

Unfalsified Control based on the \mathcal{H}_∞ controller parameterizationR. S. Sánchez-Peña^{a,b,*} and P. Colmegna^a and F. Bianchi^c^a*Depto. Matemática, Instituto Tecnológico de Buenos Aires, Av. Madero 399, (C1106ACD) C.A.B.A., Argentina.;*^b*CONICET, Argentina.*^c*Catalonia Institute for Energy Research, IREC, Jardins de les Dones de Negre 1, 08930 Sant Adrià de Besòs (Barcelona), Spain.*

(v4.1 released July 2013)

This paper presents an implementation of the Unfalsified Control (UC) method using the Riccati-based parameterization of \mathcal{H}_∞ controllers. The method provides an infinite controller set to (un)falsify the real-time data streams seeking for the best performance. Different sets may be designed to increase the degrees of freedom of the set of controller candidates to perform UC. In general, a set of m central controllers could be designed, each one seeking different objectives and all with their own parameterization as a function of a stable and bounded transfer matrix. For example, one controller parameterization could be designed to solve the robust stability of a model set which covers the physical system, therefore guaranteeing feasibility. The implementation requires the on-line optimization of either quadratic fractional or quadratic problems, depending on the selection of the cost function. A MIMO time-varying model of a permanent magnet synchronous generator illustrates the use of this technique.

Keywords: Unfalsified control, \mathcal{H}_∞ controller parameterization, linear fractional transformation**1. Introduction**

An important area of research is model-free and data driven control design (Parastvand and Khosrowjerdi (2013); Zhai et al. (2013)), where Unfalsified control (UC) can be included. The latter has been introduced in 1997 in Safonov and Tsao (1997) and developed since then by different researchers (see also Stefanovic and Safonov (2011); Jin and Safonov (2012); Safonov (1996); Tsao and Safonov (2001); Ingimundarson and Sánchez Peña (2008); Cabral and Safonov (2004); van Helvoort et al. (2007); Jin et al. (2011)). Basically, it consists of a set of candidate controllers and an algorithm (falsifier) that decides which one will achieve the best performance. The falsifier selects the most suitable controller based only on the information provided by measures of the input and output of the system.

It has several important conceptual advantages:

- It is implemented as a model-free procedure. No linearity, time-invariance or finite dimension is assumed on the system's model, and neither on the noise or perturbations.
- It is solely based on the real time input/output (I/O) data, which can be obtained from the open or closed loop behavior. No other parameter needs to be measured in real time as in the case of LPV control, nor any identification procedure needs to be performed as in classical adaptive control.
- The only objective to seek is the best possible performance, which is measured by a predetermined cost function.
- Controller selection can be performed without actually connecting it into the closed-loop in order not to perturb the system.

Although the UC implementation is not model-based, in many cases different model-based design procedures may be used in order to produce a set of candidate controllers. In addition, the falsification based on the cost function needs to be carried out in real time, which in many cases could be computationally demanding. There are basically two ways to implement an UC in real applications, which are based on *a priori* or *a posteriori* information. These two alternatives create a memory vs. computation time compromise.

- (1) Design a finite number of controllers **off-line** based on *a priori* information on all plausible future situations, including failures (Ingimundarson and Sánchez Peña (2008)). Based on an adequate cost function, select the controller that outperforms all others. This necessarily implies the use of more memory to keep the dynamics of several controllers during

*Corresponding author. Email: rsanchez@itba.edu.ar

implementation.

- (2) Establish a general controller structure and select its parameters **on-line** in order to optimize the cost function. This approach is based solely on the *a posteriori* information, and is aligned with the UC concept. Here in general, a heavier computational burden is necessary.

In this last category we can mention the ellipsoidal method (Cabral and Safonov (2004); van Helvoort et al. (2007)), which due to a particular selection of cost function and controller structure, may be optimized analytically. Nevertheless it does not guarantee feasibility¹ and the method does not generalize gracefully from SISO to MIMO models (van Helvoort et al. (2007)). Some recent references can be found in Wonghon and Engell (2012); Neamtu and Stoica (2011) and Wang et al. (2011).

A comment concerning cost functions in UC is useful here. A *detectable* cost function V (see Stefanovic and Safonov (2011)) reveals closed-loop instability, although in practice it is not clear how long it takes to detect it². Previous to this fact, feasibility should be guaranteed.

Here we seek to combine both approaches, 1. and 2., using all possible *a priori* information in order to design the controller structure (and guarantee feasibility), but accessing only the real-time *a posteriori* information at the implementation stage. In addition to cost detectability, a robust stabilizing controller set is designed in order to have a fast way to stabilize the loop. Meanwhile, another set may seek for the best performance measure.

The paper is organized as follows. The main results are presented in next Section, and are illustrated in Section 3 by their application to a MIMO time-varying model of a permanent magnet synchronous generator. Final conclusions are drawn in Section 4.

Throughout the paper, the (lower) LFT between two LTI systems M and Q is represented by

$$F_\ell(M, Q) = M_{11} + M_{12}Q(I - M_{22}Q)^{-1}M_{21}, \text{ where } M = \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix}.$$

The norm used in the cost function V is a truncated 2-norm defined as follows: $\|x\| = \sqrt{\int_0^{T_a} x^T(t)x(t)dt}$. Finally, \mathbb{Z}^+ denotes the set of positive integers.

2. \mathcal{H}_∞ -parameterization based UC

2.1 UC background

The UC concept proposed by Safonov and Tsao (1997) is illustrated in Figure 1. The performance specifications are stated as a cost-function V depending on the reference r and on the open-loop input u and output y . As a consequence, the performance specifications define a subset $\mathcal{T}_{spec} = \{(r, u, y) : V(r, u, y) < \eta\}$. The only information about the system is the set of measures $\mathcal{Z} = \{(u(\tau), y(\tau)), 0 \leq \tau \leq t\}$. The candidate controller K_i belongs to the set

$$\mathcal{K}_i \triangleq \left\{ (r, u, y) : u = K_i \begin{bmatrix} r \\ y \end{bmatrix} \right\},$$

where K_i is “causally-left-invertible”, i.e., there exists K_i^{-1} that allows the computation of a fictitious reference r_f from (u, y) . This reference is the value that r would take if the controller K_i were inserted in the loop when the input and output of the plant are (u, y) . The fictitious reference can be computed from \mathcal{Z} and K_i without actually inserting the controller in the loop from

$$r_f = K_i^{-1}u + y. \quad (1)$$

In this framework, the controller K_i is said to be unfalsified by the experimental information \mathcal{Z} if $\mathcal{K}_i \cap \mathcal{Z} \cap \mathcal{T}_{spec} \neq \emptyset$, otherwise the controller is said to be falsified by the measured information. The problem is feasible if the set of candidate controllers includes at least one which stabilizes the system (page 18, Stefanovic and Safonov (2011)).

The selection of the most adequate controller, also denoted the falsifier procedure, according to the *a posteriori* information (u, y) relies on the evaluation of a cost-detectable function such as

$$V(u, y) = \frac{\|w_u u\|^2 + \|w_e e_f\|^2}{\alpha + \|r_f\|^2}, \quad (2)$$

¹The UC problem is feasible if there exists at least one stabilizing controller in the candidate set, see Stefanovic and Safonov (2011).

²There are no bounds on the time it takes for V to detect an unstable closed-loop system Dehghani et al. (2007).

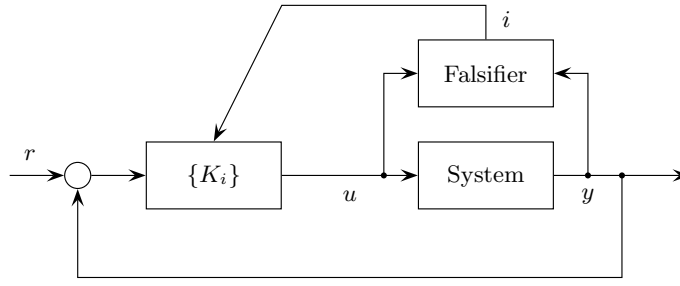


Figure 1. UC general scheme

where the fictitious reference has been defined in (1) and its corresponding tracking error is $e_f = r_f - y = K_i^{-1}u$. The linear filters w_u and w_e weight the control signal u and the fictitious error e_f . Constant α is defined in order not to unnecessarily increase the cost when $r_f \approx 0$. Then, at each iteration of the UC algorithm, the following optimization problem is solved

$$\min_{\mathcal{K}} V(u, y). \quad (3)$$

where $\mathcal{K} = \cup_{i=1}^m \mathcal{K}_i$. The results proposed here will guarantee feasibility and avoid the controller inverse in the falsifier procedure.

2.2 Controller sets

As mentioned previously, although the implementation of an UC is model-free, there is always some *a priori* information that can be used in order to design the set of candidate controllers or the controller structure. Here, the robust control framework and the \mathcal{H}_∞ -parameterization based on the results in Doyle et al. (1989) are used as a controller structure in the proposed falsification algorithm. In addition, this controller parameterization is also beneficial in order to avoid the “causally-left-invertible” assumption and the controller inverse in the cost function.

This procedure applies to MIMO systems that may be described by nonlinear equations. Such systems could be “covered” by a (bounded) set of models with dynamic uncertainty, as follows:

$$\mathcal{G} = \{F_\ell[G_0(s), \Delta(s)], \|\Delta(s)\|_\infty < 1\}. \quad (4)$$

The transfer function $G_0(s)$ is the nominal model and $\Delta(s)$ is an unknown but bounded matrix which weights the uncertainty frequency distribution and its I/O relation, e.g. additive uncertainty $\mathcal{G} = \{G_0(s) + W_\delta(s)\Delta, \|\Delta\|_\infty < 1\}$. This is a standard representation of systems in the robust control framework and can be obtained from robust identification procedures, see IEEE Special issue (1992); Chen and Gu (2000) and Chap. 10 of Sánchez Peña and Sznaiier (1998).

Nevertheless it could happen that there is no controller that could robustly stabilize such a large set of models, or that the resulting performance is not good enough. In that case, the actual system could be divided into m smaller regions and for the i -th region, a model set $\mathcal{G}_i = \{G_{0,i}(s) + W_{\delta,i}(s)\Delta_i(s), \|\Delta_i\|_\infty < 1\}$ is obtained. Next, $i = 1, \dots, m$ control problems can be configured as in Figure 2. There, the augmented model $G_{aug,i}$ is a combination of the nominal model $G_{0,i}(s)$, the robustness weight $W_{\delta,i}$ and a performance weight $W_{p,i}$, which represent the particular problem at hand.

Based on Doyle et al. (1989), for each augmented model $G_{aug,i}$ an optimal \mathcal{H}_∞ controller K_i can be parameterized as a function of a stable, bounded parameter Q_i . In addition, considering all augmented models $i = 1, \dots, m$ the set of all controllers is defined as:

$$\mathcal{K} \triangleq \{K_i(s) = F_\ell[M_i(s), Q_i(s)], \|Q_i(s)\|_\infty < \gamma_i, i = 1, \dots, m\}. \quad (5)$$

Each controller K_i solves a robust performance problem³, i.e. it ensures that the transfer matrix between $w \rightarrow z$ satisfies $\|T_{zw,i}\|_\infty < \gamma_i$ for all stable parameter Q_i with $\|Q_i(s)\|_\infty < \gamma_i$. The controller K_i corresponds to the i -th augmented model $G_{aug,i}$ as illustrated in Figure 2. The notation for the central controller is as follows $K_{0,i} = K_i$ for $Q_i = 0$.

We can interpret this result in several ways. The first interpretation has been stated previously, i.e. m different uncertainty sets \mathcal{G}_i which “cover” a physical system in order to achieve desired levels of performance in each region. These regions could be linearizations of a nonlinear model at different working points.

³Note that in the context of \mathcal{H}_∞ control theory, performance is based on an infinite amount of data, whereas in UC it refers to a finite amount of data points \mathcal{Z} , as defined in subsection 2.1.

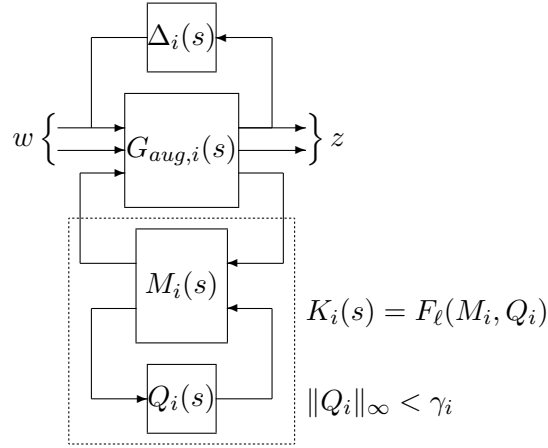


Figure 2. i -th augmented plant $G_{aug,i}$ and \mathcal{H}_{∞} controller parameterization structure.

Another interpretation is that one of the uncertainty sets, say \mathcal{G}_k , covers the complete system and as a consequence, any member of the robust controller set $K_k(s) = F_{\ell}[M_k(s), Q_k(s)]$ guarantees closed loop internal stability of all members of \mathcal{G} in equation (4), therefore assuring the feasibility of the unfalsification problem. Nevertheless, this controller subset K_k will provide an extremely low performance in practice since the uncertainty set \mathcal{G} could be very large and include more models than the actually necessary.

This leads to the third interpretation which circumvents this last issue. The m augmented model set $G_{aug,i}$ and their corresponding \mathcal{H}_{∞} controllers K_i could be designed with the same augmented structure but with different uncertainty and performance weights, i.e. different objectives. For example, different controllers could be designed focused on pre-defined faults that could appear in the system (fault-tolerance based UC can be found in Ingimundarson and Sánchez Peña (2008); Coito and Palma (2011); Jain et al. (2012)). Another important controller set would be the robust stability one K_k mentioned previously (assuring feasibility) or others which only seek for performance in a smaller region. The falsification algorithm will decide which one provides the better performance at each time, depending on the actual I/O data (u, y) . The sets could be shrunk at each iteration based on reducing the size of the γ_i .

The selection of the most adequate controller according to the *a posteriori* information (u, y) relies on the evaluation of a cost-detectable function and the optimization problem (3). It can be summarized in the following pseudo-code.

- (1) Initialization: $i \leftarrow 1, t \leftarrow 0, K_t \leftarrow K_{0,i}$;
- (2) $t \leftarrow t + T_s$, collect data u and y into U and Y , respectively;
- (3) IF $t/T_a \in \mathbb{Z}^+$
 THEN
 FOR $i \leftarrow 1 : m$
 $V_{\star}^i \leftarrow \arg \min_{\|Q_i\|_{\infty} < \gamma_i} V(U, Y, K_{0,i})$;
 END
 $[V_{\star}, i^{\star}] \leftarrow \min_i \{V_{\star}^i\}$;
 IF $V(U, Y, K) > V_{\star} + \epsilon$;
 THEN
 $K_t \leftarrow K_{i^{\star}}$;
 ENDIF;
 clear U and Y ;
 ENDIF;
- (4) Go to 2. \diamond

The index i denotes which is the current central controller ($K_{0,i}$), and T_s and T_a are the sample time, and the interval between two consecutive falsification tests, respectively. In the example section, T_a has been selected as a multiple of T_s . K_t is the current controller inserted in the closed loop, with t denoting the time.

The algorithm starts with $K_{0,1}$, the robust controller which stabilizes the set of models covering the physical system. At times T_s in step 2, the value of signals u and y are accumulated in vectors U and Y , respectively. At each period of time T_a , i.e. $t/T_a \in \mathbb{Z}^+$ in step 3, a new falsification test starts. For each $i = 1, \dots, m$, an optimization procedure searches over all elements of controller set K_i , i.e. all $\|Q_i\|_{\infty} < \gamma_i$, in order to minimize the cost function, say V_{\star}^i . Next, the following is computed $V_{\star} = \min_i \{V_{\star}^i\}$, and the index associated with V_{\star} is

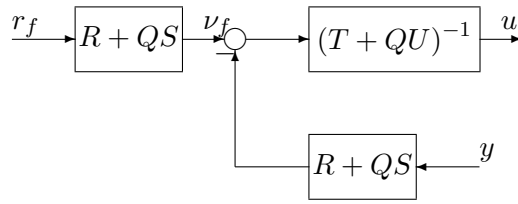


Figure 3. Controller structure based on its \mathcal{H}_∞ parameterization to avoid computing its inverse.

saved in i^* . There should always be a set of robust stabilizing controllers (such as the set K_k mentioned previously) in order to guarantee feasibility. The current controller is falsified if its cost function is greater than $V_\star + \epsilon$, otherwise it will be active for another interval T_a . In practice, $\epsilon > 0$ is defined in order to avoid infinite switching.

2.3 Search for the most adequate parameter Q

The previous problem is particularized for a given controller set K_i , and the UC algorithm must determine the most adequate parameter Q_i . This implies solving the optimization problem

$$\min_{\|Q_i\| < \gamma_i} V(u, y, K_i). \quad (6)$$

To provide an explicit solution for the problem (6), it is necessary to define the cost function V . Here, two different detectable cost functions will be proposed, the one in (2) related to a standard mixed sensitivity problem, and the one related to the ellipsoidal UC in van Helvoort et al. (2007).

In both cases the cost function involves computing the fictitious reference $r_{f,i}$ from the inverse of the controller to be able to evaluate V without actually inserting K_i in the loop. From here on, to simplify the notation, the index i will be eliminated.

Taking into account the \mathcal{H}_∞ controller parameterization and the LFT manipulations in Chapter 10, Zhou et al. (1996):

$$K^{-1} = (T + QU)^{-1} (R + QS),$$

where

$$\begin{bmatrix} R & S \\ T & U \end{bmatrix} = \begin{bmatrix} M_{12}^{-1} & -M_{22}M_{12}^{-1} \\ M_{12}^{-1}M_{11} & M_{21} - M_{22}M_{12}^{-1}M_{11} \end{bmatrix}, \quad M_i = \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix}.$$

Now consider Q as a linear combination of filters parametrized by a vector variable $\lambda \in \mathbb{R}^n$, i.e.

$$Q = \sum_{j=1}^n \lambda_j f_j, \quad \begin{cases} \|f_j\|_\infty < 1, & j = 1, \dots, n \\ \lambda \in \Lambda \triangleq \left\{ \sum_{j=1}^n |\lambda_j| \leq \gamma \right\} \end{cases} \quad (7)$$

where $\{f_j, j = 1, \dots, n\}$ is a set of predefined transfer matrices. As a consequence the required condition $\|Q\|_\infty < \gamma$ is satisfied. In order for the algorithm to learn from past information, the controller set will shrink at each step according to the optimal value of λ , i.e. its bound γ will decrease.

According to Stefanovic and Safonov (2011) (page 26) the cost function V in (2) may be modified in order not to invert the controller, as follows (see Fig. 3).

$$\begin{aligned} \tilde{V}[u, y, Q(\lambda)] &= \frac{\|w_u u\|^2 + \|w_e \tilde{e}_f\|^2}{\alpha + \|\nu_f\|^2} = \frac{\|w_u u\|^2 + \|w_e [T + Q(\lambda)U] u\|^2}{\alpha + \|[R + Q(\lambda)S] y + [T + Q(\lambda)U] u\|^2} \\ &\triangleq \frac{n_V(\lambda)}{d_V(\lambda)} \end{aligned} \quad (8)$$

Here $\nu_f = (R + QS)r_f$ and $u = (T + QU)^{-1}\tilde{e}_f$. The problem to be solved is therefore:

$$\begin{aligned} \min_{\|Q\|_\infty < \gamma} \tilde{V}[u, y, Q(\lambda)] &= \min_{\lambda \in \Lambda} \frac{\|w_u u\|^2 + \|w_e [T + Q(\lambda)U] u\|^2}{\alpha + \|(Ry + Tu) + Q(\lambda)(Sy + Uu)\|^2} \\ &= \min_{\lambda \in \Lambda} \frac{\beta + \|v + W\lambda\|^2}{\alpha + \|x + Z\lambda\|^2} \end{aligned} \quad (9)$$

where

$$\begin{aligned} v &= w_e M_{12}^{-1} M_{11} u, & w_j &= w_e f_j (M_{21} u - M_{22} v), & W &= [w_1 \cdots w_n] \in \mathbb{R}^{m \times n} \\ x &= M_{12}^{-1} (y + M_{11} u), & z_j &= f_j (M_{21} u - M_{22} x), & Z &= [z_1 \cdots z_n] \in \mathbb{R}^{m \times n}, \end{aligned}$$

$\beta = \|w_u u\|^2 \geq 0$, and $\alpha > 0$. Here (v, x, w_j, z_j) are either discrete or continuous time (vector) signals.

It is not the purpose of this work to explore into the non-convex optimization quadratic fractional program (9). This is an active line of research, some results can be found in Fang et al. (2009) and an algorithmic solution in Enkhbat et al. (2011).

Alternatively, a cost function similar to that used in van Helvoort et al. (2007) related to the analytic solution via the ellipsoidal optimization algorithm is:

$$V[u, y, Q(\lambda)] = \{Q(\lambda) : |e_f(\lambda)| \leq \Delta - |w_u u|\} \quad (10)$$

Here $\Delta(t) \geq 0$ is a bound on the tracking error and $e_f = G_m r_f [Q(\lambda)] - y$, with G_m the desired closed-loop dynamics.

Proceeding as in the previous case, in the cost function the signals are replaced as follows: $e_f \rightarrow \tilde{e}_f$ and $r_f \rightarrow \nu_f$. Also $\tilde{e}_f = G_m \nu_f - (R + QS)y$, therefore:

$$\begin{aligned} &\{\lambda \in \Lambda : |(G_m - I)[R + Q(\lambda)S]y + G_m[T + Q(\lambda)U]u| \leq \Delta - |w_u u|\} \\ &= \{\lambda \in \Lambda : |\xi + W_\ell Q(\lambda)W_r \leq \phi|\} \\ &\xi \triangleq G_m(Ry + Tu) - Ry, \quad \phi \triangleq \Delta - |w_u u| \\ &W_\ell \triangleq [G_m - I], \quad W_r \triangleq \begin{bmatrix} Sy + Uu \\ Sy \end{bmatrix} \end{aligned} \quad (11)$$

As in van Helvoort et al. (2007), the simpler (quadratic) problem (11) is obtained, which can be solved via the ellipsoidal algorithm with proven convergence guarantees. Nevertheless, to apply this optimization to MIMO systems, the procedure in van Helvoort et al. (2007) should be followed, and the absolute value in the above equations $|\cdot|$ should be considered element-wise.

2.4 Controller set design

The design of the controller set can be summarized as follows:

- (1) Define m different nominal models $G_{0,i}$, uncertainty and performance weights $(W_{\delta,i}, W_{p,i})$ according to the problem at hand.
- (2) Compute the augmented model and design a set of \mathcal{H}_∞ controllers for each $i = 1, \dots, m$, i.e. $K_i(s) = F_\ell(M_i, Q_i)$.
- (3) At each iteration of the UC implementation, the optimization problem in (3) is solved as follows:

$$\min_{\{\|Q_i\|_\infty < \gamma_i\}_{i=1}^m} V(u, y, Q_1, \dots, Q_m) \quad (12)$$

where V is a detectable cost function.

To assure feasibility, a general model of the system or a set of I/O data vectors and an identification procedure can “cover” the physical system by means of a family of models with dynamic uncertainty, say $\mathcal{G}_k = \{G_{0,k}(s) + W_{\delta,k}(s)\Delta_k(s), \|\Delta_k\|_\infty < 1\}$. Based on this, K_k can be designed and included in the controller bag. Any member of this set of controllers guarantees closed loop internal stability of the system. Again, as mentioned in footnote 3, here stability is defined in a robust control sense, i.e. based on an infinite number of data points.

In some cases, to simplify the search in (12), only the central controllers could be considered, i.e. $Q_i \equiv 0$, $i = 1, \dots, m$, therefore the set of controllers reduces to $\mathcal{K} = \{K_{0,i}, i = 1, \dots, m\}$ and the search is over a finite set.

2.5 Advantages and practical issues

This method offers an interesting alternative to other popular time-varying controllers like the case of LPV. In many applications it is not possible to measure in real-time a parameter that follows the time-varying dynamics. Therefore this method produces a time-varying controller that follows the dynamics based solely on I/O measurements, due to the lack of a representative parameter. On the other hand, the main drawback of this method is the need to perform a real-time optimization, but this is no different from other popular methods like MPC control or classical adaptive control.

Next, some practical considerations related to the implementation of this procedure are presented.

- The filter selection should consider the actual bandwidth of the system. A set of second order Kautz filters could be used for example. Also, the total number of filters n has a practical connotation, due to the fact that the order of Q and as a consequence of K , depends on it.
- In many cases for numerical reasons, different cost functions could be used. This is the case when controllers are designed seeking different objectives, e.g. controller K_j focuses on robust stability to guarantee feasibility and controller K_ℓ seeks for the best performance. The weights in each of these cost functions V_j and V_ℓ respectively, will be different. Therefore, a numerical comparison should be made among them in order for the falsifier to decide when performance or stability should be prioritized. To this end, the following comments are in order.

For the mixed sensitivity problem in equation (2), the following equivalence holds:

$$\begin{aligned} \min \left\| \begin{bmatrix} W_u(s)K(s)S(s) \\ W_e(s)S(s) \end{bmatrix} \right\|_\infty^2 &\iff \min \left\| \begin{bmatrix} W_u(s)u \\ W_e(s)e_f \end{bmatrix} \right\|_2^2, \quad \forall \|r_f\|_2 \leq 1 \\ &\iff \min (\|W_u u\|_2^2 + \|W_e e_f\|_2^2), \quad \forall \|r_f\|_2 \leq 1 \end{aligned}$$

The latter is the numerator of the cost function V which should be minimized in real time. It should be pointed out that this holds for all $\|r_f\|_2 \leq 1$. Also, for given weights $(W_u, W_e)_j$ or $(W_u, W_e)_\ell$, the controller K_j or K_ℓ achieve the optimals V_*^j or V_*^ℓ , respectively. Nevertheless, for a particular value of $\|r_f^*\|_2 \leq 1$ only the following can be stated: $V^j(r_f^*, K_j) \geq V_*^j$ and $V^\ell(r_f^*, K_\ell) \geq V_*^\ell$, but there is no obvious relation between $V^j(r_f^*, K_\ell)$ and $V^\ell(r_f^*, K_j)$. As a consequence, the weights in the cost function should be carefully selected in order to compare controllers in the set K_ℓ with the robustness controller K_j .

- Note that the feasibility is guaranteed in a LTI sense, but not in general, when switching among models is performed. This is due to the fact that the central \mathcal{H}_∞ controller covers all LTI models and guarantees their stability, but it does not cover the switching among them which is related with the nonlinear behavior of the system. For this reason a detectable cost function is selected, in order to detect possible instabilities during switching.

3. Application example

The torque control of a permanent magnet synchronous generator (PMSG) is used to illustrate the proposed method. The system is governed by the following differential equations

$$\dot{x} = \begin{bmatrix} -R/L & \omega_e \\ -\omega_e & -R/L \end{bmatrix} x + \begin{bmatrix} -1/L & 0 \\ 0 & -1/L \end{bmatrix} (v_\ell + v_g), \quad (13)$$

where

$$x = \begin{bmatrix} i_{gq} \\ i_{gd} \end{bmatrix}, \quad v_\ell = \begin{bmatrix} v_{\ell q} \\ v_{\ell d} \end{bmatrix}, \quad v_g = \begin{bmatrix} v_{gq} \\ v_{gd} \end{bmatrix}$$

are the generator currents, the converter and generator's quadrature and direct voltages, respectively. The resistance and inductance are $R = R_c + R_g$ and $L = L_c + L_g$, the subscripts meaning converter and generator respectively. The electrical frequency ω_e is connected with the mechanical speed ω_g by the expression $\omega_e = p\omega_g$, where p is the number of pole pairs. This is clearly a MIMO linear time-varying model due to the fact that the system matrices depend on ω_e .

This is a common problem in modern wind turbines, in which a power converter must control the generator torque and the reactive power to ensure maximum power capture from the wind. The objective can be cast as the tracking of two reference currents i_q^* and i_d^* . These references are computed by a high level control that is out of the scope of this application example.

The results presented here are very general, and the candidate set of controllers is very large. In this example we have simplified the search over two model sets: one covers the whole system, and the other covers a small neighborhood around the nominal model. The first one produces a single central controller which guarantees robust stability and hence feasibility. The second design is focused on performance and the search is over the $\|Q\|_\infty < \gamma$ parameter. In addition, equation (2) has been used as the cost function, and a standard optimization procedure was implemented in real-time (command `fmincon` from Matlab[®]).

First, to design the robust stabilizing controller, the time-varying model (13) is covered by set $\mathcal{G} = G_{nom} + W_\delta \Delta$, with $\|\Delta\|_\infty < 1$. The uncertainty representation is determined from a set of LTI models corresponding to (13) evaluated at eight different values of $\omega_e \in \{20\pi, \dots, 160\pi \text{ rad/s}\}$. The nominal model achieves the least uncertainty bound, and corresponds to the system at $\omega_e = 40\pi \text{ rad/s}$. Figure 1 (left) illustrates the different errors between the nominal and all other linearized models. The model errors $\delta G_i(\omega) = \bar{\sigma}(G_i - G_{nom})$ are covered by a weight which represents the additive uncertainty among models, i.e. $W_\delta = \frac{2.3(s+33)}{(s+95)} I_{2 \times 2}$, where $I_{2 \times 2}$ is the identity matrix. The sampling time is $T_s = 1 \text{ msec}$ and the falsification period is $T_a = 250 \text{ msec}$.

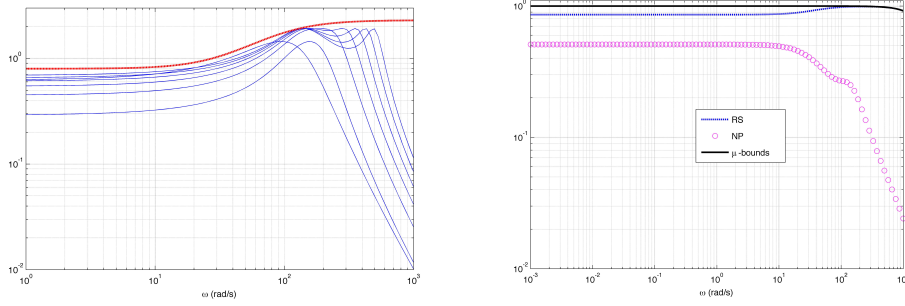


Figure 1. (left) Differences between all linearized models and the nominal G_{nom} (full), covered by W_δ (dashed). (right) Robust stability and Nominal and robust performance conditions for the design.

A central \mathcal{H}_∞ controller $K_{RS}(s)$ has been designed focused on robust stability. This implies a controller that should meet $\|W_\delta(s)K(s)S(s)\|_\infty < 1$, where $S(s)$ is the sensitivity function. In practice, this is solved as a mixed sensitivity problem with a very low performance weight. Here the performance is stated as a constraint on the error, therefore the sensitivity function was penalized with $W_p = 2 \times 10^{-4} I_{2 \times 2}$, in order to meet the robustness constraint, as seen in Figure 1 (right). The controller $K_{RS}(s)$ is obtained by solving

$$\min_{K_{RS}(s)} \left\| \frac{W_\delta(s)K(s)S(s)}{W_p(s)S(s)} \right\|_\infty \quad (14)$$

An integrator has been inserted in the loop to ensure a zero steady state error. The time response of all models connected with the central controller $K_{RS}(s)$ is presented in Figure 2, excited by a staircase signal. The time responses differ significantly with the rotational speed, and low rotation speeds tend to produce more oscillatory results.

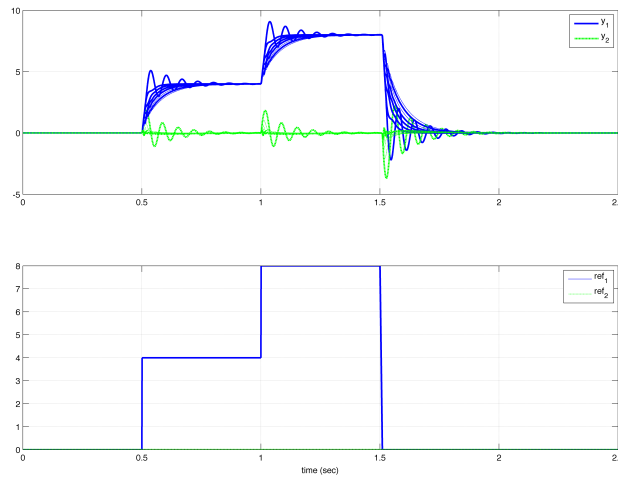


Figure 2. (upper) Time response of all (8) models connected with the central controller K_{RS} . (lower) Input reference signal $[i_{gq} \ i_{gd}]^T$.

The central controller for performance K_P was designed with the same mixed-sensitivity scheme than the

robust stability case, but the weighting functions in (14) were

$$W_\delta(s) = \frac{(s + 33)}{10(s + 95)} I_{2 \times 2}, \quad W_p = 3 \times 10^{-3} I_{2 \times 2}$$

The controller set is then defined as $K_P = F_\ell[M, Q(\lambda)]$, with $Q(\lambda)$ as in equation (7) and λ obtained at each step from the real-time optimization problem (11), based solely on real-time I/O data (u, y) . For this case, due to the fact that the robust controller produces a bandwidth of around 30 rad/s, two first-order filters have been considered for Q :

$$Q(s) = \frac{\lambda_1}{0.03s + 1} + \frac{\lambda_2}{0.01s + 1}$$

in order to produce a faster closed-loop response.

A comparison between the UC and the K_{RS} controller is performed for a parameter trajectory which starts at $t = 0$ s from $\omega_e = 130.8$ rad/s and linearly increases to 151.8 rad/s at $t = 2$ s. and then decreases abruptly to $\omega_e = 130.8$ rad/s. The reference is similar to the staircase signal in Figure 2.

Figure 3 shows the responses of the closed loop system with the robust controller K_{RS} and the UC. The evolution of the parameters λ_1 and λ_2 can be seen in Figure 4 and the corresponding control action for both controllers in Figure 5. Clearly the UC outperforms the fixed controller K_{RS} at the cost of a longer computational time at implementation (3 to 1), due to the on-line optimization.

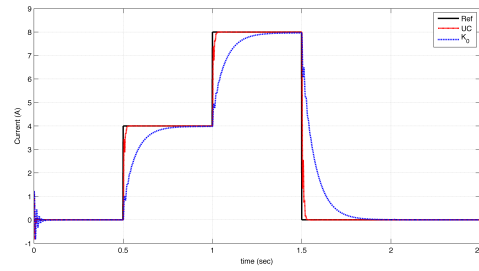


Figure 3. Reference and the closed loop responses with UC and K_{RS} .

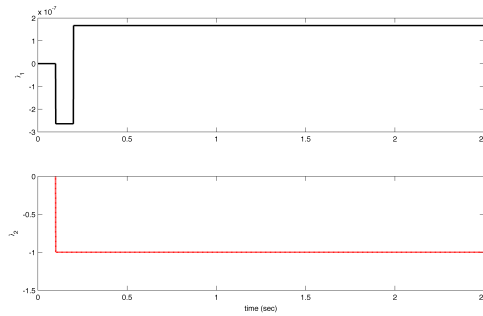


Figure 4. Evolution of the parameters λ in the UC controller.

Next we illustrate the use of the robust controller to guarantee closed-loop stability. According to equation (3), this should be performed automatically as part of the online optimization. For a $\omega_e = 41.86$ rad/s and the same reference signal, the response of the UC controller is shown in Figure 6. Note that at this speed the response is oscillatory and the K_P controller is applied at the beginning, but at $t = 1$ s the oscillations seem to increase and the robust controller K_{RS0} takes the command of the situation at $t = 1.5$ s, stabilizing the output.

4. Conclusions

Unfalsified control is an attractive adaptive approach in cases of lack of information about the plant. For example, this provides a time varying alternative to LPV control when no representative parameter can be measured in real

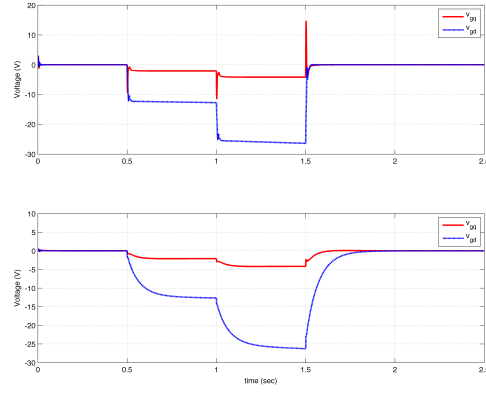


Figure 5. Control signals for UC (upper) and robust control K_{RS} (lower).

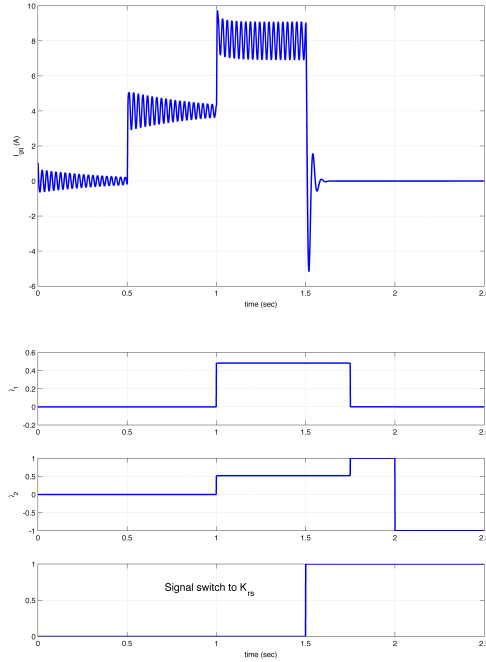


Figure 6. Automatic switch from the UC to the robust control K_{RS0} .

time. An important point in UC is the set of candidate controllers. Although the main claim of UC theory is that no *a priori* information is needed, in practice some information is necessary to ensure feasibility of the controller search. There are basically two approaches to define the controller set. One consists of a set with a finite number of elements and the other uses online optimization to compute the candidate controller for a given structure. Here an intermediate approach is proposed based on the parameterization of all optimal \mathcal{H}_∞ controllers. The set of controllers is defined as a finite set of central controllers and an on-line optimization procedure which finds the most suitable parameter Q for a particular central controller. Therefore, a trade-off can be reached between the amount of memory needed to store the controller set and the computational burden to find the optimal parameter. Another advantage of the proposed UC approach is that it allows a simpler extrapolation to MIMO systems than previous online optimization algorithms such as the ellipsoidal UC. This point is clearly shown in the example used to illustrate the application of the proposed control approach. Overall, the method presented here offers a fairly general set to implement UC on practical applications based on switching and optimizing linear controllers. Although the results are promising, further research should be made in order to simplify the online optimization procedure, which can be computationally-demanding for commonly used cost-functions.

Acknowledgments

The research of F.D. Bianchi was supported by the European Regional Development Funds (ERDF, “FEDER Programa Competitivitat de Catalunya 2007-2013”). R.S. Sánchez Peña and P. Colmegna were supported by the program PRH No. 71 (PICT 290 and PFDT) of the Ministry of Science, Technology and Innovation of Argentina.

References

- Cabral, F., and Safonov, M. (2004), “?Unfalsified Model reference adaptive control using the ellipsoidal algorithm,” *International Journal of Adaptive Control & Signal Processing*, 18(8), 683–696.
- Chen, J., and Gu, G., *Control-oriented System Identification: An \mathcal{H}_∞ approach*, Wiley & Sons (2000).
- Coito, F., and Palma, L.B. (2011), “Unfalsification based Fault Tolerant Control,” in *IEEE ETFA*.
- Dehghani, A., Anderson, B., and Lanzon, A. (2007), “Unfalsified Adaptive Control: A new controller implementation and some remarks,” in *Proceedings European Control Conference*, pp. 709–716.
- Doyle, J.C., Glover, K., Khargonekar, P.P., and Francis, B.A. (1989), “State-Space Solutions to Standard \mathcal{H}_2 and \mathcal{H}_∞ Control Problems,” *IEEE Transactions on Automatic Control*, 34(8), 831–847.
- Enkhbat, R., Bazarsad, Y., and Enkhbayar, J. (2011), “A Method for Fractional Programming,” *International Journal of Pure and Applied Mathematics*, 73(1), 93–99.
- Fang, S., Gao, D., Sheu, R.L., and Xing, W. (2009), “Global optimization for a class of fractional programming problems,” *Journal of Global optimization*, 45, 337–353.
- IEEE Special issue, (1992), “Special Issue on System Identification for Robust Control Design,” *IEEE Transactions on Automatic Control*, 37(7).
- Ingimundarson, A., and Sánchez Peña, R. (2008), “Using the Unfalsified Control Concept to achieve Fault Tolerance,” in *IFAC World Congress*, pp. 1236–1242.
- Jain, T., Yamé, J., and Sauter, D. (2012), “Model-free reconfiguration mechanism for Fault Tolerance,” *International Journal of Applied Math. Computer Science*, 22(1), 125–137.
- Jin, H., Chang, M., and Safonov, M. (2011), “A Fading memory Data-driven Algorithm for controller switching,” in *Proceedings Conference on Decision and Control*, IEEE.
- Jin, H., and Safonov, M. (2012), “Unfalsified adaptive control: Controller switching algorithms for nonmonotone cost functions,” *International Journal of Adaptive Control and Signal Processing*.
- Neamtu, A., and Stoica, A. (2011), “Unfalsified Control: Application to automatic flight control system design,” *INCAS Bulletin*, 3(3), 103–114.
- Parastvand, H., and Khosrowjerdi, M. (2013), “Controller synthesis free of analytical model: fixed-order controllers,” *International Journal of Systems Science*.
- Safonov, M., and Tsao, T. (1997), “The unfalsified control concept and learning,” *IEEE Transactions on Automatic Control*, 42(6), 843–847.
- Safonov, M.G. (1996), “Focusing on the Knowable,” *Control using Logic-based switching*, Springer-Verlag, pp. 224–233.
- Sánchez Peña, R.S., and Sznaier, M., *Robust Systems Theory and Applications*, John Wiley & Sons, Inc. (1998).
- Stefanovic, M., and Safonov, M., *Safe Adaptive Control*, Springer (2011).
- Tsao, T.C., and Safonov, M.G. (2001), “Unfalsified direct adaptive control of a two-link robot arm,” *International Journal of Adaptive Control and Signal Processing*, 15, 319–334.
- van Helvoort, J., de Jager, B., and Steinbuch, M. (2007), “Data-driven multivariable controller design using Ellipsoidal Unfalsified Control,” in *American Control Conference*, pp. 510–515.
- van Helvoort, J., de Jager, B., and Steinbuch, M. (2007), “Direct data-driven recursive controller unfalsification with analytic update,” *Automatica*, 43, 2034–46.
- Wang, L., Ding, Y., Hao, K., and Liang, X. (2011), Vol. 123, “An Immune-Enhanced Unfalsified Controller for a High-Speed Spinning Process,” *Knowledge Engineering and Management*, Springer Berlin Heidelberg, pp. 19–24.
- Wonghon, T., and Engell, S. (2012), “Automatic controller tuning via unfalsified control,” *Journal of Process Control*, 22(10), 2008–2015.
- Zhai, D., Zhang, Q.L., and Liu, G.Y. (2013), “Data-driven criteria synthesis of system with two-degree-of-freedom controller,” *International Journal of Systems Science*.
- Zhou, K., Doyle, J.C., and Glover, K., *Robust and Optimal Control*, Prentice-Hall (1996).