z-\hat{z}$ on L_2. That is, minimizing the H-infinity norm of the transfer function, denoted by $H_{\varepsilon\kappa}$, of the worst-case disturbance to the filter output. The worst-case performance measure is given by

$$J(\mathrm{K},\kappa)=\sup\frac{\|z-\hat{z}\ \|_2}{\|\kappa\ \|_2}=\|H_{\varepsilon\kappa}(s)\|_\infty \quad (7)$$

The filter gain K can be obtained by solving a linear-quadratic optimization problem, using the procedure presented below (see also in [8]).

With substitution of the decision variable $Y \in R^{nxn}$, which is a positive definite matrix, the observer equation can be written as

$$\begin{aligned}\dot{\hat{\mathrm{x}}}(t)&=(A-YC^TC)\,\hat{\mathrm{x}}(\mathrm{t})+Bu(t)+YC^Ty(t),\\ \hat{\mathrm{z}}(\mathrm{t})&=\mathrm{C}_z\,\hat{x}(t).\end{aligned} \quad (8)$$

From the bounded-real lemma, we have $\|H_{\varepsilon\kappa}\|_\infty < \gamma$ if and only if there exists $Y \geq 0$ such that the MFARE is

$$AY+YA^T-Y(C^TC-\frac{1}{\gamma^2}C_z^TC_z)Y+B_KB_K^T=0 \quad (9)$$

The algorithm, which is used to find an optimal solution for Y, iteratively reduces the γ until Y has no longer a positive definite solution. Note that the γ_{min} obtained this way is within a given arbitrarily small tolerance $\varepsilon > 0$.

After solving equation (9) and getting the solution for Y, the filter gain can be obtained as

$$K = Y\,\mathrm{C}^T \quad (10)$$

Using the $\boldsymbol{\gamma_{min}}$ the detection threshold of the filter can be given as

$$\tau(C_z)=\gamma_{\min}\|\kappa\|_2 \quad (11)$$

Note that the failure modes, which have the magnitude smaller than that of the detection threshold, cannot be detected by the filter.

Solving the MFARE with LMI

The reason for dealing with LMIs is that, that a lot of problems subjected to the control theory can be cast as convex optimization problem. What is more, most of them can be converted to a standard LMI problem such as a computing of global optimum with some pre-specified accuracy, as even if it is to be done in our case, the solving the H-infinity optimization problem. The main benefit of the LMI formulation is that it defines a convex constraint with respect to the variable vector. For that reason, it has a convex feasible set which can be found guaranteed by convex optimization. A detailed survey about the LMIs can also be found in the mathematical literature, see e.g. in [9], [10] and also in textbooks for control engineering e.g. in [11], [12], [13] and [14].

Standard problems involving LMIs

A linear matrix inequality is a matrix inequality of the form

$$F(x)\overset{\Delta}{=}F_0+\sum_{i=1}^{m}x_iF_i>0 \quad (12)$$

where $x \in \boldsymbol{R}^m$ is the vector of decision variables, and $F_i = F_i^T \in \boldsymbol{R}^{n\,x\,n},\ i = 0,\cdots,m$ are symmetric matrices .

Let $A(x)$, $B(x)$ and $C(x)$ be symmetric matrices that depend affinely on $x \in \mathbb{R}^m$. Then the standard LMI problems can be formulated in three different ways (see e.g. in [13]):

1. Feasibility problem with the task of finding a solution for decision variable x so that the constraint sufficient.

$$A(x) < 0 \quad (13)$$

2. Linear objective minimization i.e. searching for x which minimizes the linear function subject to an LMI.

That is

$$\text{minimize } c^T x \quad \text{subject to } A(x) < 0. \quad (14)$$

3. Generalized eigenvalue minimization problem i.e. minimizing the maximum generalized eigenvalue of a pair of matrices, that depend affinely on a variable, subject to an LMI constraint.

The task is

minimize λ

subject to an LMI constraint:

$$\begin{aligned}A(x) &< \lambda B(x)\\ B(x) &> 0\\ C(x) &< 0\,.\end{aligned} \quad (15)$$

Rewriting the MFARE in LMI

Unfortunately most of the control synthesis problems are not formulated as an LMI, but the nonlinear (convex) inequalities can be converted to an LMI form using the Schur complements lemma (Boyd et. al. in 1994) see in [13].

According to this lemma the expressions (16) and (17) are equivalent.

$$\begin{bmatrix}Q(x) & S(x)^T\\ S(x) & R(x)\end{bmatrix}<0, \quad (16)$$

$$R(x)<0,\quad Q(x)-S(x)^TR(x)^{-1}S(x)<0 \quad (17)$$

$Q(x) = Q(x)^T,\ R(x) < 0$, and $S(x)$ depend affinely on x.

On this way the set of nonlinear inequalities in (17) can be represented as the LMI in (16).

Back to our problem, we have to solve the MFARE as

$$AY+YA^T-Y(C^TC-\frac{1}{\gamma^2}C_z^TC_z)Y+B_KB_K^T=0 \quad (18)$$

To transform (18) into an LMI, at first we rewrite it in form of inequalities. For this let $R = Y^{-1}$, so we get

$$A^T R + RA - C^T C + \frac{1}{\gamma^2} C_z^T C_z + \tag{19}$$

$$+ RB_K B_K^T R < 0 \; , \; R > 0.$$

Applying the Schur complement lemma (17) for (19) yields to

$$\underbrace{A^T R + RA - C^T C}_{Q(x)} - \underbrace{\begin{bmatrix} C_z^T & RB_\kappa \end{bmatrix}}_{S^T(x)}$$

$$\underbrace{\begin{bmatrix} -\gamma^2 I & 0 \\ 0 & -I \end{bmatrix}^{-1}}_{R^{-1}(x)} \underbrace{\begin{bmatrix} C_z \\ B_\kappa^T R \end{bmatrix}}_{S(x)} < 0. \tag{20}$$

Finally by using the Schur complement lemma in (16) we obtain the LMI for the MFARE as

$$\begin{bmatrix} RA + A^T R - C^T C & C_z^T & RB_\kappa \\ C_z & -\gamma^2 I & 0 \\ B_\kappa^T R & 0 & -I \end{bmatrix} < 0, \tag{21}$$

which has a solution $R = R^T \in \mathbb{R}^{n\, x\, m}$ for $\gamma > 0$.

Consequently we can solve the MFARE by minimizing γ with respect to $R \succ 0$ subject to (21), that is

$$\begin{cases} \min \gamma \\ s.t. \; R > 0 \\ \begin{bmatrix} RA + A^T R - C^T C & C_z^T & RB_\kappa \\ C_z & -\gamma^2 I & 0 \\ B_\kappa^T R & 0 & -I \end{bmatrix} < 0. \end{cases} \tag{22}$$

<u>Computing the optimal solution using the Hamiltonian–matrix</u>

In most cases it is possible to solve the Algebraic Riccati Equation also through similarity transformation of the Hamiltonian-matrix, see e.g. in [15]. Although this method is not for solving the MFARE as an optimization problem, so it won't lead to an expected result, it may be useful to evaluate the result obtained via the optimization for fixed γ. According to this idea we seek the optimal solution of the MFARE for fixed $\gamma > 0$.

The Hamiltonian-matrix corresponding to the MFARE can be derived as:

$$AY + YA^T - Y(C^T C - \frac{1}{\gamma^2} C_z^T C_z)Y +$$

$$+ B_K B_K^T = 0,$$

$$\begin{bmatrix} P & I \end{bmatrix} \begin{bmatrix} A^T & -(C^T C - \frac{1}{\gamma^2} C_z^T C_z) \\ -B_\kappa B_\kappa^T & -A \end{bmatrix} \begin{bmatrix} I \\ -P \end{bmatrix}, \tag{23}$$

where the term is placed in the middle is the Hamiltonian-matrix:

$$H_\gamma = \begin{bmatrix} A^T & -(C^T C - \frac{1}{\gamma^2} C_z^T C_z) \\ -B_\kappa B_\kappa^T & -A \end{bmatrix}, \tag{24}$$

which has no eigenvalue on the imaginary axis.

The method is described as follows [15]. Build the $(2n, n)$ matrix V, which contains the eigenvectors associated to the eigenvalues with negative real parts (stable invariant subspace) of the Hamiltonian-matrix as

$$V = \begin{bmatrix} V_1 \\ V_2 \end{bmatrix}, \tag{25}$$

thus it is valid that

$$H_\gamma \begin{bmatrix} V_1 \\ V_2 \end{bmatrix} = \begin{bmatrix} V_1 \\ V_2 \end{bmatrix} diag\, \lambda_i \,. \tag{26}$$

By multiplying (26) with V_1^{-1} from the right side, the equation can be formulated as

$$H_\gamma \begin{bmatrix} I \\ V_2 V_1^{-1} \end{bmatrix} = \begin{bmatrix} V_1 (diag\, \lambda_i) V_1^{-1} \\ V_2 (diag\, \lambda_i) V_1^{-1} \end{bmatrix}. \tag{27}$$

Then multiplying (27) from the left side with $[V_2 V_1^{-1} \; I]$, so obtained

$$\begin{bmatrix} V_2 V_1^{-1} & I \end{bmatrix} H_\gamma \begin{bmatrix} I \\ -V_2 V_1^{-1} \end{bmatrix} = 0. \tag{28}$$

By comparing (27) to (23) and then recognizing the similarity, the matrix Y can be calculated as

$$Y_H = V_2 V_1^{-1}. \tag{29}$$

It can be stated, that the solution for Y can easily be obtained from the eigenvectors V_1, V_2 of the Hamiltonian-matrix of the MFARE. The method can be used in evaluating the optimal solution obtained by solving the LMI, for instance.

Calculation of the filter gain based on the LTI -model of the air path of the diesel engines

By the investigation of fault detection filtering we are interested in the efficiency and robustness of the optimal solution for the filter gain. To this aim, two different methods for solving MFARE are compared. First the algorithm gamma-iteration is implemented, then the MFARE is formulated as a LMI and solved it as a linear objective minimization problem.

<u>LTI-model for the air path of diesel engines</u>

As mentioned in the introduction, the robust fault detection filter design, that we apply in our investigation requires the using of the LTI-model. Here we refer to a simplified nonlinear model of the air path which was first suggested by Jankovic and Kolmanovsky in 1998 [16] and later by Jung [17] for the purpose of robust control of the diesel engines. In our earlier investigation [18] we have already linearized this model around a specified operating point (Herceg, 2006) [19]. For the sake of simplification, we have considered a fuelling of diesel oil as a constant input and not as disturbance, furthermore the disturbance was modelled as a fluctuating change of the engine speed.

As a result we derive the following LTI-model at the chosen operating point [18] as

(30)
$$A=\begin{bmatrix}-5.2643 & 4.7316 & 28.5021\\ 50.7697 & -156.9827 & 0\\ 0 & 0.4287 & -9.0909\end{bmatrix},$$
$$B=\begin{bmatrix}1.6111\cdot10^{9} & 0 & 0\\ -1.5720\cdot10^{10} & 8.3514\cdot10^{4} & 1.46083\cdot10^{8}\\ 0 & -141.6484 & 0\end{bmatrix},$$
$$B_{\omega}=\begin{bmatrix}-47.7946\\ 466.3408\\ 0\end{bmatrix},\ C=\begin{bmatrix}1 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & 3.924\cdot10^{-5}\end{bmatrix},$$

where A, B, C and B_{ω} are appropriate constant matrices, B_{ω} is the matrix for the effect of worst-case disturbance acting on the system.

Solution of the MFARE by a gamma-iteration algorithm

This section discusses a conventional numerical method called gamma-iteration to get an optimal solution of MFARE. It has to be noted, that this method is often referred to, see e.g. in [1], [2] and [20], [21], but we have not found any algorithm about it. This has been the motivation for its description.

For the start of the explanation, the estimation weight of the filter is chosen arbitrarily according to the methodology described in [8] as

$$C_Z=\begin{bmatrix}5 & 0 & 0\\ 0 & 5 & 0\\ 0 & 0 & 25\end{bmatrix}. \quad (31)$$

The MFARE is written again as

$$AY+YA^T-Y(C^TC-\frac{1}{\gamma^2}C_z^TC_z)Y+B_KB_K^T=0 \quad (32)$$

Arranged it for the use of the MATLAB function ***care*** [22], (32) becomes:

$$AY+YA^T-Y\begin{bmatrix}C_z^T & C^T\end{bmatrix}\underbrace{\begin{bmatrix}-\gamma^2 I & 0\\ 0 & I\end{bmatrix}}_{R_{care}}^{-1}\begin{bmatrix}C_z\\ C\end{bmatrix}Y+B_KB_K^T=0. \quad (33)$$

Note that the function ***care*** is basically used for solving the H-infinity Riccati Equation for purpose of control synthesis. However, according to the principle of duality between controller and observer the ***care*** function can be parameterized to be used it for a filter synthesis in the form:
[Y L Gr report] = care (A', CC, B_K B_K', R_{care}, 'report')*,

where $CC=[C_z^T \quad C^T]$.

The function ***care*** returns the optimal value for the decision variable, denoted by *Y.*

Of course the R_{care} - matrix contains γ, but this has a constant value for a specified level for that. It results that the function ***care*** cannot be directly used for a quadratic minimization problem, that is, the value of γ is to be iteratively reduced and the decision variable minimized. In this manner in order to get the γ_{min} value, and so the corresponding optimal solution for *Y,* we implemented an algorithm called gamma-iteration in which an interval halving method is used iteratively. The algorithm reduces the value of γ until *Y* has no longer positive definite solution. The γ_{min}, which is reached, is within the limits given by an arbitrarily small tolerance $\varepsilon>0$.

The gamma-iteration algorithm can be formulated as follows:

- The inputs for the method are: the *A*, B_w, *C*, C_z matrices, which define the LTI-system, *eps* as the relative accuracy of the solution, *maxgamma* as the right limit of the interval (the left limit is zero).
- The second variables: *a*, *b* and *i*. They stand for assignation of interval and counting cycle, respectively. The *gamma* as step size (midpoint), the *minigamma* variable, which contains the value of gamma at the end of an iteration. The *Lambda* vector contains the eingenvalues of the solution.
- The outputs are: the matrix *Y* as a positive definite decision variable, the *mingamma, which* contains the *gamma* value when the iteration is finished.

Each iteration performs the following steps:

1. Calculate gamma, the midpoint of the interval, which is assigned by *a* and *b*. That is *gamma = a+(b-a)/2*;
2. Call the MATLAB function *care* which returns the matrix *Y* and the *"report"*;
3. Calculate the eigenvalues of *Y,* called *Lambda*;
4. If the convergence criteria of the iteration are not satisfied, namely:
   a. *Y* is NOT positive definite that is *prod(Lambda)<=0* *or*
   b. The associated R_{care} – matrix in (33) contains gamma had eigenvalues on or very near the imaginary axis;

   Then the upper and lower bounds of interval are changed;
   Otherwise the value of *gamma* is saved, that is *minigamma = gamma* and the iteration is continued;
5. Examine whether the new interval assigned by *b-a* reached the relative accuracy of the solution - called *epsilon*:

   If not, the iteration is repeated;

If yes, the iteration is finished and the filter gain is calculated based on the previous value of *minigamma,* that is *mingamma*.

The algorithm is implemented in MATLAB the example is based on the LTI-system in (30) and its script is shown in [23].

The input matrices were:

$$A=\begin{bmatrix}-5.2643 & 4.7316 & 28.5021\\ 50.7697 & -156.9827 & 0\\ 0 & 0.4287 & -9.0909\end{bmatrix}, \quad (34)$$
$$B=\begin{bmatrix}1.6111*10^{9} & 0 & 0\\ -1.5720*10^{10} & 8.3514*10^{4} & 1.4608*10^{9}\\ 0 & -141.6784 & 0\end{bmatrix},$$

$$C=\begin{bmatrix}1 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & 3.924*10^{-5}\end{bmatrix}, C_z=\begin{bmatrix}5 & 0 & 0\\ 0 & 5 & 0\\ 0 & 0 & 25\end{bmatrix}.$$

$$B_\omega=\begin{bmatrix}-47.7946 & 0 & 0\\ 466.3408 & 0 & 0\\ 0 & 0 & 0\end{bmatrix},$$

Further conditions were: *maxgamma = 1100* and *eps = 0,01.*

By performing the γ-iteration repeated it 21-times the optimal value of $\gamma_{min} = 4.9698$ is obtained. With (10) from Y the corresponding filter gain given as

$$K=\begin{bmatrix}257.2236 & -39.2216 & -0.0000\\ -39.2216 & 699.2298 & 0.0000\\ -0.7934 & 1.6744 & 0.0000\end{bmatrix}. \tag{35}$$

Note that in the steps 8,11,13 and 20 we did not get solution because the ***care*** returned with a report = -1, which means that the associated R_{care} - matrix had its eigenvalues on or very near the imaginary axis, which resulted in failure, see in [22]. According to the interval halving algorithm, in these steps the upper and lower bounds of the interval are changed in order to keep the eigenvalues away from the imaginary axis.

In order to prove the filter performance for disturbance attenuation, the transfer function of the disturbance to a filter residual for the obtained filter gain K is

$$H_{\varepsilon\omega}(s)=C_Z(sI-A+KC)^{-1}B_\omega. \tag{36}$$

The evolution of the disturbance attenuation during the iteration steps can be observed on the value of $\|H_{\varepsilon\omega}(s)\|_\infty$, which was calculated in MATLAB and plotted in Fig. 2.

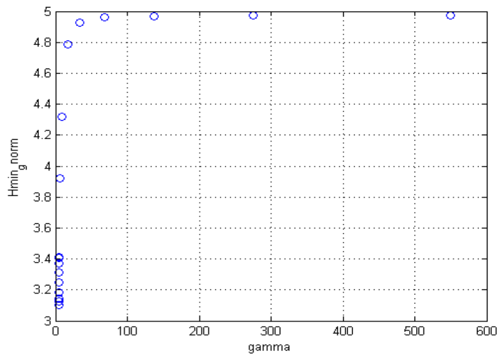

Fig.2. Changing $\|H_{\varepsilon\omega}(s)\|_\infty$ value as a function of gamma values during the iteration

The optimal value obtained at the end of the iteration is for $\|H_{\varepsilon\omega}(s)\|_\infty = 3.3737$.

<u>Evaluating the optimal solution of MFARE using the Hamiltonian-matrix</u>

It is possible to verify the solution for the decision variable by calculating the eigenvectors of the Hamiltonian-matrix of MFARE as it was explained in Subsection 3.3.

The resulting Hamiltonian-matrix for MFARE in case of $\gamma_{min} = 4.9698$ is

$$H_\gamma=10^5\begin{bmatrix}-0.0001 & 0.0005 & 0.0000 & 0.0000 & 0.0000 & 0.0000\\ 0.0000 & -0.0016 & 0.0000 & 0.0000 & 0.0000 & 0.0000\\ 0.0003 & 0.0000 & -0.0001 & 0.0000 & 0.0000 & 0.0003\\ -0.0228 & 0.2229 & 0.0000 & 0.0001 & -0.0000 & -0.0003\\ 0.2229 & -2.1747 & 0.0000 & -0.0005 & 0.0016 & 0.0000\\ 0.0000 & 0.0000 & 0.0000 & 0.0000 & -0.0000 & 0.0001\end{bmatrix}. \tag{37}$$

At first, we calculate the eigenvalues and the corresponding eigenvectors of the Hamiltonian-matrix via a similarity transformation. The resulting matrix, containing the eigenvalues obtained as

$$diag\,\lambda_i(H_\gamma)=diag\,[150.0393, -150.0393, 6.8273, 6.8273, 0.4622, -0.4622, -6.8273]. \tag{38}$$

Secondly, we have to build a $(2n,n)$ matrix for the V, which contains the eigenvectors of the Hamiltonian-matrix corresponding to the eigenvalues with negative real parts (25).

The submatrices of V, which contain the eigenvectors, are:

$$V_1=\begin{bmatrix}0.0005 & 0.0039 & 0.0029\\ -0.0014 & 0.0001 & -0.0003\\ 0.0004 & 0.0062 & -0.1194\end{bmatrix}, \quad V_2=\begin{bmatrix}0.1749 & 0.9986 & 0.8475\\ -0.9846 & -0.0516 & -0.5171\\ -0.0027 & -0.0023 & -0.0139\end{bmatrix}. \tag{39}$$

Let Y_H denote a solution calculated using Hamiltonian-matrix, which has a solution

$$Y_H=V_2V_1^{-1}=\begin{bmatrix}258.0838 & -32.9691 & -0.7468\\ -33.7848 & 691.7261 & 1.7722\\ -0.7809 & 1.6763 & 0.0932\end{bmatrix} \tag{40}$$

From the gamma-iteration in Subsection 4.2 we got an optimal solution as

$$Y=\begin{bmatrix}257.2236 & -39.2216 & -0.7934\\ -39.2216 & 699.2298 & 1.6744\\ -0.7934 & 1.6744 & 0.0934\end{bmatrix}. \tag{41}$$

The matrices of the absolute –and relative faults between the two solutions are

$$Y_H-Y=\begin{bmatrix}0.8602 & 6.2525 & 0.0466\\ 5.4368 & -7.5037 & 0.0978\\ 0.0125 & 0.0019 & -0.0002\end{bmatrix}, \quad (Y_H-Y)Y^{-1}=\begin{bmatrix}0.0058 & 0.0083 & 0.3995\\ 0.0237 & -0.0129 & 1.4808\\ 0.0000 & 0.0000 & -0.0019\end{bmatrix}. \tag{42}$$

It can be stated, that the matrices Y_H and Y are slightly different. This leads to the conclusion of plausibility of an optimal solution Y obtained by the gamma- iteration.

Getting the solution for the MFARE using LMI

In Section 3 we introduced the method for finding the optimal solution for MFARE implemented analytically as an interval halving algorithm. However, the task of the minimization results in the task of computing a system of matrix equations which is not always convex [8].

For this reason, let us now consider the problem of finding the optimal solution for the filter gain by solving of MFARE formulated as a LMI.

To handle it, several commercial software tools can be chosen. In this study the LMI Control Toolbox of MATLAB has been used, which provides a set of convenient functions to solve problems involving LMIs [24].

Generally, the solution of LMIs is carried out in two stages in MATLAB. At first, the decision variables of the LMI are defined, then it is defined the system of LMIs based on these decision variables. These are mostly represented in matrix form. In the second stage, the optimization problem is solved numerically using the chosen solvers as it is explained in Section 3.1.

In our case study the LMI in (21) is formulated as a linear objective minimization problem. That is, the task is to minimize a linear function of x subject to an LMI constraint as

(43) $$\min_x \left\{ c^T x : F(x) \succ 0 \right\}.$$

That is

(44) $$\begin{cases} \min \gamma \\ s.t.\ R > 0 \\ \begin{bmatrix} RA + A^T R - C^T C & C_z^T & RB_\kappa \\ C_z & -\gamma^2 I & 0 \\ B_\kappa^T R & 0 & -I \end{bmatrix} < 0. \end{cases}$$

The Matlab script for the linear objective minimization problem in (44) is shown in [23].

Comparing the performances of the LMI and the gamma-iteration

The efficiency and robustness of the optimal solution are interesting aspects of the fault detection filtering problem. Consequently two different methods for solving MFARE are compared, namely the LMI formulated as a linear objective minimization problem and the numerically implemented gamma-iteration.

The results of the MATLAB simulations are shown in Table 1.

Table 1. Comparison of the different solutions for the MFARE

Performance-measure	LMI as an linear objective minimization problem	gamma-iteration
γ_{min}	4.9704	4.9698
K	278.80 -52.70 0.0000 -52.70 1308.4 0.0000 -0.500 0.600 0.0000	257.2236 -39.2216 0.0000 -39.2216 699.2298 0.0000 -0.7934 1.6744 0.0000
A-KC’	-284.100 57.400 28.500 103.500 -1465.40 0.0000 0.500 -0.200 -9.100	-262.4879 43.9532 28.5021 89.9913 -856.2125 0.0000 0.7934 -1.2457 -9.0909
eig(A-KC’)	-1470.400 -279.100 -9.0	-862.8079 -255.9683 -9.0152
$\|\|H_{\varepsilon\omega}(s)\|\|_\infty$	4.4345	3.4047
Number of the iteration	9	21
Computation cost (sec)	0.1	1

The matrix of the absolute fault between the two solutions:

(45) $$K_{LMI} - K_g = \begin{bmatrix} 21.5764 & -13.4784 & 0 \\ -13.4784 & 609.1702 & 0 \\ 0.2934 & -1.0744 & 0 \end{bmatrix}.$$

From the simulation results of the comparison of the two different methods it can be concluded that each one gives an optimal solution. To be more precise the minimization algorithm has been applied until the satisfaction of the positive definiteness. As it can be seen in the Table 1, the smallest γ_{min} value could be reached using the simple gamma-iteration, but the result obtained on this way is just slightly different from the result obtained using LMI. The maximum absolute fault between the two solutions was 21.5764. However, the higher filter gain obtained in case of LMI suggests that the filter may be faster but less effective against disturbance. On other hand, by the gamma-iteration the burden of successive numerical computation of the quadratic matrix equality and repeated calling the function CARE as well resulted in a significant computation cost, as it can be seen in Table 1. We have found the solutions by running the codes in MATLAB CPU a PC with Intel® Celeron® CPU B815 (1.60 GHz). From these results it is visible that modern computation methods as a LMI are more capable to handle such complex mathematical problems as a solution of the MFARE.

However, the algorithm of gamma-iteration makes it possible to examine the solution for MFARE during the iteration. For example, it is easy to analyse the impact of gamma value on the number of iteration steps or the impact of changing of the disturbance on the optimal solution. One can easily perform experiments and get answers e.g. to the following questions: How does the iteration converge? How do the eigenvalues of the decision variable change? How close are they to the imaginary axis? How are they distributed? How does the filter gain change by reduction of the value of gamma? All these issues can be easily examined, step by step during the iterations, which can also be useful for better understanding the theory of H-infinity filtering.

Conclusion

In our paper we have performed a benchmark based on two concepts for solution of the MFARE. First the algorithm gamma-iteration was implemented, then the MFARE was formulated as a LMI and solved it as a linear objective minimization problem. From the simulation results of the LMI, it can be concluded, that both methods, the gamma-iteration and the LMI formulated as a linear objective minimization problem, are capable for solving the MFARE. Moreover, they deliver only slightly different results, but the LMI lead to an optimal solution much faster in about 100ms. However, the analytically implemented gamma-iteration, gives much more flexibility to examine the entire minimization process. For this reason we propose using both approaches, that is, using the gamma-iteration in the preliminary stage in order to perform an analysis and using LMI either in the stage of the synthesis to better perform the implementation. Our further work will include an extension this LMI problem to use it for switched linear H-infinity filtering.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

***Author**: Msc. Zsolt Horváth, School of Postgraduate Studies of Multidisciplinary Technical Sciences, Faculty of Technical Sciences, Széchenyi István University, Győr, E-mail : Zsolt2.Horvath@audi.hu.*

REFERENCES

[1] Edelmayer A., Bokor J., Keviczky L., An H_∞ Filtering Approach to Robust Detection of Failures in Dynamical Systems, *in Proc. 33th Annual Decision and Control*, Conf., Buena Vista, USA, 1994, pp. 3037-3039.
[2] Edelmayer A., Bokor J., Keviczky L., An H_∞ Filter Design for Linear Systems: Comparison of two Approaches, *IFAC 13th Triennial World Congress*, San Francisco, USA, 1996
[3] Yaesh I. , Shaked U., Game Theory Approach to Optimal Linear State Estimation and Its Relation to the Minimum H_∞-norm Estimation, *IEEE Trans. Aut. Control, AC-37(6)*, 1992, pp. 828-831.
[4] Chen J., Patton R.J., Robust Model-Based Fault Diagnosis for Dynamic Systems, *First Edition, Springer Science & Business Media*, New York, 1999
[5] Matusu R., Linear Matrix Inequalities and Semidefinite Programming: Applications in Control, *Internal Journal of Mathematical Models and Methods in Applied Sciences, Vol. 8*, 2014
[6] Gahinet P., Apkarian P., A linear matrix inequality approach to H_∞ control, *International Journal of Robust and Nonlinear Control, Vol.4*, 1994, pp.421–448.
[7] Iwasaki T., Skelton R.E., All controllers for the general H_∞ control problem: LMI existence conditions and state space formulas, *Automatica, Vol. 30*, 1994, pp. 1307–1317.
[8] Edelmayer A., Fault detection in dynamic systems: From state estimation to direct input reconstruction, *Universitas-Győr Nonprofit Kft.*, Győr, 2012
[9] Chong E. K. P., Zak S. H., An Introduction to Optimization, *4th Edition, Wiley*, New Jersey, 2013
[10] Ostertag E., Mono- and Multivariable Control and Estimation: Linear, Quadratic and LMI Methods, *Matematical Engineering, Vol.2*, Springer, Berlin, Heidelberg, 2011
[11] Ankelhed D., On the design of low order H_∞ controllers, *Ph.D. These, Linköping University*, Linköping, 2011
[12] Duan G., Yu R., H. H., LMIs in Control Systems: Analysis, Design and Applications, *CRC Press*, Boca Raton, 2013
[13] Boyd S., Ghaoui L.E., Feron E. and Balakrishnan V., Linear Matrix Inequalities in System and Control Theory, *SIAM,* Philadelphia, 1994
[14] Bokor J., Gáspár P., Szabó Z., Robust Control Theory, *Typotex*, Budapest, 2013
[15] Lunze J., Regelungstechnik 2 – Mehrgrößensysteme, *Digitale Regelung, Springer, 7. Auflage*, 2013
[16] Jankovic M., Kolmanovsky I., Robust Nonlinear Controller for Turbocharged Diesel Engines, *Procedings of the American Control Conference,* Philadelphia, 1998
[17] Jung M., Mean-value modelling and robust control of the airpath of a turbocharged diesel engine, *Ph.D. These, University of Cambridge*, 2003
[18] Horvath Zs., Edelmayer A., LTI-modelling of the Air Path of Turbocharged Diesel Engine for Fault Detection and Isolation, *Mechanical Engineering Letters, Vol. 14*, Gödöllő, 2016, pp. 172-188.
[19] Herceg M., Nonlinear Model Predictive Control of a Diesel Engine with Exhaust Gas Recirculation and Variable Geometry Turbocharger, *Diploma Thesis, Slovak University of Technology in Bratislava*, Bratislava, 2006
[20] Yung C. F., Reduced-order H_∞ controller design: An algebraic Riccati equation approach, *Automatica, Vol. 36*, 2000, pp. 923–926.
[21] Lanzon A., Feng Y., Anderson B. D. O. and Rotkowitz M., Computing the Positive Stabilizing Solution to Algebraic Riccati Equations With an Indefinite Quadratic Term via a Recursive Method, *IEEE Trans. Aut. Control, Vol. 53*, NO. 10. November, 2008, pp. 2280–2291.
[22] https://de.mathworks.com/help/control/ref/care.html?requestedDomain=www.mathworks.com
[23] Horvath Zs., Edelmayer A., Solving of the Modified Filter Algebraic Riccati Equation for H-infinity fault detection filtering, *Acta Universitatis Sapientiae Electrical and Mechanical Engineering, Vol. 9*, 2017, pp. 57-77.
[24] Gahinet P., Nemirovski A., Laub A.J. and Chilali M., LMI Control Toolbox for Use with Matlab, *The MathWorks Inc.*, Natick, 1995