ISSUES, PROGRESS AND NEW RESULTS IN ROBUST ADAPTIVE CONTROL

Sajjad Fekri[1*], Michael Athans[1§], and Antonio Pascoal[1]

[1] *Institute for Systems and Robotics (ISR), Instituto Superior Técnico (IST), Lisbon, Portugal*
E-mail: {sfekri,athans,antonio}@isr.ist.utl.pt, Tel: +351-21-8418288, Fax: +351-21-8418291

SUMMARY

We overview recent progress in the field of robust adaptive control with special emphasis on methodologies that use multiple-model architectures. We argue that the selection of the number of models, estimators and compensators in such architectures must be based on a precise definition of the robust performance requirements. We illustrate some of the concepts and outstanding issues by presenting a new methodology that blends robust nonadaptive mixed μ-synthesis designs and stochastic hypothesis-testing concepts leading to the so-called Robust Multiple Model Adaptive Control (RMMAC) architecture. A numerical example is used to illustrate the RMMAC design methodology, as well as its strengths and potential shortcomings. The later motivated us to develop a variant architecture, denoted as RMMAC/XI, that can be effectively used in highly uncertain exogenous plant disturbance environments.

KEY WORDS: Multiple model adaptive control; adaptive control; robust control; robust adaptive control; robust μ-synthesis

1. INTRODUCTION

The solution to the so-called "adaptive control" problem is akin to the elusive search for the "Holy Grail" in the context of feedback control system design. In spite of forty years of research, several books and hundreds of articles we still lack, in our view, a universally accepted design methodology for adaptive control which is based on sound theoretical issues and suitable for engineering implementations in real-life control systems. In this paper we overview some recent progress in adaptive designs that employ multiple-models. Currently available results seem to

---

*S. Fekri is now Research Associate in Department of Engineering, Control & Instrumentation Research Laboratory, University of Leicester, Leicester, UK. E-mail: sf111@le.ac.uk, Tel: +44-116-2522874, Fax: +44-116-2522619 (Correspondences to: University Road, Leicester, LE1 7RH).
§M. Athans is also Professor (emeritus) of the Dept. of EE&CS, MIT, Cambridge, MA, USA.
An earlier version of this paper was based on a plenary talk by M. Athans and published in the Proceedings of the 2005 IFAC World Congress, Prague, Czech Republic, 2005 [1].
This work is based upon S. Fekri's doctoral thesis [2] and was supported in part by the Portuguese FCT POSI programme under framework QCA III and by project MAYA-Sub of the AdI.
This paper was recommended for publication in revised form by Associate Editor XXX. YYYYYYY.

be very promising, but still require a great deal of theoretical and pragmatic research to arrive at the "Holy Grail" of adaptive control. Thus in our view, *the* solution to the adaptive control problem is still not available.

We first discuss a general philosophy for designing "robust" adaptive multivariable feedback control systems for linear time-invariant (LTI) plants that include *both* unmodeled dynamics and uncertain real parameters in the plant state-space description. The adjective "adaptive" refers to the fact that the real parameter uncertainty and performance requirements require the implementation of a feedback architecture with better performance and greater complexity than that of the best possible fixed non-adaptive controller. The word "robust" refers to the desire that the adaptive control system remains stable and *also meets the posed performance specifications* for all-possible "legal" parameter values and unmodeled dynamics.

In order to place our remarks in a proper perspective and motivate the development of our new RMMAC method we pose a set of basic engineering questions that naturally arise when we deal with adaptive control:

(1). What do we gain by using adaptive control?

(2). How do we fairly *predict and compare* performance improvements (if any) of a proposed adaptive design vis-à-vis the "best nonadaptive" one?

(3). How do we design adaptive controllers with *guaranteed robust-stability and robust-performance* in the presence of unmodeled dynamics and unmeasurable plant disturbances and sensor noises?

(4). Is the increased complexity of an adaptive controller justified by the performance improvement? What should be the level of complexity for designer-specified adaptive system performance guarantees?

Even though there is no precise universally accepted definition of "adaptive control" the above questions are (or should be) at the heart of adaptive control research; they have motivated the perspective adopted in this paper.

It is important to stress at this point that the vast majority of approaches to adaptive control deal with the case of constant *uncertain* real parameters. Furthermore, they invariably focus upon stability without explicit quantitative specifications for the desired adaptive system performance. However, from an engineering perspective, the true value of an adaptive system can only be judged by its performance when the uncertain real parameters change *"slowly with time",* within their predefined limits.

Thus, one designs and tests an adaptive system for constant real parameters, using whatever theoretical approaches developed, but it should be also tested, and its performance evaluated, for time-varying parameters as well.

The current accepted concept of a robust (nonadaptive) feedback control system, for linear time-invariant (LTI) plants, is that the designed compensator must be such so as to guarantee (if possible) closed-loop stability and to also meet posed performance specifications, most often reflecting superior disturbance-rejection. This attribute is often referred to as *"stability- and performance-robustness."* The physical plant is assumed to belong to a "legal family" of possible plants, where the nominal plant together with frequency-dependent upper bounds on unmodeled dynamics and upper- and lower-bounds on key uncertain real parameters defines this "legal family" of plants. The performance specifications are *explicitly* stated in the frequency domain; they typically require superior disturbance-rejection in the lower frequency region while safeguarding for excessive control action at higher frequencies. Such robust nonadaptive compensators can be designed, more or less in a routine manner, using the

so-called *mixed-μ synthesis* methodology and associated MATLAB software. At present, robust synthesis can not deal with slowly-varying parameters, from a theoretical perspective, although some recent research in "Linear Parameter Varying (LPV)" systems shows some promise.

It is highly desirable that the above attributes of nonadaptive robust feedback systems be also reflected in the design of robust adaptive controllers as well. Thus, in our view, an adaptive control design must explicitly yield stability- and performance-robustness guarantees, not just stability (which has been the central focus of almost all adaptive control methodologies).

1.1. Brief Historical Perspective

Early approaches to adaptive control, such as the *model-reference adaptive control* (MRAC) method and its variants, were concerned with real-time parameter identification and simultaneous adjustment of the loop-gain. Representative references are [3, 4, 5, 6, 7]. In the classical MRAC method the emphasis was on proving global convergence to the uncertain real parameter, while using deterministic Lyapunov (hyperstability) arguments for inferring closed-loop stability. However, the assumptions required for stability and convergence (such as relative degree, positive-realness, knowledge of the high-frequency gain) did not include the presence of unmodeled dynamics, unmeasurable disturbances and sensor noise. Moreover, no explicit and quantifiable performance requirement was posed for the adaptive system; rather the "goodness" of the MRAC design was judged by the nature of the command-following error based upon simulations. It turned out that classical MRAC systems can become unstable in the presence of plant disturbances, sensor noise and high-frequency unmodeled dynamics [8]. In [9] "averaging analysis" was used to partially overcome the problems identified in [8]. Moreover the MRAC methodology was limited to single-input single-output (SISO) plants; attempts to extend the MRAC methodology to the multi-input multi-output (MIMO) case were extremely cumbersome. Because of these shortcomings, we shall not further address the MRAC methodology in the sequel.

Later, and the more recent, approaches to the adaptive control problem involved *multiple-model techniques* which, in principle, are also applicable to the MIMO case. The (large) real-parameter uncertainty set was subdivided into smaller parameter subsets; each parameter subset gives rise to a different "plant model set" with reduced real-parameter uncertainty. One then designed a set of control gains or dynamic compensators for each model set so that, if indeed the true parameter was "close" to a specific model, then a "satisfactory" performance was obtained.

In most of the multiple-model approaches, the identification of the most likely model is carried out by a "supervisor" which switches into the feedback loop different controllers, based primarily on deterministic concepts [10, 11, 12, 13, 14, 15, 16, 17, 18]. These proofs and results were presented for the case of SISO systems. The second approach relied upon stochastic designs that generated on-line *posterior probabilities* reflecting which of the models is more likely. In the latter approach the controllers could be designed either by classical LQG methods [19, 20, 21, 22, 23, 24, 25, 26] or by more sophisticated RMMAC methods [27, 28, 29, 2] which can deal with MIMO designs. A philosophically different, more direct approach, called *unfalsified control*, also merits attention [30, 31, 32, 33, 34, 35] and we shall briefly discuss it in the sequel. Finally, we do not discuss numerous approaches to adaptive control utilizing "intelligent" methods, such as neuro-fuzzy designs, since they are void of any analytical insights.

In all the proposed multiple-model adaptive methods the complexity of the adaptive feedback system directly depends on the number of models employed, N. By decreasing the size of the parametric subsets one obtains more models. Thus, all multiple model approaches must address the following:
(a) how to divide the initial large parameter uncertain set into N smaller parameter subsets,
(b) how to determine the "size" or "boundary" of each parameter subset, and
(c) how large should N be? Presumably the "larger" the N, the "better" the performance of the adaptive system should be.

Although not commonly considered as an "adaptive" design, the widely used engineering designs utilizing *gain-scheduling* can be viewed as multiple-model methods. Gain scheduling is actually used to control nonlinear systems, such as aircraft and jet engines, where an exogenous *measured* variable, such as dynamic pressure, defines a family of LTI models over the operating envelope [36, 37, 38, 39, 40, 41]. A set of control gains is defined for each LTI model. The measured dynamic pressure (or equivalent) is used to *interpolate* between the control gains.

The basic difference between the adaptive multiple-model approaches discussed above and the gain-scheduled methods is the fact that in adaptive multiple-model approaches there is no externally measured variable to accomplish the equivalent of gain-scheduling. Rather, the information of which model is most likely (and what controller to use) must be obtained from organic plant measurements.

1.2. The RMMAC Design Philosophy

In this paper we shall stress the use of "robust performance" requirements on the adaptive system implemented by one of the available multiple-model methods. We follow our recent research on *"Robust Multiple-Model Adaptive Control (RMMAC),"* [1, 42, 27, 28, 29, 2], which will be discussed and evaluated in much more detail in the sequel.

If we turn our attention to the non-adaptive literature there exists a well-documented design methodology, and associated MATLAB design software, for linear time-invariant multivariable plants (both SISO and MIMO) that addresses simultaneously both *robust-stability and robust-performance* in the presence of unmodeled dynamics and parametric uncertainty as well as unmeasurable plant disturbances and sensor noise. This methodology, pioneered by J.C. Doyle and his colleagues, is often called the *mixed-μ* design method [43, 44, 45, 46, 47, 48, 49, 50]. We assume that the reader is familiar with this robust design methodology and associated software. The mixed-μ design method incorporates the state-of-the-art in non-adaptive multivariable robust control synthesis and exploits the proper use of frequency-domain weights to quantify desired performance. Typically, using the mixed-μ design method, one finds that as the size of the parametric uncertainty is reduced the guaranteed desired performance, say superior disturbance-rejection, increases. Unfortunately, very little has been done in integrating the non-adaptive mixed-μ design methodology with that of robust adaptive control studies; even though it should be apparent that the mixed-μ design method should provide us guidance on the selection and number, N, of the models to be used in any multiple-model adaptive control scheme. Notable exceptions are [32, 51, 16, 27, 28, 29, 2].

We now summarize our design philosophy regarding adaptive control designs that employ multiple models. We assume that:

(1). Independent of the size of uncertainty for the plant real parameter(s), the plant *always contains unmodeled dynamics* whose size must be bounded *a priori* only in the frequency

domain. Therefore, the adaptive design must *explicitly* reflect these frequency-domain bounds on the unmodeled dynamics. The presence of unmodeled dynamics immediately brings into sharp focus the fact that we must use the state-of-the-art in nonadaptive robust control synthesis, i.e. mixed-μ synthesis [43, 44, 45, 46, 47, 50]; and associated MATLAB software [48, 49].

(2). The plant is subject to *unmeasurable plant disturbances* whose impact upon the chosen performance variables (error signals) must be minimized, i.e. we must have superior "disturbance-rejection". The modern trend is to use frequency-dependent weights to emphasize and define superior disturbance-rejection performance. This design objective can also be accommodated by the mixed-μ design methodology.

(3). The plant measurements are not perfect; thus *sensor measurements are corrupted by unmeasurable sensor noise*. The performance variables must be "insensitive", to the degree possible, to such sensor noise.

(4). *Performance requirements must be explicitly defined*, up to constants whose values can be optimized for superior performance. In the mixed-μ design methodology these "disturbance-rejection" performance requirements are explicitly quantified by frequency-domain weights typically involving the selected "error signals" and the control variables.

(5). Given the information in (1) to (4), we can design what we call the best *"global non-adaptive robust compensator (GNARC)"* for the entire (large) uncertain real-parameter set and by taking into account both the unmodeled dynamics and the performance requirements. This non-adaptive feedback design must be *optimized* so as to yield the best possible performance, i.e. superior disturbance-rejection with reasonable control effort. The GNARC design then provides a yardstick (lower-bound) for performance, so that any performance improvements by more complex adaptive designs can be quantified. The GNARC is designed using the mixed-μ synthesis methodology.

(6). An *upper-bound* for adaptive performance can be obtained by optimizing the performance under the assumption that the real-parameter values are known *exactly*, but still reflecting the presence of complex-valued unmodeled dynamics and frequency-dependent performance requirements. This implies that we compute for a large number of grid points in the original parameter uncertainty set what we call a *"fixed non-adaptive robust compensator (FNARC)"* which defines the *best possible* performance for each parameter value. The FNARC design is carried out using the complex-μ synthesis methodology, since we still must take into account the unmodeled dynamics and frequency-dependent performance specifications. We use the *same* quantitative performance requirements as in part (5) above. The set of the FNARCs corresponds to having an *infinite number of models* in the multiple-model implementation.

The difference between the lower-bound on performance defined by the GNARC in part (5) and the FNARC upper-bound from part (6) provides a valuable quantitative decision aid to the designer on *what performance improvements are possible by some multiple-model adaptive control method*. The designer must then make a *quantifiable choice on the degree of performance improvement* that he/she desires from the adaptive system. As we shall show in the sequel, this approach will then define the number of models (parameter subsets) required, N, and their numerical specification (boundary of parameter subsets) in a natural manner.

One possible approach is that the designer demands that the adaptive performance *equals or exceeds* a certain percentage, say 75%, of the (best possible) FNARC performance for each parameter value. Another possible approach is to demand that the adaptive system yield a performance that equals a certain multiple, say 10, of the (lower-bound) GNARC performance,

if possible (there may be inherent limitations due to non-minimum phase zeros and/or unstable poles) and consistent with the FNARC upper-bound. Yet another approach is to fix the number N of the models, i.e. specify the complexity of a multiple-model scheme, and maximize the performance for each model. Many other approaches are also possible which use both the GNARC lower-bound and the FNARC upper-bound for performance.

By following the above *performance-driven* methodology one directly arrives at the number, N, of required models in the adaptive multiple-model system, as well as the quantification of each model. The robust synthesis method requires that each "model" is represented by a parameter-subset which must be a hyperparallelpiped. In general, the more stringent the performance requirements on any adaptive implementation – consistent, of course, with the FNARC upper-bound – the larger the number of models and the greater the complexity of the multiple-model adaptive system. *We stress that such a systematic definition of the required models and numerical specification would not be possible if we did not explicitly pose the performance specifications and optimized performance to the extent possible.*

The procedure summarized above can be used with any of the adaptive multiple-model methods. We shall illustrate its detailed design and properties by using the multiple-model method in the context of dynamic hypothesis-testing, which involves generating the posterior probability for each model, the so-called Robust Multiple-Model Adaptive Control (RMMAC) architecture. However, it can also be used in conjunction with the "switching" supervisory controllers developed by Morse, Anderson, Hespanha and others, as well as by the *unfalsified-control* approaches (although in the unfalsified-control methods one does not assume any models for the plant to be controlled, no lower- and upper-bounds on the real uncertain parameters, no frequency-domain bounds on unmodeled dynamics nor explicit performance requirements in the sense of (4)-(6) above). The important point to remember is that all multiple-model adaptive schemes require the definition of the minimum number of models required to achieve both robust-stability and robust-performance, and these can only be defined *after* we pose realistic performance requirements for the adaptive system as discussed above.

In Section 2 we present an overview of four different multiple-model architectures for adaptive estimation, identification and control. In Section 3 we discuss the designs of the robust dynamic compensators, such as the GNARC and FNARC discussed above, and how to determine the number N of models in the multiple-model architectures, as well as the collection of the compensators. In Section 4 we focus upon the "identification" aspect of the RMMAC. In Section 5 we define a variant of the RMMAC architecture, denoted by RMMAC/XI. The RMMAC/XI was developed because of possible performance degradation of the standard RMMAC when the plant disturbances had very wide variability [2]. We shall demonstrate that such performance degradations are eliminated by the RMMAC/XI architecture. In Section 6 we present numerous results and simulations using the RMMAC and RMMAC/XI architectures. These results provide a concrete illustration of our design philosophy as well as several performance evaluations and comparisons of the RMMAC design, even when we violate the theoretical assumptions to a significant degree. Section 7 discusses the commonalties and differences of the supervisory *switching multiple model adaptive control (SMMAC)* and *RMMAC architectures* and we summarize our conclusions in Section 8.

2. MULTIPLE-MODEL ARCHITECTURES IN ADAPTIVE CONTROL

In this section we shall discuss the architectures that utilize multiple-models. We follow a historical development that traces the concepts over the past 40 years or so. First we shall discuss the architecture associated with *multiple-model adaptive estimation (MMAE).* Second, we discuss the early extension of the MMAE concepts to *classical multiple-model adaptive control (CMMAC).* Third, we present architectures associated with SMMAC. Next, we discuss the architecture associated with *robust multiple-model adaptive control (RMMAC). Finally, we comment briefly on the unfalsified control concept.*

2.1. Multiple-Model Adaptive Estimation (MMAE)

One of the earliest uses of multiple-models was motivated by the need for accurate stochastic state-estimation for dynamic systems subject to significant parameter uncertainty. In many such applications the estimation accuracy provided by standard Kalman filters was not adequate. For some early references regarding MMAE see [52, 53, 54, 55, 56]. We remark that the MMAE algorithms were also referred to as "partitioned estimation", especially in the research by Lainiotis. We also remark that a similar adaptive estimation architecture, the so-called "Sum of Gaussians" method, utilizing a bank of extended Kalman filters, was used for nonlinear filtering problems [57, 58, 59].

Fig. 1 shows the architecture of the MMAE system. It is assumed that a discrete-time linear time-invariant plant is driven by white process noise, as well as a known deterministic input signal, and generates measurements that are corrupted by white measurement noise. If there is no parameter uncertainty in the plant, then the Kalman filter (KF) is the optimal state-estimation algorithm in a well-defined sense; see, for example, [60, 57]. Moreover, under the usual linear-gaussian assumptions, the KF state-estimate is the true conditional mean of the state, given the past controls and observations. If the plant has an uncertain real-parameter vector, p, one can imagine that it is "close" to one of the elements of a *finite* discrete parameter

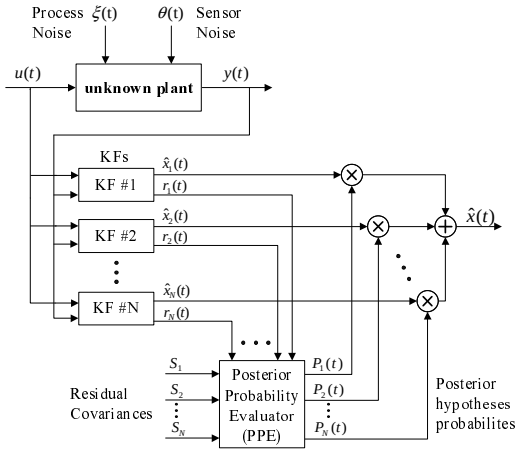


Figure 1. The MMAE architecture for adaptive estimation.

set, $P_D = \{p_1, p_2, \ldots, p_N\}$. One can then design a bank of standard Kalman filters (KFs) where each KF uses one of the discrete parameters p_k in its implementation, $k = 1, 2, \ldots, N$. It turns out that, if indeed the true plant parameter is identical to one of its discrete values – and this is modeled by the hypothesis $H = H_k$, then *the conditional probability density of the state is the sum of gaussian densities.* Then, the MMAE of Fig. 1 will indeed generate the *true conditional mean* of the state and one can calculate the *true conditional covariance matrix*; see, for example, [54]; [57, Chap. 10]. Appendix I summarizes the notation and formulas associated with the MMAE algorithm for the discrete-time case.

From a technical point of view the MMAE system of Fig. 1 blends optimal estimation concepts (i.e. Kalman filtering) and dynamic hypothesis-testing concepts that lead to a system identification algorithm. As explained in Appendix I, each KF generates a "local" state estimate, $\hat{x}_k(t|t)$ and a residual (or innovations) signal, $r_k(t)$, which is the difference between the actual measurement and the predicted measurement (the residual is precisely the *prediction error* common to all adaptive systems). Furthermore, the (steady-state) residual covariance matrix, $S_k,\ k = 1, 2, \ldots, N$, associated with each KF can be computed off-line. The key to the MMAE algorithm is the so-called *"posterior probability evaluator (PPE)"* which calculates, in real time, the posterior conditional probability that each model generates the data, i.e. $P_k(t) = Prob\{H = H_k|Y(t)\},\ k = 1, 2, \ldots, N$; see eq. (2.4). *Thus, the PPE represents an identification subsystem.* The "global" state-estimate is then obtained by the probabilistic weighting of the local state-estimates as shown in Fig. 1; this "global" state estimate is *precisely* the true conditional mean of the state given the set $Y(t)$ of past measurements and controls. The true conditional covariance can also be calculated on-line; see Appendix I for mathematical details.

The key property of the MMAE algorithm is that, under suitable assumptions, one of the posterior probabilities, say $P_j,\ P_j(t) \to 1$, where j indexes the model that is "closest" to the correct hypothesis $H = H_j$, even though the actual plant parameter is different than p_j, as $t \to \infty$. These asymptotic convergence results hinge upon information-theoretic arguments and involve non-trivial stationarity and ergodicity assumptions. The detailed convergence proofs involve either the so-called "Baram Proximity Measure (BPM)" [61, 62, 63], as discussed in Appendix II, or the Kullback information metric [57, pp. 267–279]. The detailed proofs are beyond the scope of this paper. These (asymptotic) convergence results to the "nearest probabilistic neighbor" using the BPM represent the key "system-identification" algorithms associated with both the CMMAC and the RMMAC algorithms discussed in the sequel.

It should be noted that the MMAE architecture is essentially identical to that of the "sum of Gaussians" estimators used extensively in nonlinear filtering [57, 58, 59] which utilize banks of extended Kalman filters. Furthermore, it is important to stress that the blend of dynamic hypothesis-testing concepts and optimal estimation theory is the workhorse of all modern defense and civilian surveillance and fusion algorithms that deal with several sensors and several targets (crossing, manoeuvring, disappearing, re-appearing, etc.)

2.2. Classical Multiple-Model Adaptive Control (CMMAC)

The intriguing convergence properties of the MMAE algorithm, coupled with the robustness shortcomings of MRAC systems to disturbances and sensor noise, gave rise to what we call the classical MMAC (CMMAC) algorithms which simply integrated design concepts from Linear-Quadratic-Gaussian (LQG) control system design [64, 65] with the MMAE architecture of

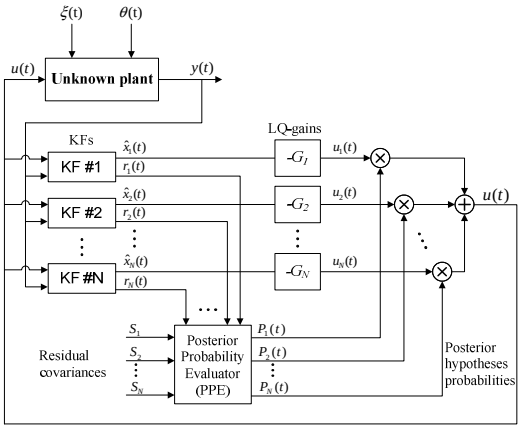


Figure 2. The CMMAC algorithm where the "local" KF state estimates are multiplied by "local" LQ gains to form "local" controls which, in turn, generate a probabilistically weighted "global" control.

Section 2.1 in a purely ad-hoc manner. The classical MMAC architecture (CMMAC) is shown in Fig. 2. Each "local" state-estimate of the (steady-state) KF, $\hat{x}_k(t|t),\ k = 1, 2, \ldots, N$, is multiplied by the associated linear-quadratic (LQ) optimal control-gain matrix, G_k, to generate a "local" control signal $u_k(t)$ by

$$u_k(t) = -G_k \hat{x}_k(t|t),\ k = 1, 2, \ldots, N \tag{2.1}$$

Using precisely the same MMAE calculation by the PPE of the posterior probabilities $P_k(t)$, eq. (2.4), the "global" control applied to the plant – and to the bank of KFs – is obtained by probabilistic weighting of the local controls, i.e.

$$u(t) = \sum_{k=1}^{N} P_k(t) u_k(t) \tag{2.2}$$

The CMMAC algorithm of Fig. 2 represented a purely *ad-hoc* approach to adaptive control. *It is not optimal* in a stochastic LQG sense [19]. However, several adaptations, extensions and simulations have been reported [20, 22, 21, 23, 24, 25, 26]. In the context of this paper it is important to stress that no robustness to unmodeled dynamics was considered in early CMMAC designs (such robustness issues were unknown in the 1970s) and that performance specifications were generated by LQG "tricks" and not with the frequency-weight concepts and $\mathcal{H}_2$ and $\mathcal{H}_\infty$ designs widely adopted at present.

From a high-level philosophical perspective, the CMMAC represents, in our view, a form of "probabilistic gain-scheduling." As we remarked earlier in classical gain-scheduling designs [10, 36, 37, 38, 39, 40, 41] an exogenous measured signal is used to "schedule" or "interpolate" from a finite-set of gains. In the CMMAC no such exogenous signal is available; rather, it is the posterior probabilities in Fig. 2 that are used to accomplish this "probabilistic gain-scheduling." As we shall see, the same situation happens in the RMMAC architecture(s) discussed in the sequel.

It is also noteworthy to mention that a "switching" version of the CMMAC can be readily (and naturally) implemented in which the "local" control with the largest posterior probability

is used as the "global" control; such switching versions of the CMMAC are much more sensitive to the stochastic signals. Finally, in retrospect, the fact that both the "local" KF state estimates and their residuals, via the posterior probabilities, were combined in the generation of the "global" adaptive control has certain shortcomings, because one mixes state-estimation and feedback control generation. Thus, errors in the local state-estimates will directly impact the local controls. In the CMMAC architecture no clear-cut "separation-principle" of an "identification subsystem" and a "control subsystem" is made.

2.3. Supervisory Switching Multiple-Model Adaptive Control (SMMAC)

During the past decade a different, with a strong deterministic flavour, approach to adaptive control using multiple models has been initiated [11, 12, 13, 14, 16, 17, 18] and active research along these lines is still in progress. We refer to these methodologies as *Supervisory Switching Multiple-Model Adaptive Control (SMMAC)*. The basic architecture of the different SMMAC algorithms is shown in Fig. 3 (which is an adaptation of Fig. 1 in [18], so that comparisons with the CMMAC and RMMAC become easier). We remark that in all versions of the SMMAC architectures there exists a "separation" between identification and control (unlike the CMMAC). In other words, the state-estimates inside each multi-estimator do not directly influence the associated local control signal.

We briefly discuss the SMMAC architecture to point out some similarities and differences to the CMMAC of Fig. 2 and the RMMAC architecture to be discussed below – see Fig. 4. The approach is deterministic and the goal is to prove "local" bounded-input bounded-output stability of the SMMAC system under certain assumptions. The plant-disturbance and sensor-noise signals are assumed bounded rather than being characterized as stochastic processes as in the CMMAC and RMMAC. The presence of unmodeled dynamics is also considered, although the bound on the unmodeled dynamics is simply an $\mathcal{H}_\infty$ bound. The theoretical results to date are restricted to single-input single-output (SISO) systems, although research is underway to extend them to the MIMO case [10].

In reference to Fig. 3, the SMMAC employs a finite number of stable deterministic estimators

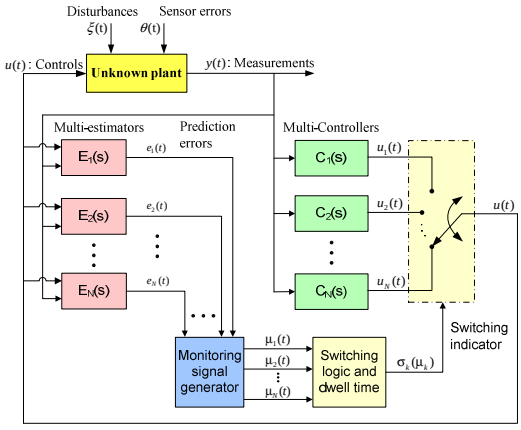


Figure 3. The SMMAC architecture.

(Luenberger observers), called *multi-estimators* and denoted by $E_k(s),\ k = 1, 2, \ldots, N$, designed for a grid of distinct known parameter values $p_k \in P$, where P is the compact real-parameter set. The output of each estimator $y_k(t)$ is compared with the measured (true plant) output $y(t)$ to form the estimation (prediction) errors $e_k(t) = y_k(t) - y(t),\ k = 1, 2, \ldots, N$. We remark that the errors $e_k(t)$ in Fig. 3 are completely analogous to the residuals $r_k(t)$ in the MMAE (Fig. 1), the CMMAC (Fig. 2) and the RMMAC (Fig. 4). The *monitoring signal generator*, $M(s)$, is a dynamical system that generates monitoring signals $\mu_k(t)$; these are suitably-defined integral norms of the estimation errors $e_k(t)$. The size of these monitoring signals indicates which of the multi-estimators is "closer" to the true plant. In addition to the bank of multi-estimators, it is assumed that a family of *multi-controllers*, $C_k(s),\ k = 1, 2, \ldots, N$, has been designed so that each provides *satisfactory stable feedback performance* for at least one discrete parameter $p_k \in P_D$. However, no quantitative design performance specifications are defined for the CMMAC system. The basic idea is to use the monitoring signals $\mu_k(t)$ to "switch-in" the suitable controller. This is accomplished – see Fig. 3 – by a switching logic S which generates a signal $\sigma(t) \in P_D$ that can be used to switch-in the appropriate controller. *A key property of the switching logic S is that it keeps its output $\sigma(t)$ constant over some suitably long "dwell time";* this avoids rapid switching of the controllers and allows most transients to die-out between controller switchings. The details of the switching logic algorithms differ in the cited SMMAC references.

The basic structural difference between the CMMAC and the SMMAC – compare Figs. 2 and 3 – is that in the SMMAC the "identification" process is completely separated from the "control" process. Even in the case of the "switching" CMMAC, where the largest posterior probability switches the corresponding LQ control gain, the identification and control get mixed-up. The separation of the identification and control processes in the SMMAC seems to have an advantage, coupled with the idea of infrequent controller-switching. Otherwise, the KFs in the CMMAC serve the same objective as the multi-estimators of the SMMAC. Moreover, the multi-controllers in the SMMAC can be more complex than the simple LQ-gains in the CMMAC. Some potential shortcomings of the SMMAC methodology will be discussed after we present the RMMAC approach below. Unfortunately, SMMAC numerical simulations have been reported for only a couple of (very) academic SISO plants.

2.4. Robust Multiple-Model Adaptive Control (RMMAC)

We now overview the newest multiple-model architecture which we call RMMAC, Fig. 4, to emphasize the fact that both stability-robustness and performance-robustness are addressed from the start. Our preliminary results on RMMAC can be found in [1, 42, 28, 29] and a more complete treatment is available in Fekri's Ph.D. thesis [2]. We note that the RMMAC architecture has a "separation" between identification and control, like the SMMAC and unlike the CMMAC.

As in the CMMAC, the RMMAC uses a bank of (steady-state) Kalman Filters (KFs) and relies on stochastic processes for the disturbance signals and the sensor noise measurements. The stochastic plant disturbances provide the necessary "sufficient excitation" for identification [66]. *However, unlike CMMAC the "local" KF state-estimates (the $\hat{x}_k(t|t)$ in Fig. 2) are not used in generating the control signals.* Only the KF residuals, $r_k(t),\ k = 1, 2, \ldots, N$, generated on-line and their pre-computed residual covariance matrices, S_k, are utilized by the posterior probability evaluator (PPE) to generate the posterior probability signals $P_k(t)$. The calculation

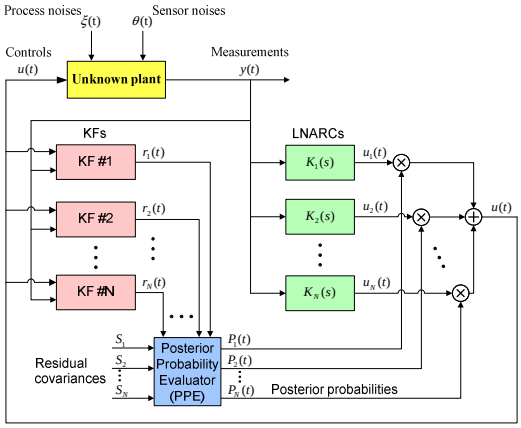


Figure 4. The RMMAC architecture.

of the posterior probabilities is identical to that in the MMAE or CMMAC (see Appendix I); see also eq. (2.4) below. A crucial difference is that the "nominal" KF design is based upon explicit use of the Baram Proximity Measure (BPM) to ensure asymptotic convergence of the posterior probabilities. Another key difference lies in the construction of the bank of the "local" robust compensators, $K_j(s),\ j = 1, 2, \ldots, N$, in Fig. 4, which are designed using the state-of-the-art in robust mixed-μ synthesis [43, 45, 44, 46, 49, 48]. These compensators, $K_j(s);\ j = 1, 2, \ldots, N$, referred to as the *"local non-adaptive robust compensators (LNARC)"*, are designed so as to guarantee "local" stability- and performance-robustness, as will be explained in detail in Section 3. Each "local" compensator $K_j(s)$ generates a "local" control signal $u_j(t)$. The "global" control is then generated, as in the CMMAC case, by the probabilistic weighting of the local controls $u_j(t)$ by the posterior probability $P_j(t)$

$$u(t) = \sum_{j=1}^{N} P_j(t) u_j(t) \tag{2.3}$$

A switching version of RMMAC can also be implemented by finding the largest probability $P_j(t)$, at each instant of time, and using the corresponding "local" $u_j(t)$ as the "global" control $u(t)$. Moreover, a "dwell-time" could be incorporated to avoid frequent switching.

One relies upon the convergence properties of the posterior probabilities to the nearest probabilistic neighbor, using the Baram proximity measure (BPM) [61, 62, 63], to ensure that the RMMAC operates in a superior manner, so as to ensure correct asymptotic identification. See Appendix II. This requires careful design of the KFs, perhaps robustified through the use of "fake-white-plant-noise" – a time-honoured design trick in linear and nonlinear estimation practice [67].

One of the key algorithms in the RMMAC architecture is the "Posterior Probability Evaluator (PPE)" in Fig. 4. Since we are concerned with both robust-stability and robust-performance – and the LNARCs $K_1(s), \ldots, K_N(s)$ are designed with this objective in mind – it is imperative that one of the posterior probabilities converges to the right model. We

repeat from Appendix I (eqs. (A.20) to (A.22)) the recursive formula by which the posterior probabilities $P_k(t)$, $k=1,2,\ldots,N$ are generated:

$$P_k(t+1) = \left[\frac{\beta_k \left(e^{-\frac{1}{2} r_k'(t+1) S_k^{-1} r_k(t+1)} \right)}{\sum_{j=1}^{N} \beta_j \left(e^{-\frac{1}{2} r_j'(t+1) S_j^{-1} r_j(t+1)} \right) \cdot P_j(t)} \right] P_k(t) \tag{2.4}$$

where:

$r_j(t)\,;\ j = 1, 2, ..., N$ is the residual of the j-th Kalman filter

$S_j\,;\ j = 1, 2, ..., N$ is the steady-state constant residual covariance matrix of $r_j(t)$

$\beta_j \equiv \frac{1}{(2\pi)^{m/2}\sqrt{\det\ S_j}}$ is a constant scaling factor.

Notice that the propagation of the posterior probabilities involves the quadratic forms $r_j'(t) S_j^{-1} r_j(t)$. If the residuals are, in a sense, "regular", i.e. their size and correlation is more-or-less consistent with that predicted by their associated covariance matrix (say within 3 to 4 sigma levels), then the posterior probabilities will behave in a smooth manner and will result in correct identification (convergence to the nearest probabilistic neighbour in the BPM sense. This is accomplished by the very careful design of the KFs as explained in Section 4. If the theoretical assumptions of Appendix II are valid or *mildly violated*, the RMMAC yields excellent performance, as evidenced by thousands of simulations [2]. If, however, the theoretical assumptions are *severely violated*, then the residuals can become very large compared with their predicted size inherent in their covariances S_j, the posterior probabilities can get "confused" and the inaccurate identification can cause the RMMAC to yield degraded performance and, in rare circumstances, break into instability. These issues will be discussed further in the sequel; it is important to state explicitly the potential shortcomings of any adaptive method (a rarity in the adaptive literature).

2.5. Unfalsified Control

As we have remarked the recent unfalsified control method developed by M.G. Safonov and colleagues [30, 31, 32, 33, 34, 35] is a promising approach to the control of uncertain plants. We stress that in that methodology there are no assumptions about the plant to be controlled; one applies control(s) and makes real-life measurements. It is assumed that there exists a collection of N precomputed compensators and that at least one of them will stabilize the actual plant. Thus, the emphasis is on stability (which is certainly robust since it involves the actual plant). It can be argued that the availability of N compensators makes unfalsified control a multiple-model scheme.

When one of the available compensators is connected to the plant and it does not stabilize the plant, then a *safe algorithm* should recognize rapidly in real-time this instability and discard this compensator. Next, another compensator is tried and so on. By assumption, eventually one will find one of the stabilizing compensators.

Since there are absolutely no assumptions on the plant, we cannot make any direct comparisons with the architectures discussed above. It is hard to say how one can design for this method the required family of compensators to achieve not only robust-stability but also robust-performance based upon explicit specification of closed-loop performance specifications.

For these reasons, we will not comment any further on this potentially useful methodology.

2.6. Discussion

The three adaptive multiple-model algorithms (CMMAC, SMMAC, and RMMAC) presented above have the following common characteristics.

(a). One must design a set of N multi-controllers or dynamic compensators.

(b). One must design a set of N Kalman filters or multi-estimators (observers).

(c). One must implement an identification process by which the actual (global) adaptive control is generated.

Clearly, the complexity of the adaptive system will depend on the number, N, of models that are required to implement. Ideally, N should be as small as possible. However, it should be intuitively obvious that if N is too small the performance of the adaptive system may not be very good. On the other hand, if N is very large, one may reach the point of diminishing returns as far as adaptive performance improvement is concerned. It follows that we need a more-or-less systematic procedure by which, starting from a compact parameter set $p \in P$, to define a finite set, N, of discrete values (models), $P_D = \{p_1, p_2, \ldots, p_N\} \subset P$, that are used subsequently in designing the KFs in the CMMAC and RMMAC, the multi-estimators in SMMAC as well as the compensators. The following section summarizes our suggested methodology which hinges upon the recent developments in robust feedback control synthesis using the so-called mixed-μ methodology and associated software.

Even though all MMAC architectures are made by piecing together LTI systems, the probabilistic weighting in the RMMAC as well as the supervisory switching logic of the SMMAC result in a *highly nonlinear and time-varying* closed loop MIMO feedback system. Hence, *it is naïve to expect foolproof global asymptotic stability results in the near future*, because there does not, as yet, exist a solid mathematical theory for global (stochastic) nonlinear time-varying stability which can be readily adapted to the multiple-model architectures discussed in this paper. Even in the simpler CMMAC, involving LQG controllers, attempts to prove global stability were not successful [20, 22, 23]. Thus, it is the opinion of the authors, what is needed in the short run is additional pragmatic understanding of the different multiple-model approaches, their similarities and differences and *consistent fair comparisons on performance improvement* over non-adaptive designs. Thus, there are numerous opportunities for future theoretical research to investigate such global stability-robustness and performance-robustness issues, especially in the MIMO case.

3. DESIGNING ROBUST COMPENSATORS IN THE RMMAC ARCHITECTURE

3.1. Introduction

During the past several years a very sophisticated and complete *non-adaptive* design methodology, accompanied by MATLAB design software, has been developed for the robust feedback control of MIMO linear time-invariant (LTI) uncertain dynamic systems with simultaneous dynamic and parametric errors. This design methodology is often called the "mixed-μ synthesis" method, which involves the so-called *D,G-K* iteration, and it requires the design of different $\mathcal{H}_\infty$ compensators at each iteration. The outcome of the "mixed-μ synthesis" process is the definition of a *non*-adaptive LTI MIMO dynamic compensator with fixed parameters, which guarantees that the closed-loop feedback system enjoys stability-robustness and performance-robustness, i.e. it meets the posed performance specifications in the frequency

domain (if such a compensator exists). The "mixed-μ synthesis", loosely speaking, de-tunes an optimal $\mathcal{H}_\infty$ nominal design, that meets more stringent performance specifications, to reflect the presence of inevitable dynamic and parameter errors. In particular, if the bounds on key parameter errors are large, then the "mixed-μ synthesis" yields a robust LTI design albeit with inferior performance guarantees as compared to the $\mathcal{H}_\infty$ nominal design. *So the price for stability-robustness is poorer performance.* Experience has shown that if the bounds on the parameter errors decrease, then the mixed-μ synthesis yields a design with better guaranteed performance. Two recent applications [68, 69] illustrate this uncertainty/performance tradeoff in a clear manner. We cannot provide details about this elegant methodology in this paper.

Whether we are dealing with non-adaptive or adaptive feedback designs we must take into account the following engineering issues:

(a). Complex-valued plant unmodeled dynamics (e.g. unmodeled time-delays, plant-order reduction, parasitic high-frequency poles and zeros, high-frequency bending and torsional modes, etc).

(b). Errors in key real-valued plant parameters in its state space realization.

(c). Explicit definition of performance requirements typically in the frequency-domain (rather than just the shape of step responses, location of dominant closed-loop poles etc); these reflect the common objective to have small tracking errors in the low-frequency region and small control signals in the high-frequency region.

(d). Unmeasurable plant disturbances, perhaps with information on their power spectral densities.

(e). Unmeasurable sensor noises, perhaps with information on their power spectral densities.

From now on, we focus our attention on the problem of "disturbance-rejection" in the presence of noisy sensor measurements so as to simplify the exposition. Adding "command-following" to the specifications is straightforward but complicates the exposition. What we want to stress relates to our philosophy that we cannot design adaptive control systems without explicit quantification of desired performance.

Assume that we have a state-space description of the plant (excluding unmodeled dynamics) of the form

$$\begin{aligned} \frac{d}{dt}x(t) &= A(p)x(t) + B(p)u(t) + L(p)d(t) \\ y(t) &= C_1(p)x(t) + D(p)n(t) \\ z(t) &= C(p)x(t) \end{aligned} \tag{3.1}$$

where $x(t)$ is the state vector, $u(t)$ the control vector, $d(t)$ the plant-disturbance vector, $y(t)$ the (noisy) measurement vector, $n(t)$ the sensor noise and $z(t)$ the performance (output or error) vector, i.e. the vector for which we wish to minimize the effects of the disturbance $d(t)$ and noise $n(t)$, i.e. have superior disturbance-rejection. The system matrices depend upon a real-valued parameter vector p, where p is constrained to be in a (hyper) parallelepiped, $p \in P_\pi$; this is a required μ-synthesis constraint. Thus, we must have a lower- and an upper-bound (real-valued) for each independent uncertain parameter. The disturbances, $d(t)$, can be either deterministic time-functions or stochastic processes. In either case, the robust μ-synthesis requires a disturbance frequency weight for superior performance. If *d(t)* is a stochastic process, its power-spectral density naturally defines this weight.

From a performance point of view, in order to achieve superior disturbance-rejection, the designer specifies a frequency weight on $z(t)$. Typically, to achieve superior disturbance-

rejection in the low-frequency region, the designer specifies, in the μ-synthesis methodology, a frequency-weight, say of the form

$$\bar{z}(s) = A_p \left(\frac{\alpha}{s+\alpha} \right)^m \cdot I \cdot z(s) \tag{3.2}$$

which implies that superior disturbance-rejection is most important in the frequency range $0 \leq \omega \leq \alpha$. *The larger the performance-gain parameter A_p, the better the desired performance-rejection.*

To complete the robust design synthesis the designer must provide frequency-dependent bounds for all (structured or unstructured) unmodeled dynamics, and frequency weights for the control, disturbance and sensor noise vectors. The control and sensor-noise weights, together with the bound(s) on the unmodeled dynamics, safeguard against very high-bandwidth feedback designs.

3.2. The GNARC Design

Before a wise designer can make a decision on whether or not to implement an adaptive system, he/she must have a solid knowledge on *what is the best robust non-adaptive design.* We called the best "global non-adaptive robust compensator" GNARC-based design. The GNARC is computed via mixed-μ synthesis and it takes into account the frequency-domain bounds on unmodeled dynamics, the various frequency weights that quantify disturbance-rejection requirements, control effort and, perhaps, power spectral densities for the plant-disturbances and sensor-noises.

We certainly do not intend to provide here a tutorial exposition of the mixed-μ synthesis method. To compute the GNARC, one fixes the performance gain A_p in eq. (3.2) to some initial value and exercises the MATLAB software which, after a sequence of the so-called *DG-K* iterations, determines a compensator $K(s)$ and generates an upper-bound $\mu_{ub}(\omega)$. If

$$\mu_{ub}(\omega) < 1 \ \ \forall \omega \tag{3.3}$$

then the resulting feedback design is guaranteed to be stable for all "legal" unmodeled dynamics and the entire parameter uncertainty $p \in P_\pi$. Moreover, *in addition to stability-robustness, we are guaranteed that we meet or exceed the posed performance requirements.* Therefor, in order to find the "best" GNARC we must *maximize the performance-parameter A_p* in eq. (3.2) *until the μ-upper bound is just below unity*, say $0.995 \leq \mu_{ub}(\omega) \leq 1 \ \forall \omega$.

The GNARC is a single dynamic (SISO or MIMO) compensator that can be used by the designer to fully understand what is the best possible robust performance in the absence of adaptation. Since the feedback system is LTI a whole variety of performance evaluations are possible, using *representative* values of the uncertain real-parameter vector $p \in P_\pi$ for the plant, as summarized in Table I.

We believe that such a thorough understanding of the non-adaptive GNARC is essential prior to making a design decision on using some sort of multiple-model adaptive control. It has been our experience that such GNARC analyses point out which parameter uncertainty is most critical; this can be used to eliminate from further consideration non-critical parameters. Moreover, the performance of the GNARC, say for a SISO system, can change drastically if one adds more measurements and/or controls. Thus, an unacceptable performance for a SISO GNARC non-adaptive system may, indeed, become acceptable if more controls and/or sensors

Table I. List of Performance Evaluation Tools

(1) Magnitude (or singular-value) Bode plot of the closed-loop transfer function from the plant disturbance, d, to output, z, which measures the quality of disturbance-rejection vs frequency. Assuming that the plant has no integrators, the magnitude of this transfer function, in the low frequency region, will be approximately $1/A_p$. *This is why we must maximize the performance parameter A_p for superior disturbance-rejection.*
(2) Magnitude (or singular-value) Bode plot of the closed-loop transfer function from the sensor noise, n, to output, z, which measures the quality of insensitivity to sensor noise vs frequency
(3) Magnitude (or singular-value) Bode plot of the closed-loop transfer function from the plant disturbance, d, to control, u, which measures the impact of the plant-disturbance on the control vs frequency
(4) Magnitude (or singular-value) Bode plot of the closed-loop transfer function from the sensor noise, n, to control, u, which measures the impact of the sensor noise on the control vs frequency
(5) Root-mean-square (RMS) tables assuming that the plant-disturbance, d, and the sensor noise, n, are stationary stochastic processes. Such RMS values are readily evaluated via the solution of Lyapunov equations. Individual or combined RMS tables for the output, z, and the control, u, as a function of the plant-disturbance, d, and the sensor noise, n, can be computed.
(6) Time-domain responses, e.g. step- or sinusoidal-disturbance, stochastic signals, etc for different values of the uncertain constant parameters *and for slowly time-varying parameters* (within their predefined ranges).

are introduced [2] and a MIMO GNARC analyzed, thereby eliminating the need for complex adaptive control.

3.3. The FNARC Design

If the GNARC analyses discussed above indicate the need for adaptive control, they provide a *lower-bound* upon robust performance. The *"fixed non-adaptive robust compensators (FNARC)"* provide the means for *quantifying an upper-bound on robust performance.* "Ideally", the FNARC analyses assumes an infinite number of models, $N \to \infty$, in any multiple-model adaptive scheme. Thus, we understand *what is the best possible performance if we knew the real parameter(s) exactly.*

In practice, to determine the FNARC one uses a dense grid of parameters $p_j,\ j \to \infty$, in P_π and determines the associated robust compensator for each p_j using exactly the same bounds on unmodeled dynamics and frequency weights employed in the GNARC design. Thus, we can make fair and meaningful comparisons. For each p_j we use the *complex-μ* design methodology and MATLAB software [49], because both the performance weights and bounds on unmodeled dynamics are complex-valued and there are no real parameter uncertainties. For each p_j, we again maximize the performance-parameter A_p in (3.2) until the complex-μ upper-bound, $\mu^c_{ub}(\omega)$ is just below unity for all frequencies, say $\mu^c_{ub}(\omega) \approx 0.995 \quad \forall\omega$, to be consistent with the GNARC lower-bound.

One can then again analyze each FNARC design using the six techniques outlined in Table I.

Our experience indicates that detailed analyses, especially at the corners and faces of the hyper-parallelepiped P_π, provide useful insights regarding the impact of the subset of the uncertain real parameters upon performance; this determines the need for sophisticated adaptive control.

3.4. The Potential Benefit of Adaptive Control

Recall that we had posed the following question: *do we need adaptive control?* The GNARC and FNARC results provide the designer with the means to answer this question.

In Fig. 5 we visualize a hypothetical plot of the outcome of the GNARC and FNARC designs by plotting the (maximized) performance parameter A_p, see (3.2), as a function of a scalar uncertain real parameter, p, $p_L \leq p \leq p_U$. We denote the (constant) value associated with the GNARC design by A_p^G. We denote the parameter-dependent value associated with the FNARC by $A_p^F(p)$, $p_L \leq p \leq p_U$. The difference, $A_p^F(p) - A_p^G \geq 0$ quantifies the impact of the uncertain parameter $p \in [p_L, p_U]$ upon performance. The FNARC process indicates that if the parameter p were *known exactly*, then the (low-frequency) disturbance-rejection is approximately $1/A_p^F(p)$. The GNARC process indicates that if the parameter p were *unknown*, then the (low-frequency) disturbance-rejection is approximately $1/A_p^G$. *Note that to obtain the FNARC benefits we must implement a multiple-model architecture with an infinite number of models.* In this manner we have quantified the potential performance benefits of adaptive control; the non-adaptive (single-model) GNARC provides the lower-bound upon expected performance, while the adaptive (infinite-model) FNARC provides the performance upper-bound. This information is critical in deciding whether to implement an adaptive control system or not.

The shapes of the curves in Fig. 5 provide additional valuable information. In the hypothetical case of Fig. 5, we should expect the benefit of using adaptive control to be greatest if the true parameter was near its upper-bound, i.e. $p \approx p_U$. The benefits decrease if the unknown parameter is closer to its lower-bound, i.e. $p \approx p_L$. Indeed it may well happen that $A_p^F(p_L) = A_p^G$. This can occur, for example, if the parameter p, in rad/sec, represents the value of a non-minimum phase zero which places inherent restrictions upon disturbance-rejection (see, e.g. [70]). Such a non-minimum phase system has been analyzed, using the RMMAC, in [27].

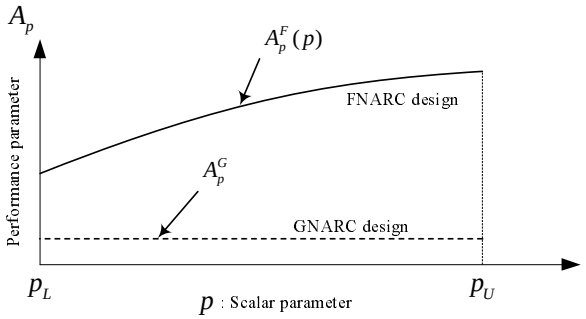


Figure 5. Hypothetical comparison of the performance parameter, A_p, for the GNARC and FNARC for the case of a scalar uncertain real parameter, p, $p_L \leq p \leq p_U$.

If we have an uncertain m-dimensional parameter vector, constrained in a hyper-parallelepiped, i.e. $p \in P_\pi \subset R^m$, the calculations are more complex to construct the two hypersurfaces along the lines suggested by Fig. 5 [2]. However, the same philosophy still applies. In [42] a two-input two-output RMMAC system with two uncertain real parameters is designed and analyzed.

3.5. Determining the Number of Models and Designing the LNARCs

Let us suppose that we have decided that there is a substantial benefit in using adaptive control and that we wish to use a multiple-model architecture. As we remarked before, the adaptive complexity is directly related to the number N of models in either the SMMAC or RMMAC implementation. Recall that the non-adaptive GNARC requires $N = 1$ model while the FNARC requires $N = \infty$ models. Clearly, there must be a happy medium.

3.5.1. A "Brute-Force" Approach. A "brute-force" approach is to decide on the number of models, say N=4, and their parameter variation as illustrated in the hypothetical visualization of Fig. 6.

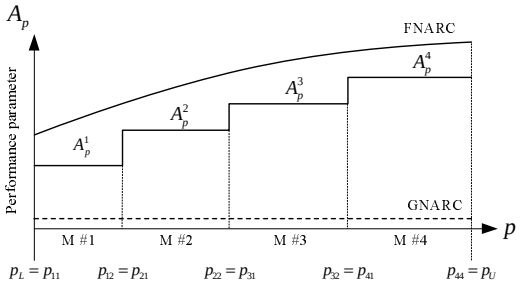


Figure 6. Illustration of a "brute-force" selection of four models. An *ad-hoc* decision is made on the number of models, N=4, and the specification of their "boundaries", $[p_{kL}, p_{kU}]$, $k = 1, 2, 3, 4$, for each model M#k. The A_p^k denote the maximized value of the performance parameter in eq. (3.2).

Essentially, in this approach, the designer fixes the "adaptive complexity", quantified by N, of the multiple-model system. Each model, denoted by M#k (k=1,2,3,4) requires definition of its "local" lower-bound, p_{kL}, $k = 1, 2, 3, 4$, and upper-bound, p_{kU}, $k = 1, 2, 3, 4$, i.e.

$$\begin{aligned}
&\text{Model } \#1\,(M\#1): \quad p_L = p_{1L} \le p \le p_{1U}\\
&\text{Model } \#2\,(M\#2): \quad p_{1U} = p_{2L} \le p \le p_{2U}\\
&\text{Model } \#3\,(M\#3): \quad p_{2U} = p_{3L} \le p \le p_{3U}\\
&\text{Model } \#4\,(M\#4): \quad p_{3U} = p_{4L} \le p \le p_{4U} = p_U
\end{aligned}$$

The model upper- and lower-bounds must be found by trial-and-error in this brute-force method.

After the models are selected, all frequency-dependent bounds and weights are fixed as in the GNARC and FNARC designs. Next, for each model the performance parameter A_p – see (3.2) – denoted now by A_p^k, k=1,2,3,4, is maximized using the mixed-μ software, in an iterative mode, for the smaller parameter uncertainty subset associated with each model. Thus, for each model, we are again attempting to attain as large a disturbance-rejection as possible. Moreover,

at the end of this iterative optimization process, the software also generates what we call the *"local non-adaptive robust compensator (LNARC)"* which we denote by $K_j(s),\ j = 1, 2, 3, 4$.

The values of the A_p^k, k=1,2,3,4, as illustrated in Fig. 6 can be used as a "guide" for adjusting the boundaries of each model or changing the number of models. Each of the resulting LNARC designs can be evaluated in more detail by following the suggestions in Table I.

The RMMAC numerical results presented in [27, 29] followed such an *ad-hoc* approach for determining the models and the associated LNARCs. The same process can be used to design the multi-controllers in the SMMAC.

3.5.2. A More Systematic Approach to Model Selection and Definition: The % FNARC method. The numerical simulations for the RMMAC design in Section 6 utilize a much more systematic approach to the determination of the number of models and their numerical specification. This method fully exploits the information provided by the FNARC curve in Fig. 5.

We have remarked that the number of models required for any adaptive multiple-model design should be the natural outcome of performance design specifications. In the so-called % FNARC approach the designer specifies that the performance parameter, A_p, should be *equal or greater* than X% of the best possible performance as defined by the FNARC.

The basic idea is illustrated in Fig. 7, starting from the GNARC and FNARC curves of Fig. 5. Using the designer-specified value of X%, we construct the X% FNARC, shown in Fig. 7, and we proceed as follows. Since the FNARC is maximum at $p = p_U$, we slowly increase, starting from the upper limit, the size of the parameter uncertainty set $\Omega_\alpha = \{p : \alpha \le p \le p_U\}$. For each value of α, we use the mixed-μ software to design the best robust controller by maximizing the performance parameter A_p denoted by $A_{p\,\max}^{\alpha}$. As long as $A_{p\,\max}^{\alpha} > (X\%) \cdot A_p^F(\alpha)$, where $A_p^F(\alpha)$ is the parameter value of the FNARC at $p = \alpha$, then we decrease α until at $\alpha = \alpha^*$ we have $A_{p\,\max}^{\alpha^*} = (X\%) \cdot A_p^F(\alpha^*)$. The outcome of this process defines the dashed curve labeled Γ_1 in Fig. 7. The point α^* is at the intersection of the Γ_1 curve with the X% FNARC curve. This defines Model #1 (M#1) with uncertainty set $\Omega_1 = \{p : \alpha^* \le p \le p_U\}$; we also remark that the μ-software determines in addition the LNARC #1 controller denoted by $K_1(s)$.

The process is repeated from the (right) boundary of M#1, α^*. Starting at $p = \alpha^*$, we define the set $\Omega_\beta = \{p : \beta \le p \le \alpha^*\}$. For each value of β, we use the mixed-μ software to

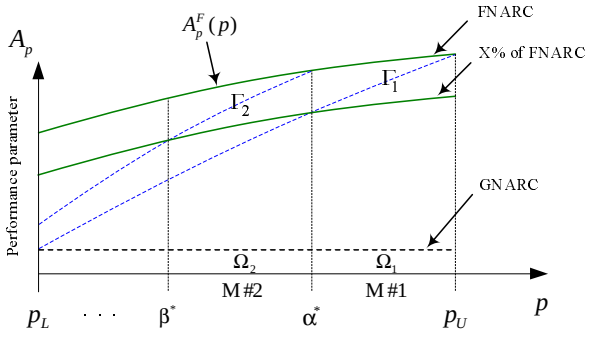


Figure 7. Visualization of the % FNARC model definition process.

design the best robust controller by maximizing the performance parameter A_p denoted by $A^{\beta}_{p\max}$. As long as $A^{\beta}_{p\max} > (X\%) \cdot A^F_p(\beta)$, where $A^F_p(\beta)$ is the parameter value of the FNARC at $p = \beta$, then β is decreased until at $\beta = \beta^*$ we have $A^{\beta^*}_{p\max} = (X\%) \cdot A^F_p(\beta^*)$. The outcome of this process defines the dashed curve labeled Γ_2 in Fig. 7. The point β^* is at the intersection of the Γ_2 curve with the X% FNARC curve. This defines Model #2 (M#2) with uncertainty set $\Omega_2 = \{p : \beta^* \leq p \leq \alpha^*\}$. It is noted that the μ-software also determines the LNARC #2 controller denoted by $K_2(s)$. The process is repeated until the parameter lower-bound is reached.

The % FNARC method is straightforward for systems involving a single *scalar* uncertain real parameter. In the case of two, or more, uncertain parameters the procedure has to be modified. For the case of two or more uncertain parameters, the GNARC, FNARC and % FNARC become surfaces. The above process yields intersection of surfaces (the equivalent of $\Gamma_1, \Gamma_2, \ldots$ are also surfaces). Unfortunately, the intersection of these surfaces *does not* occur along rectangular (or parallelepiped) parameter subsets. However, recall that the mixed-μ software requires rectangular (or parallelepiped) constraints for uncertain parameter sets. Suggested modifications to the % FNARC concept are discussed in [2].

3.6. Discussion

In this section we presented an overview of what we believe is the proper way of designing compensators or multi-controllers for the RMMAC and SMMAC architectures. We are driven by the desire that we must guarantee (local) robust-stability *and* robust-performance. This implies that we must exploit the state-of-the-art of the mixed-μ synthesis methodology and software. We also stressed the value of having optimized performance lower-bounds (via the GNARC) and upper-bounds (via the FNARC) to aid the control system designer in the selection of the models, their number and their numerical specification (and hence complexity of the adaptive system). We presented two methods (there are more) for defining the models and the associated compensators (the LNARCs) driven by designer-specified performance or complexity specifications.

4. DESIGNING THE BANK OF KALMAN FILTERS FOR THE RMMAC ARCHITECTURE

4.1. Introduction

In this section we discuss the critical issues related to the design of the Kalman filters (KFs) in the RMMAC architecture. We remark that the design of the bank of KFs is much more systematic (and complex) that the *ad-hoc* multi-estimators employed in SMMAC architectures.

The proper design of each KF in the RMMAC architecture of Fig. 4 is *crucial* in order to satisfy the theoretical assumptions [61, 62, 63] which will imply that the PPE will yield the correct model identification. The KF design explicitly utilizes the Baram proximity measure (BPM) Appendix II presents the summary concepts leading to the on-line generation of the posterior probabilities and contains the key equations for calculating the Baram proximity measure (BPM).

The design of the KFs is done *after* the number of models and their boundaries have been established using the procedures in Section 3.5. Recall that the original parameter set, $p \in P_\pi$ (a

parallelepiped) is subdivided into N subsets (parallelepipeds) denoted by Ω_k, $k = 1, 2, \ldots, N$, such that

$$\bigcup_{k=1}^{N} \Omega_k = P_\pi \tag{4.1}$$

and each Ω_k defines the associated Model #k. We need to design a discrete-time steady-state KF for each Ω_k. To accomplish this we need to specify the "nominal" value of the parameter, denoted by $p_k^* \in \Omega_k$, which will be used to design the k^{th} Kalman filter. Each KF requires specification of the intensity matrices of the plant white noise and of the sensor white noise.

The "naïve" point of view would be to choose p_k^* at the center of Ω_k. Unfortunately, this may lead to unpredictable behavior in the convergence of the posterior probabilities. The correct way is to select the p_k^* using the Baram proximity measure (BPM).

The presence of unmodeled dynamics (and their frequency-domain bounds) is neglected in the design of the KFs. Unfortunately, we are not aware of any theory that is currently available to help us design KFs that are "robust" in the presence of unmodeled dynamics. Thus, to robustify the KFs we often use "fake plant white noise". This engineering trick [67] is extremely common in engineering estimation applications. It causes the KF gains to increase and pay more attention to the measurements, which after all contain the actual information about the uncertain plant.

4.2. The Baram Proximity Measure (BPM)

Let $p \in \Omega$ and let p^* denote the nominal value used to implement a KF. If $p = p^*$, then the steady-state KF residual, $r^*(t)$, would be a stationary white-noise sequence with a specific covariance matrix S^*. If the data were generated by a different LTI system, with $p \neq p^*$, then the KF residual $r(t)$ would no longer be white. The BPM is a real-valued function, denoted by $L(p, p^*)$ which measures how large is the "stochastic distance" between the residuals $r(t)$ and $r^*(t)$. See Appendix II for details.

Now suppose that $p \in \Omega$, as above, but we have designed two different KFs, one (KF #1) with nominal value $p_1^* \in \Omega$ and another (KF #2) with $p_2^* \in \Omega$. Now we can calculate two BPMs, $L_1 \equiv L(p, p_1^*)$ and $L_2 \equiv L(p, p_2^*)$.

Fig. 8 illustrates this idea for a scalar parameter, p, where we visualize the BPMs L_1 and L_2 for all $p \in \Omega$. For the specific value $p = p_A$ shown, we can see that $L(p_A, p_1^*) < L(p_A, p_2^*)$. The implication of this is that for a 2-model MMAE, when the true value is $p = p_A$, the posterior probabilities $P_1(t) \to 1$, $P_2(t) \to 0$, so that we have convergence to Model #1 (defined by KF #1), even though the Euclidean distances would indicate the opposite ($|p_A - p_1^*| > |p_A - p_2^*|$).

Fundamental convergence result: In [61] this result was generalized and proved for an arbitrary number of stable KFs designed for the nominal values $p_1^*, p_2^*, \ldots, p_N^*$. If the BPM satisfies the inequality

$$L(p, p_j^*) < L(p, p_k^*) \;\; \forall k \neq j = 1, 2, \ldots, N \tag{4.2}$$

then, under some additional stationarity and ergodicity assumptions – see Appendix II – the posterior probabilities converge *almost surely* to the correct model, i.e.

$$\lim_{t \to \infty} P_j(t) \to 1 \tag{4.3}$$

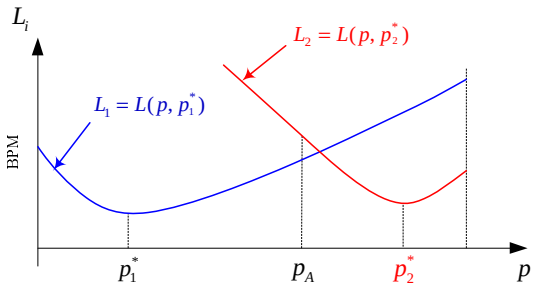


Figure 8. Illustration of Baram proximity measures (BPM).

4.3. Integrating Probability Convergence Results for the RMMAC

The outcome of the controller design of Section 3 yielded the number of models, N, and their boundaries, in terms of the parallelepipeds $\Omega_1, \Omega_2, \ldots, \Omega_N$ which define the models $M\#1$, $M\#2$, $\ldots$, $M\#N$. The question now is: *how do we select the nominal values* $p_1^*, p_2^*, \ldots, p_N^*$ *to design the KFs?*

The idea, illustrated for three models in Fig. 9 for a scalar parameter, is to use an iterative algorithm to calculate the nominal KF values p_1^*, p_2^*, p_3^*, *so that the BPMs are equal at the boundary of adjacent* Ω_s.

In this manner, the fundamental probability convergence result will guarantee that $p \in \Omega_j \Rightarrow P_j(t) \to 1$ *a.s.* This, is the method we use in the numerical RMMAC simulations of Section 6.

This method becomes more complicated when we have two, or more, uncertain parameters. It becomes necessary to use some sort of genetic algorithm to determine the optimized nominal KF design points so that the BPM surfaces agree at the model boundaries [42, 2].

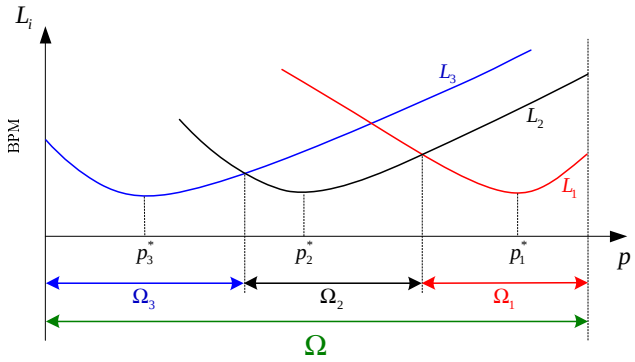


Figure 9. Optimizing the KF nominal design points using the BPMs so that they are equal at the boundaries of adjacent models.

4.4. Discussion

We have presented a brief summary on how to optimize the design of the KFs in the RMMAC architecture using the BPM, so as to ensure "correct model identification" by the posterior probabilities. We stress that a KF is optimal if indeed the data is generated by the same model as that used to design the KF. In adaptive control, however, the true parameter is NOT identical to that of the "closest" KF in the sense of Section 4.3. Also, the unmodeled dynamics are not taken into account in the design of the KF. Moreover, the KFs and the PPE must perform well even when some of the stochastic assumptions are violated. Thus, we may need to "robustify" the KF to be tolerant of such "errors". The engineering practice of using suitable "fake plant white noise" is a very useful tool. We have found [42, 2] that judicious use of "fake plant white noise" (which causes the KF gains to increase and pay more attention to the measurements [67]) can be a very valuable tool in improving the quality and speed of probability convergence. We believe that future fundamental studies of robustifying KFs, in the context of multiple-model adaptive control, are very relevant. Also, the identification must work when the parameters change "slowly" as a function of time.

It is also important to note that similar probability convergence results for the MMAE can be found in [57, pp. 267–279] and cited references therein) which used the so-called Kullback information metric. The Kullback metric, however, is not a true norm (it violates the triangle inequality). Nonetheless, it would be highly desirable that future research examines in-depth the relationship of the BPM and the Kullback metric, especially with respect to posterior probability convergence behavior.

5. THE RMMAC/XI ARCHITECTURE

The simulations in Section 6 will demonstrate that the RMMAC works quite well even if we violate the theoretical assumptions for MMAE convergence in a "mild" manner. This fact is supported by hundreds of different simulations in Fekri's Ph.D. thesis [2]. Nonetheless, we must always remember that: *theories have limitations, stupidity does not!*

Obviously, the superior performance of the RMMAC hinges upon the rapid convergence of the posterior probabilities to the correct model (see eq. 2.4 and Appendices I and II). As we remarked in Section 2.5 such a convergence critically depends upon the "regularity" of the residuals generated by the bank of Kalman filters (KFs) in Fig. 4. Each KF is designed for a particular covariance matrix $cov[\xi(t);\xi(\tau)] = E\{\xi(t)\xi'(\tau)\} = \Xi\delta(t-\tau)$ of the continuous-time zero-mean plant white noise $\xi(t)$, where Ξ is the plant-noise intensity matrix. The numerical value of the intensity noise Ξ will determine the size of the KF gains and the residual covariance matrices S_k. These will determine the size of the colored stochastic disturbances applied to the plant.

Under normal operating conditions these exogenous plant disturbances will fall within a specific range. It may very well happen that once-in-a-while the actual plant disturbances become *very much larger* than the normal ones*. During these "abnormal" time-intervals, the

*Flight load-alleviation automatic control systems are often designed for different levels of turbulence and the pilot can "switch-in" the appropriate control system. However, in case of extreme turbulence (not anticipated

actual KF residuals will be much larger than those predicted by their associated "normal" residual covariances, S_k. Our experience has been that during these "abnormal" time intervals [2], the posterior probabilities generated by eq. (2.4) can undergo rapid oscillations and frequent switching among the models and, as a consequence, the RMMAC performance deteriorates; often it is worse than that of the nonadaptive GNARC design. A numerical example in Section 6 will provide concrete evidence of this phenomenon.

In short, the RMMAC can behave poorly when the true stochastic plant disturbances, as quantified by the intensity matrix Ξ_{act} of the actual plant white noise, are *very different* from those associated with the intensity Ξ_d used to design the linear KFs. If $\Xi_{act} \gg \Xi_d$, then the actual residuals are much higher than, say, the 3-sigma values expected by the residual covariance matrices S_k – see eq. (2.4) – and this can lead to rapid switching of the posterior probabilities, "confused identification" and poor RMMAC performance.

If, on the other hand, $\Xi_{act} \ll \Xi_d$, then the actual residuals are much lower than, say, the 3-sigma values expected by the residual covariance matrices S_k and this can also lead to rapid switching of the posterior probabilities, or slow convergence, "confused identification" and poor RMMAC performance. In this case, the smaller actual plant disturbance may not indeed provide sufficient *persistent excitation*, required by the identifiability conditions, to generate signals that are not masked by the measurement noise.

These pragmatic considerations led us to develop the so-called RMMAC/XI architecture illustrated in Fig. 10. By increasing the complexity of the RMMAC we can mitigate the RMMAC performance deterioration problems. The basic idea in RMMAC/XI is to introduce additional models, i.e. increase the number of hypotheses, to reflect different ranges of the plant noise intensity matrix Ξ. Let us assume that the number, N, of models and their size has been determined by the required adaptive performance specifications and that we have already calculated the N LNARCs denoted by $K_k(s), \quad k = 1, 2, \ldots, N$. Let us further suppose that the plant noise intensity Ξ is bounded by

$\Xi \in [\Xi_L, \Xi_H]$

and that, for the sake of simplicity, we decide to select two intermediate values, Ξ_1 and Ξ_2, such that

$\Xi_L < \Xi_1 < \Xi_2 < \Xi_H$

Then we design two sets of linear KFs, one set indexed by $k = 1, 2, \ldots, N$ using the noise intensity Ξ_1 and the second set indexed by $k = N+1, N+2, \ldots, 2N$ using the noise intensity Ξ_2. It is important to stress that the nominal points for designing each KF, as determined by the BPM method of Section 4 will be *different* (and must be recomputed) for each value of Ξ_1 and Ξ_2. Essentially, we have doubled the number of hypotheses in the associated MMAE; but we can still apply the MMAE-based identification methodology. This allows us to define the RMMAC/XI architecture shown in Fig. 10.

Note that each of the $2N$ KFs will now generate a *different* residual, $r_k(t)$, $k = 1, 2, \ldots, 2N$. Also, we pre-compute the $2N$ residual covariance matrices S_k. These are introduced to the PPE of Fig. 10 which will generate $2N$ posterior probabilities, $P_k(t)$, $k = 1, 2, \ldots, 2N$, one for each of the $2N$ hypotheses. Note, however, that the bank, and number, of the N LNARCs is *unchanged*, since their design does *not* depend on Ξ; the LNARCs are only influenced by the

in the design phase) the pilot typically disconnects the control system and flies the aircraft himself.

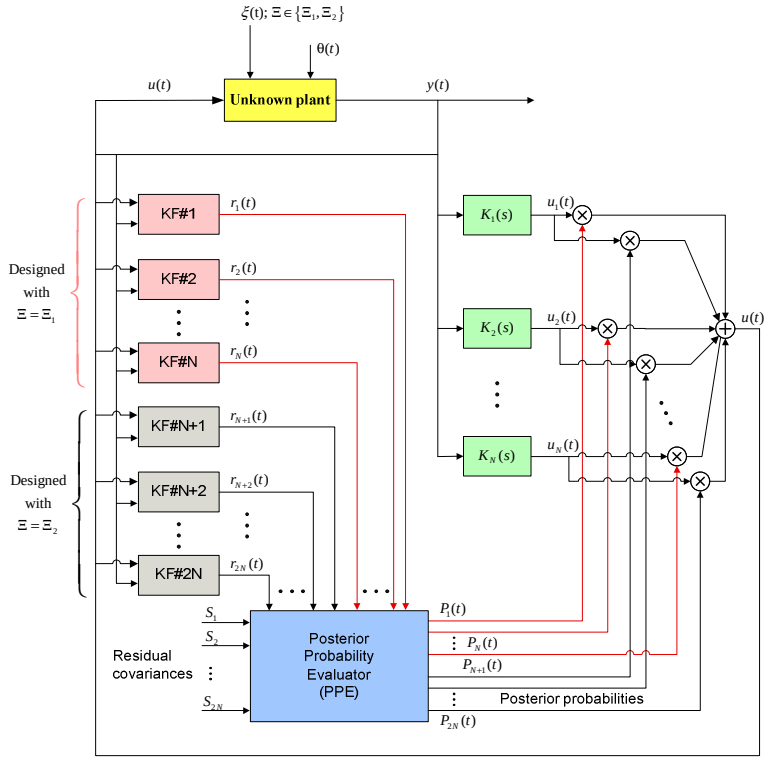


Figure 10. The RMMAC/XI architecture.

disturbance frequency weight that is independent of Ξ. Thus, it is important to note in Fig. 10 that each LNARC control signal $u_k(t)$, $k = 1, 2, \ldots, N$, is multiplied by the *two* different associated posterior probabilities, $P_k(t)$ and $P_{k+N}(t)$, to generate the overall adaptive control $u(t)$.

It is obvious how to generalize this idea for more than two values of Ξ. It remains an open research question on how to select "intelligently" the values of Ξ, other than by brute-force and trial-and-error.

The advantages of implementing the RMMAC/XI architecture will be clearly demonstrated in the numerical simulations of Section 6. There we will compare the performance of the standard RMMAC with that of the RMMAC/XI during "abnormal" disturbance conditions.

6. RMMAC SIMULATIONS

We described above how the RMMAC architecture combines the state-of-the-art in mixed-μ robust synthesis and multiple model adaptive estimation (MMAE) system identification. Furthermore, we described the step-by-step design process required to implement a RMMAC design.

In this section we test and evaluate the disturbance-rejection performance of the RMMAC

feedback system as compared to the "best" GNARC non-adaptive design. We also illustrate, in a concrete way, the step-by-step design methodology of Section 3,4, and 5.

We present the outcome of several Monte Carlo (MC) simulations, which demonstrate the RMMAC performance improvements, as compared to that of the best non-adaptive GNARC system. Overall, for the example considered, the proposed RMMAC design yields significant performance improvements over the non-adaptive GNARC, thereby confirming the positive aspects of this adaptive control method.

We present simulation results when we satisfy or "mildly violate" the theoretical assumptions for both a constant as well as a slowly time-varying uncertain scalar parameter. We also discuss what happens when we "severely violate" the theoretical assumptions and compare the performance of the standard RMMAC and RMMAC/XI in "abnormal" disturbance environments.

6.1. The Two-Cart Example Dynamics

The RMMAC was tested and evaluated using the two-cart mass-spring-damper (MSD) system, shown in Fig. 11. A different topology of the two-cart system was analysed by the RMMAC in [29].

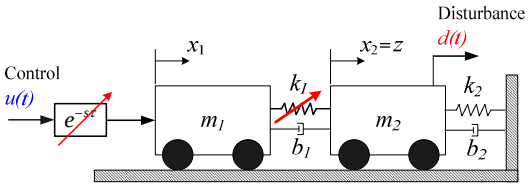


Figure 11. The two-cart MSD system. The spring-constant k_1 is uncertain. Also, the time-delay, τ, is uncertain; it represents unmodeled dynamics.

The system in Fig. 11 includes a random colored disturbance force, $d(t)$, acting on mass m_2 and sensor noise on the only measurement of the position of mass m_2. The control force $u(t)$ acts upon the mass m_1. The disturbance force $d(t)$ is a stationary first-order (colored) stochastic process generated by driving a low-pass filter, $W_d(s)$, with continuous-time white noise $\xi(t)$, with *zero mean* and *unit intensity*, i.e $\Xi = 1$, as follows:

$$d(s) = \underbrace{\frac{\alpha}{s+\alpha}}_{W_d(s)} \xi(s) \tag{6.1}$$

The overall state-space representation, including the disturbance dynamics via the state variable $x_5(t)$, is:

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) + L\xi(t) \\ y(t) &= Cx(t) + \theta(t) \end{aligned} \tag{6.2}$$

where the state vector is

$$x^T(t) = [x_1(t)\ x_2(t)\ \dot{x}_1(t)\ \dot{x}_2(t)\ d(t)]$$

and

$$A = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ -\frac{k_1}{m_1} & \frac{k_1}{m_1} & -\frac{b_1}{m_1} & \frac{b_1}{m_1} & 0 \\ \frac{k_1}{m_2} & -\frac{(k_1+k_2)}{m_2} & \frac{b_1}{m_2} & -\frac{(b_1+b_2)}{m_2} & \frac{1}{m_2} \\ 0 & 0 & 0 & 0 & -\alpha \end{bmatrix}$$

$$B^T = \begin{bmatrix} 0 & 0 & \frac{1}{m_1} & 0 & 0 \end{bmatrix}; C = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

$$L^T = \begin{bmatrix} 0 & 0 & 0 & 0 & \alpha \end{bmatrix}$$

The following parameters in (6.2) are fixed and known:

$$m_1 = m_2 = 1,\ k_2 = 0.15,\ b_1 = b_2 = 0.1,\ \alpha = 0.1 \tag{6.3}$$

The upper and lower-bounds for the *uncertain spring constant*, k_1, are:

$$\Omega = \{k_1 : 0.25 \leq k_1 \leq 1.75\} \tag{6.4}$$

The performance variable (output) $z(t)$ is the position of mass m_2,

$$z(t) \equiv x_2(t) \tag{6.5}$$

All feedback loops utilize a single measurement $y(t)$, that includes additive white sensor noise $\theta(t)$, independent of $\xi(t)$ defined by

$$\begin{aligned} &y(t) \equiv x_2(t) + \theta(t) \\ &E\{\theta(t)\} = 0,\ E\{\theta(t)\theta(\tau)\} = 10^{-6}\,\delta(t-\tau) \end{aligned} \tag{6.6}$$

The desired disturbance-rejection requires that the effects of $d(t)$ and $\theta(t)$ be minimized at the output, so that $z(t) \approx 0$.

Remark: The control problem is hard even if the spring, k_1, is known. Clearly, the control problem becomes much harder in our adaptive design, because the control $u(t)$ is applied via the uncertain spring, so we are not sure how much force is exerted through the uncertain spring to the mass m_2. Thus, we have a *non-collocated actuator problem* because the control is not applied to the mass m_2 whose position we wish to regulate.

In addition to the uncertain spring stiffness, we assume that there is, in the control channel, an *unmodeled* time-delay τ whose maximum possible value is 0.05 sec, i.e.

$$\tau \leq 0.05 \text{ sec} \tag{6.7}$$

The frequency-domain upper-bound for the unmodeled time-delay, which serves, for this example, as a surrogate for unmodeled dynamics, is required by the mixed-μ synthesis design and is the magnitude of the first order transfer function

$$W_{unmod}(s) = \frac{2.1s}{s+40} \tag{6.8}$$

6.2. Designing the Global Non-Adaptive Robust Compensator (GNARC)

In this section we discuss the details behind the mixed-μ design of the GNARC system, which guarantees the "best" robust-stability and robust-performance for the entire large spring uncertainty of (6.4). As explained in Section 3, the GNARC system will define what we can best expect in the absence of adaptation.

As it is common in $\mathcal{H}_2$ or $\mathcal{H}_\infty$ designs, we use the following frequency domain weights on the control and measurement noise,

$$\begin{aligned} &\text{Control weight: } W_u(s) = \frac{10(s+10)}{s+10^3} \\ &\text{Measurement noise weight: } W_n = 10^{-3} \text{ (constant)} \end{aligned} \tag{6.9}$$

These, together with the unmodeled dynamics weight (6.8), limit the bandwidth of the closed-loop system by penalizing large high-frequency control signals. The control weight $W_u(s)$ allows large controls at lower frequencies, where we desired superior disturbance-rejection, while penalizing controls at higher frequencies. The frequency weights (6.8) and (6.9) are incorporated in the definition of the *nominal generalized plant*, together with (6.2) with $k_1 = 1.0$. The weights (6.1), (6.8), and (6.9) will *not* change in any of the subsequent designs.

To carry out the mixed-μ synthesis, the parameter uncertainty of (6.4) is represented by

$$0.25 \leq k_1 \leq 1.75 \;\Rightarrow\; k_1 = 1.0 + 0.75\,\delta_{k_1};\;\; |\delta_{k_1}| \leq 1 \tag{6.10}$$

In order to design the "best possible" non-adaptive feedback system the following type of performance weight upon the output $z(t)$ is used.

$$W_p(s) = A_p \frac{0.1}{s+0.1} \tag{6.11}$$

which reflects our specification for good disturbance-rejection for the frequency range $\omega \leq 0.1$ rad/sec where the disturbance $d(t)$ has most of its power; see eq. (6.1). Notice that the performance weight $W_p(s)$ penalizes the output error in the same frequency range as the disturbance dynamics $W_d(s)$, while the gain parameter A_p in $W_p(s)$ specifies our desired level of disturbance-rejection. *The larger A_p, the greater the penalty on the effect of the disturbances on the position.* As we described in Section 3, for superior disturbance-rejection A_p should be as large as possible; how large it can be is only limited by the required guarantees on robust-stability and -performance inherent in the mixed-μ synthesis methodology.

Fig. 12 shows the MSD plant with weights as required by mixed-μ synthesis. One can note that there are two frequency-weighted "errors" $\tilde{z}(t)$ and $\tilde{u}(t)$. This figure is in fact a block diagram of the uncertain closed-loop MSD system illustrating the disturbance-rejection performance objective namely the closed-loop transfer function from $\xi(t) \rightarrow z(t)$, or $d(t) \rightarrow z(t)$.

The "position error" $\tilde{z}$ is our main performance variable for evaluating the quality of the disturbance-rejection. Since

$$\tilde{z}(s) = W_p(s)\, z(s) = A_p \left(\frac{0.1}{s+0.1} \right) z(s) \tag{6.12}$$

we communicate to the μ-design that position errors are most important below the "corner frequency" 0.1 rad/sec. *The larger the "performance parameter" A_p, the more one cares about position errors at all frequencies.*

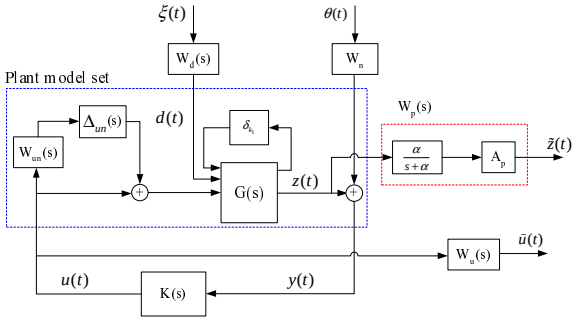


Figure 12. The MSD system with weights for mixed-μ synthesis.

The "control error" $\tilde{u}(t)$ is defined by

$$\tilde{u}(s) = W_u(s)\, u(s) = 10 \left(\frac{s+10}{s+10^3} \right) u(s) \tag{6.13}$$

Thus, $W_u(s)$ is a high-pass filter that penalizes the system for using large controls at high frequencies.

Using the mixed-μ software [48] the performance parameter A_p in (6.11) is increased, as much as possible, until the upper-bound on the mixed-μ, $\mu_{ub}(\omega)$, satisfies the inequality

$$\mu_{ub}(\omega) \leq 1, \ \forall \omega \tag{6.14}$$

which is only a *sufficient condition* for both stability- and performance-robustness. The largest value of the performance parameter A_p in (6.11) was to be

$$A_p^G = 50.75 \ \ (\text{with } \mu_{ub} \approx 0.995) \tag{6.15}$$

which leads to the GNARC design, i.e. the "best" LTI non-adaptive compensator $K(s)$ that guarantees stability- and performance-robustness for the entire parameter uncertainty interval (6.4). As explained in Section 3, the performance characteristics of the GNARC are to be used as the comparison-basis for evaluating performance improvement (if any) of our proposed RMMAC design. See Section 3.2.

6.3. Designing the Local Non-Adaptive Robust Compensators (LNARCs)

Following the procedure presented in Section 3, the plots of the maximized performance parameter A_p for the GNARC design and the FNARC designs (requiring an infinite number of models) is shown in Fig. 13. This figure shows that there is a potential 20-fold improvement in performance by using adaptive control.

Remark: In Fig. 13, the GNARC and FNARC look flat for all values of k_1. Actually, the FNARC becomes a little smaller as we approach small values of k_1, but this can not be noticed in the figure. This "flatness" disappears if we change our control specifications [2].

Next, we specify a desired level of adaptive performance to be 70% of the FNARC, i.e. $X = 70\%$, following the discussion of Section 3.5.2.

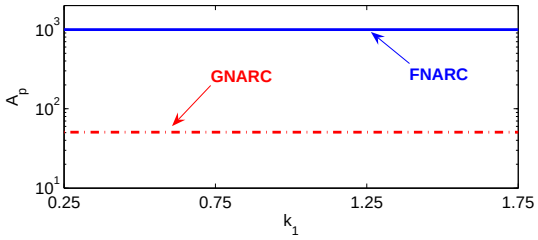


Figure 13. Best GNARC and FNARC performance gains for spring-stiffness uncertainty interval in (6.4); note the *log* scale along the vertical axis.

Fig. 14 shows how four models (subintervals) are determined to be used for the RMMAC based upon the required performance level of $X = 70\%$.

Their characteristics are summarized in Table II. Explanation. We now provide a bit more detail on how the four models of Fig. 14 are obtained. Starting at the North-East corner labeled $F_1, k_1 = 1.75$, (because the FNARC is maximum there) we slowly open-up the uncertain interval denoted by $\Omega_b = \{k_1 : b \leq k_1 \leq 1.75\}$. For each interval, we iterate (using the mixed-μ software [48]) to calculate the maximum performance parameter A_p^b which results in the dashed curve, labeled Γ_1 in Fig. 14. When the curve Γ_1 intersects the 70% FNARC curve we stop. In this example, this occurs at $k_1 = 1.02$ which yields the subset $\Omega_1 = [1.02, 1.75]$, i.e. $1.02 \leq k_1 \leq 1.75$, Model M#1, and LNARC #1. The left boundary of M#1 defines the point labeled F_2 on the FNARC. Starting from this point we slowly open up the interval denoted by $\Omega_c = \{k_1 : c \leq k_1 \leq 1.02\}$. Once more, we iterate using the mixed-μ software to calculate the maximum performance parameter A_p^c which results in the dashed curve, labeled Γ_2 in Fig. 14. When the curve Γ_2 intersects the 70% FNARC curve we stop. In this example, this occurs at $k_1 = 0.64$ which yields the subset $\Omega_2 = [0.64, 1.02]$, i.e. $0.64 \leq k_1 \leq 1.02$, Model #2, and LNARC #2. Repeating this process leads to the curves labeled Γ_3 and Γ_4 in Fig. 14 and the four models summarized in Table II.

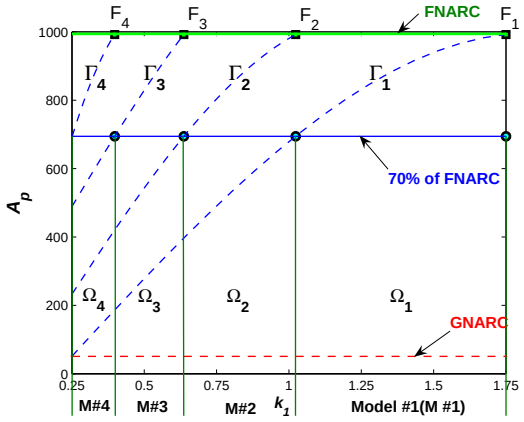


Figure 14. Construction of the four models using the performance requirement X=70% of the FNARC.

Table II. Summary of models and performance parameters

Compensator	Ω	X%	A_p^*
GNARC	$\Omega = [0.25, 1.75]$	5.1%	A_p^G=50.75
LNARC #1	$\Omega_1 = [1.02, 1.75]$	70%	A_p^1=694.5
LNARC #2	$\Omega_2 = [0.64, 1.02]$	70%	A_p^2=694.5
LNARC #3	$\Omega_3 = [0.40, 0.64]$	70%	A_p^3=694.5
LNARC #4	$\Omega_4 = [0.25, 0.40]$	70%	A_p^4=694.5

* Best performance gains for the GNARC and each of the four LNARCs used in subsequent designs vs "*FNARC*"

As a result, the entire initial uncertainty set of (6.4) is covered by four models requiring four local compensators. Clearly, the reduction in parameter uncertainty allows larger performance gains for designing the LNARCs, resulting into guaranteed *both* stability- and -performance robustness over the corresponding subintervals of Table II. Note that we should expect, on the average, about 20 times better performance from the RMMAC as compared to the GNARC (694.5/50.75)

The above model selection procedure also generates four "local" robust compensators, $K_1(s)$, $\ldots$, $K_4(s)$, designed for each subinterval (model) defined in Table II; these are referred to as LNARCs. In the mixed-μ synthesis, the weights (6.1), (6.8) and (6.9) were the same as in the GNARC design of Section 6.2. However, for each LNARC design, the performance parameter A_p in (6.11) is maximized until the mixed-μ bound of (6.14) is achieved, and their optimized values are shown in the last column of Table II.

Fig. 15 compares the GNARC compensator $K(s)$ with the four LNARCs, namely $K_1(s), \ldots,$ $K_4(s)$ by examining their Bode magnitude plots; it is clear that at low frequencies the LNARCs generate a loop-gain about 20 times as large compared to the GNARC and this, naturally, leads to the performance improvements in disturbance-rejection discussed below, especially in the specified frequency region $\omega < 0.1$ rad/sec. We emphasize that each individual LNARC closed-loop design has guaranteed performance- and stability-robustness over its associated parameter subinterval of Table II. These also confirm the "rough" predictions implied by Table II. These also confirm the "rough" predictions implied by Table II.

Remark: In the design philosophy adopted in this paper we have stressed that the adaptive

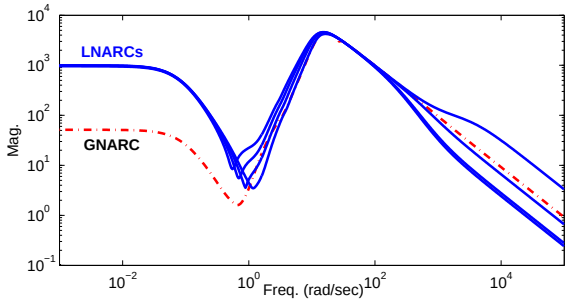


Figure 15. Compensator frequency-domain characteristics (Bode plot) of the GNARC and the four LNARCs.

controller complexity, as measured by the number of models in the RMMAC should be the natural by-product of the performance requirements. We have just demonstrated that if we demand that the performance equals or exceeds X=70% of the FNARC, we require the four models summarized in Table II.

If we are willing to accept a somewhat inferior performance and select, say, X=50% then the procedure outlined yields only two models. If we wish to have much better performance and select, say, X=90%, then the outlined procedure yields nine models. Therefore, as we demand better and better performance we must increase the RMMAC complexity, and this agrees with engineering intuition. If explicit performance requirements were used in the design of the SMMAC, the same philosophy could be applied for those architectures as well.

6.4. Predicting Potential RMMAC Performance Benefits

Testing the RMMAC requires significant computation using multiple Monte Carlo (MC) runs under different scenarios. It is highly desirable, as explained in Section 3, to use the LTI feedback designs, using the GNARC and LNARCs, to quantify the *potential benefits* of using adaptive control in general, and the RMMAC in particular. *From a pragmatic engineering perspective we must have tradeoffs that contrast the performance improvements (if any) of the very sophisticated RMMAC vis-a-vis the much simpler non-adaptive GNARC design.* To the best of our knowledge, such performance tradeoffs have not been explicitly quantified in other adaptive control studies.

Referring to Fig. 4, the RMMAC requires the on-line computation of its four Kalman filters (KFs) as well as of its four dynamic LNARCs, $K_1(s)$, ..., $K_4(s)$, in addition to the calculation of the four posterior probabilities, $P_1(s)$, ..., $P_4(s)$, by the posterior probability evaluator (PPE) – a lot of computations!

In order to understand how one can easily predict the potential RMMAC performance characteristics, assume that one of the posterior probabilities converges to its nearest probabilistic neighbor (which it does, as we demonstrate in the sequel); it follows that, after a transient time, a specific LNARC is used. *After the probability convergence, the RMMAC essentially operates as an LTI stochastic feedback system*!

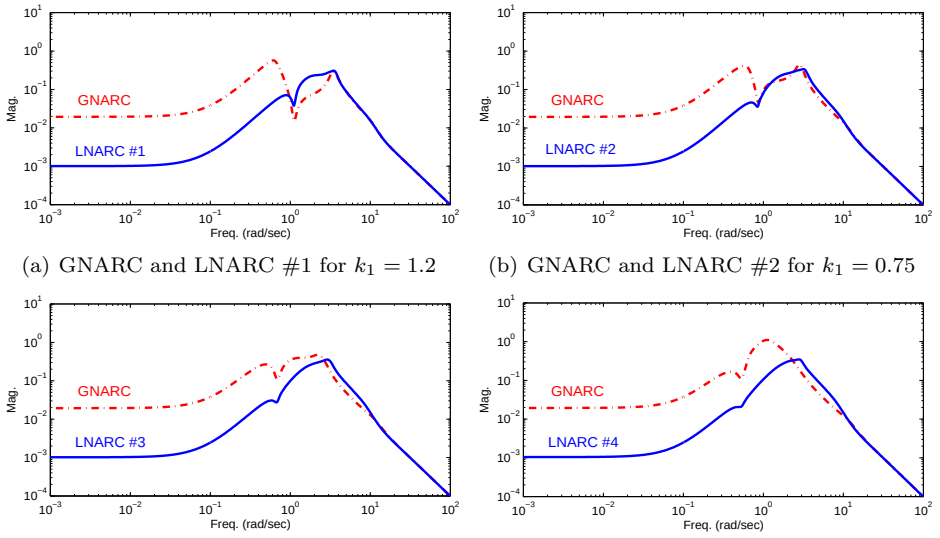
In the spirit of Table I, this allows us to calculate two key transfer functions for disturbance-rejection and control signal characteristics

$$\begin{array}{ll} \text{Disturbance - rejection:} & \\ & M_{\xi z}(s) \equiv \frac{z(s)}{\xi(s)} \quad \text{or} \quad M_{dz}(s) \equiv \frac{z(s)}{d(s)} \\ \text{Control - signal:} & \\ & M_{\xi u}(s) \equiv \frac{u(s)}{\xi(s)} \quad \text{or} \quad M_{du}(s) \equiv \frac{u(s)}{d(s)} \end{array} \tag{6.16}$$

for different values of the uncertain spring stiffness of (6.4), for the GNARC and for each LNARC design.

Fig. 16 illustrates the above using the different spring constants quantifying the potential RMMAC improvement in disturbance-rejection in the frequency domain. Fig. 16 predicts that the RMMAC has the potential to significantly improve disturbance-rejection on the order of $\frac{1}{A_p}$. These predictions will be again validated in the sequel.

Similar plots for control signal characteristics can be made for other values of the uncertain spring constant. Other transfer functions could also be computed (not shown) from the plant disturbance ξ(t) and the sensor noise θ(t) to the control $u(t)$.



(a) GNARC and LNARC #1 for $k_1 = 1.2$ (b) GNARC and LNARC #2 for $k_1 = 0.75$

(c) GNARC and LNARC #3 for $k_1 = 0.47$ (d) GNARC and LNARC #4 for $k_1 = 0.30$

Figure 16. Potential improvement from RMMAC visualized by Bode plots of the disturbance-rejection transfer function $|M_{\xi z}(j\omega)|$ of eq. (6.16).

Fig. 17 evaluates the potential performance improvement of using RMMAC by using stochastic metrics, namely by comparing the steady-state RMS errors of the output z and the control u, for different values of k_1. Assuming that ξ and θ are indeed white noises, these RMS results are readily computed by solving standard covariance algebraic Lyapunov equations for stochastic LTI systems, e.g. [60]. The graphs of Fig. 17 vividly suggest that RMMAC has the potential of decreasing the output RMS by a factor of 2–5 over that obtained from the GNARC system. Note that the potential improvement in output RMS requires controls with higher RMS values, as expected. Note also that, since we are using μ-synthesis, the GNARC, FNARCs and LNARCs we design are $\mathcal{H}_\infty$ compensators. These $\mathcal{H}_\infty$ compensators do not necessarily minimize RMS errors, unlike $\mathcal{H}_2$ controllers. Nonetheless, RMS metrics are important in engineering analyses and complement the frequency-domain plots of Fig. 16.

Finally, it is important to construct what we call the "Mismatch Model/LNARC" table shown in Table III.

The interpretation of Table III answers the question: *what happens to closed-loop stability if we use the LNARC #$K_j(s)$ when the true spring constant is in subinterval (or model) #i?* The diagonal entries in this table are always robustly-stable, by construction. Examining the first row in Table III it is observed that for Model #1, i.e. for all $k_1 \in [1.20, 1.75]$, if we use the LNARC #4 we always have instability (U); if we use the LNARC #2 we have instability for smaller values of k_1, but have stability for larger values (CU). This is due to the fact that the mixed-μ upper-bound inequality (6.14) is only a sufficient condition for both robust-stability and robust-performance and, hence, each LNARC design will actually have a wider robust-stability region. It turns out that for this example, LNARC #1 maintains stability for all $k_1 \in [0.69, 1.75]$, LNARC #2 for all $k_1 \in [0.43, 1.58]$, LNARC #3 for all $k_1 \in [0.25, 1.01]$, and

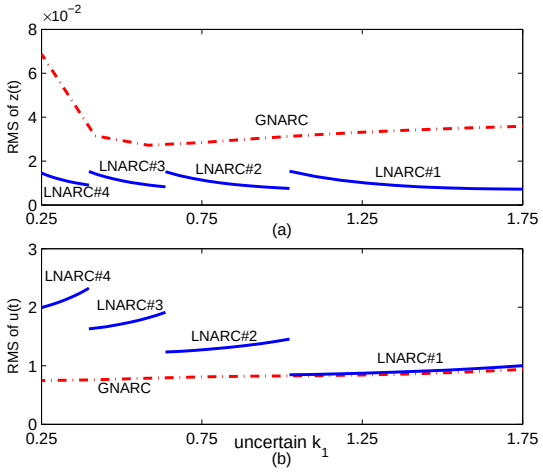


Figure 17. Predicted potential RMS performance of the RMMAC vs GNARC. We assume zero sensor noise for these plots. (a): Output RMS comparisons from $\xi(t)$ to $z(t)$. (b): Control RMS comparisons from $\xi(t)$ to $u(t)$.

Table III. Mismatched Model/LNARC Stability

Model #	LNARC #1	#2	#3	#4
1	S	CU	U	U
2	CU	S	CU	U
3	U	CU	S	S
4	U	CU	S	S

Legend: S ≡ always stable
U ≡ always unstable
CU ≡ conditionally unstable

LNARC #4 for all $k_1 \in [0.25, 0.73]$. Of course, performance-robustness is only guaranteed for the models defined in Table II.

6.5. Designing the MMAE and RMMAC Identification Subsystem

We now follow the process described in Section 4 to design the four Kalman filters (KFs) outlined in the RMMAC architecture of Fig. 4.

As explained in Section 4, a great deal of care must be exercised in designing the Kalman filters (KFs) in the RMMAC architecture, since the convergence of the appropriate posterior probability to its nearest probabilistic neighbor is at the heart of the RMMAC identification process (see Appendices I and II). The basic decision in designing the KFs in Fig. 4 is how to select the nominal value of the uncertain spring stiffness, k_1, denoted by $k_{1i}^*,\ i = 1, 2, 3, 4$, which must be used for designing each of the four KFs. *We stress that these nominal values are not at the centers of the sets Ω_j defined in Table II.* Rather, as explained in Section 4, the KF design points, $k_{1i}^*,\ i = 1, 2, 3, 4$, must be determined by an iterative optimization process

so that the Baram proximity measures (BPMs) agree at the boundaries of adjacent sets Ω_j. The outcome of this optimization process is shown in Fig. 18 using the optimized KF nominal values.

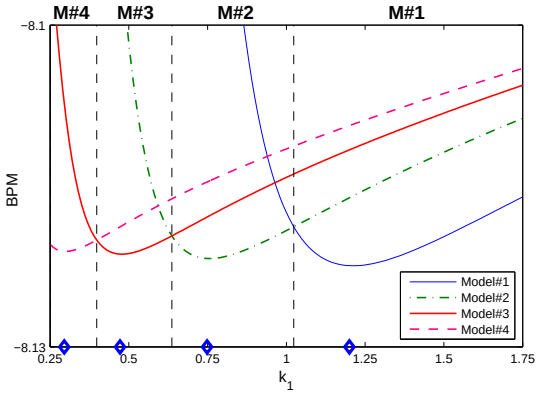


Figure 18. Baram proximity measures (BPM) for the optimized four KFs. Note that the BPMs are the same at the boundaries of adjacent model pairs.

In Fig. 18 the curves show the BPM, denoted by L_i, for any value $0.25 \leq k_1 \leq 1.75$ for each of the four optimized KFs. This process is required so that for any $k_1 \in \Omega_j$, the corresponding posterior probability $P_j \to 1,\ P_k \to 0\ \forall k \neq j$. The specific nominal KF numerical values for this example were obtained by iteration and are

$$\mathcal{K}_1^* = [1.2\ \ 0.75\ \ 0.47\ \ 0.30] \tag{6.17}$$

where $k_{1i}^* = \mathcal{K}_1^*[i]$ is the nominal spring constant k_1 used in Model $\#i$ associated with KF $\#i$. In Fig. 18 the numerical values (6.17) are shown as "diamonds" on the k_1 axis.

Although the GNARC and LNARC compensators are designed in continuous time, all simulations for this example were implemented in *discrete-time* using a zero-order hold with a sampling time of $T_s = 0.01$ secs. The KFs in the MMAE algorithm were designed in discrete-time using the sampling interval T_s.; all continuous-time dynamics (plant, frequency weights, etc) were transformed to their discrete-time equivalents. In addition, the correct covariance intensities of the discrete-time white noise sequences, $\xi(.)$ and $\theta(.)$, defined in (6.1) and (6.6), were calculated [60] and used to design the four steady-state discrete-time KFs as well as the posterior probability evaluator (PPE) in Fig. 4; these discrete-time numerical results were used in all Monte Carlo (MC) simulations that are presented in the sequel.

As explained in Section 2.4, the real-time KF residual sequences in Fig. 4, $r_j(t);\ j = 1, \ldots, 4;\ t = 0, 1, 2, \ldots$, are used by the PPE to generate on-line the four posterior probabilities, $P_j(t);\ j = 1, \ldots, 4;\ t = 1, 2, \ldots$, which are next used to generate the overall RMMAC control signal $u(t)$ by probabilistic weighting, i.e.

$$u(t) = \sum_{j=1}^{4} P_j(t)\, u_j(t) \tag{6.18}$$

where the $u_j(t)$ are the "local" controls generated by each LNARC, $K_j(s)$, designed in Section 6.3.

6.6. RMMAC Stochastic Simulations and Performance Evaluations

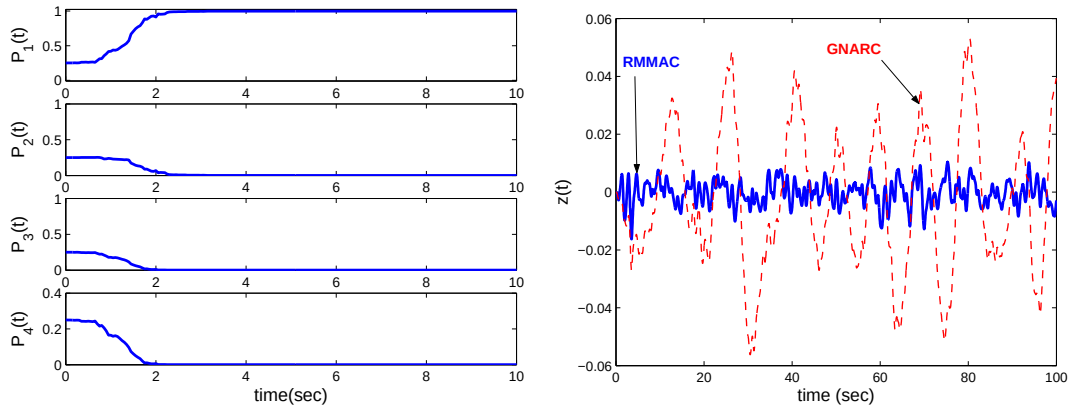
Unless stated otherwise:
(a) all simulations use a plant-stochastic disturbance and white measurement noise generated according to (6.1) and (6.6). The true system includes an actual (but unmodeled) "legal" time-delay of 0.01 secs in the control channel.
(b) all initial model probabilities are initialized to be $P_k(0) = 0.25$ $(k = 1, \ldots, 4)$ at $t = 0$ secs.
(c) we present numerical averages for 5 MC simulations.

In the sequel some representative stochastic simulations are shown using the complete RMMAC closed-loop system. Due to space limitations, we only show "typical" plots; however, our conclusions are based on thousands of other MC runs not explicitly shown in this paper [2].

6.6.1. "Easy" Identification, I: The dynamic evolution of the four posterior probabilities when the true $k_1 = 1.65$, well inside the Model #1 subinterval, and the corresponding outputs for the RMMAC and the GNARC systems are shown in Fig. 19. The correct model (Model #1) is identified quickly in about 2 secs. The significant improvement in disturbance-rejection by the RMMAC vis-a-vis that of the GNARC is evident as shown in 19(b).

6.6.2. "Easy" Identification, II: The dynamic evolution of the four posterior probabilities when the true $k_1 = 0.3$, well inside the Model #4 subinterval, and the corresponding outputs for the RMMAC and the GNARC systems are shown in Fig. 20(a). The correct model (Model #4) is identified quickly in about 10 secs. The improvement in disturbance-rejection by the RMMAC is again evident as shown in Fig. 20(b).

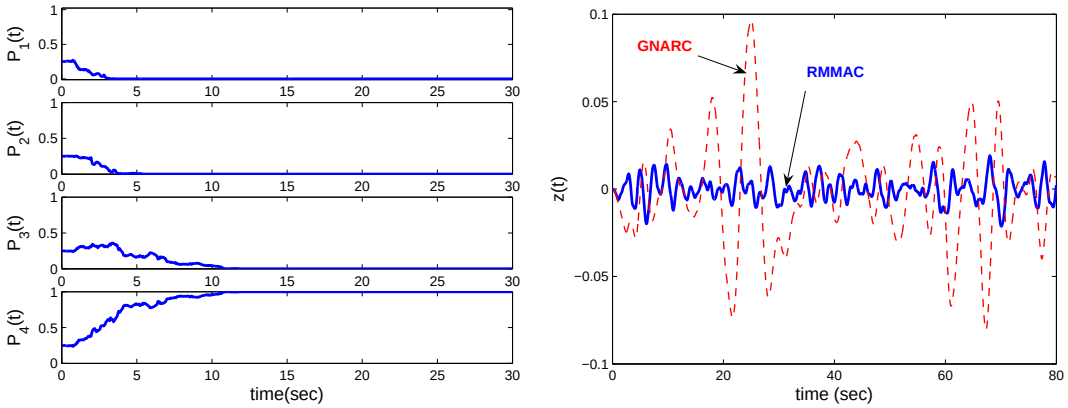
6.6.3. "Harder" Identification: When the actual spring constant is near the boundary between two models, it takes longer (more data) to resolve the true hypothesis. In this example, $k_1 = 0.405$ is selected which belongs to Model #3 but is also very "close" to Model #4, see



(a) RMMAC Posterior probabilities

(b) Output performance comparisons

Figure 19. Simulation results for $k_1 = 1.65$ in Model #1.



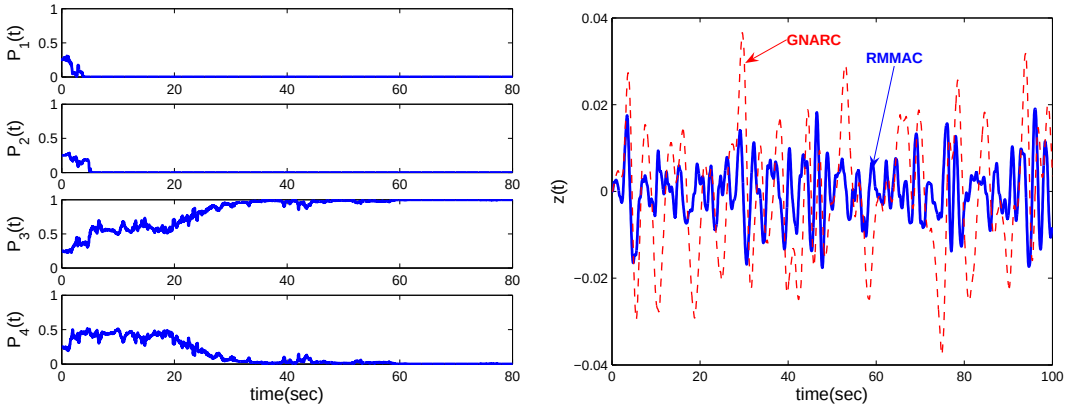
(a) RMMAC Posterior probabilities (b) Output performance comparisons

Figure 20. Simulation results for $k_1 = 0.3$ in Model #4.

Fig. 18. The probabilities vs time as well as output comparisons are shown in Fig. 21. It takes about 50 secs to resolve the ambiguity between Models #3 and #4.

Fig. 21(a) shows that the probabilistic weighting of the control according to (6.18) persists for about 50 secs. However, as evidenced by Fig. 21(b), there is no significant degradation of the RMMAC disturbance-rejection as compared to the GNARC.

In order to demonstrate that the probabilistic weighting (6.18) works very well we present the following result. In Fig. 22 the performance of the nonadaptive system is shown when we fix $P_3(t) = 1, \forall t = 0, 1, 2, \ldots$, i.e. *when we have "perfect" identification from the start,* with that of the RMMAC and compare the resulting output response. The results shown in Fig. 21 demonstrate that the probabilistic averaging results in *insignificant* performance deterioration. Hence our "probabilistic gain-scheduling" works well.



(a) RMMAC Posterior probabilities (b) Output performance comparisons

Figure 21. Simulation results for $k_1 = 0.405$ (in Model #3), but also close to Model #4.

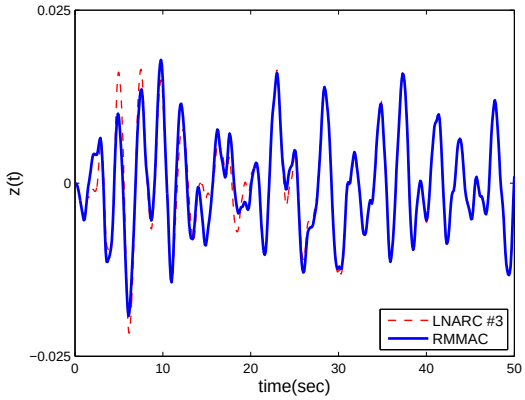


Figure 22. LNARC #3 (perfect identification) and RMMAC output comparisons for $k_1 = 0.405$. Note the insignificant deterioration in disturbance rejection by the RMMAC.

6.6.4. Mismatch Model/LNARC Instability: In Section 6.4 we raised the critical issue of Model/LNARC mismatch instability. In Table III the mismatch-stability properties of the LNARC designs are summarized. We discuss this important issue by some simulations.

First, we evaluate the RMMAC response *when we force it to be unstable at time* $t = 0$. Fig. 23 illustrates a typical result selected from several different MC simulations. In Fig. 23 the true value of k_1 is 1.75 in Model #1. From Table III we know that if we use LNARC #4, $K_4(s)$, with Model #1 we have an unstable closed-loop system. In order to *force this initial instability*, the initial values of the probability vector are selected to be

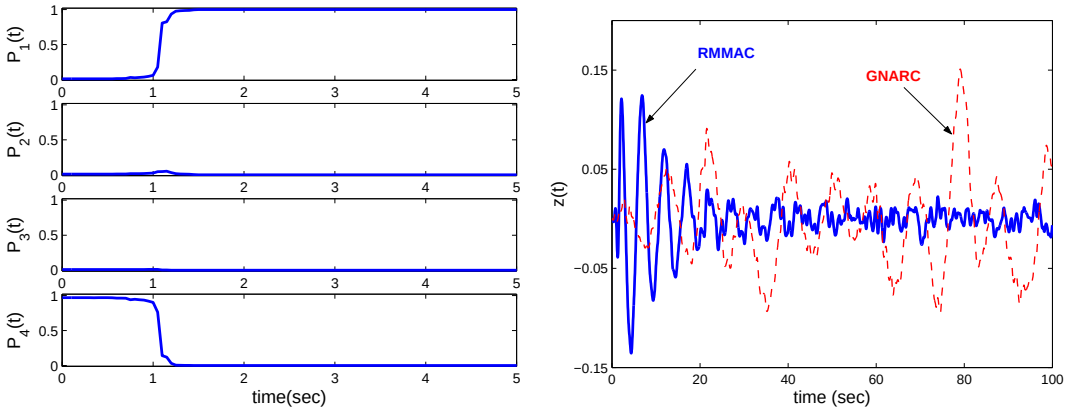
$$P_1(0) = P_2(0) = P_3(0) = 0.01,\ P_4(0) = 0.97$$

so that initially, at least at $t = 0$, the RMMAC system is *forced to be unstable*. However, as illustrated in Fig. 23, the RMMAC rapidly recovers to a stable configuration. Fig. 23(a) shows that the "correct probability" $P_1(t) \to 1$ within 1.3 secs, starting from its initial value $P_1(0) = 0.01$; the other three probabilities converge to zero within 1.3 secs as well. Fig. 23(b) shows the output response in which, after an initial period of brief "instability", the RMMAC recovers and returns to its predictable superior disturbance-rejection.

Fig. 24 illustrates another mismatch-instability result, similar to the above case with $k_1 = 1.75$ in Model #1. This test was suggested by Prof. B.D.O. Anderson. To force this instability, using LNARC #4 as above, we force the posterior probabilities starting at time $t = 60$ secs to be

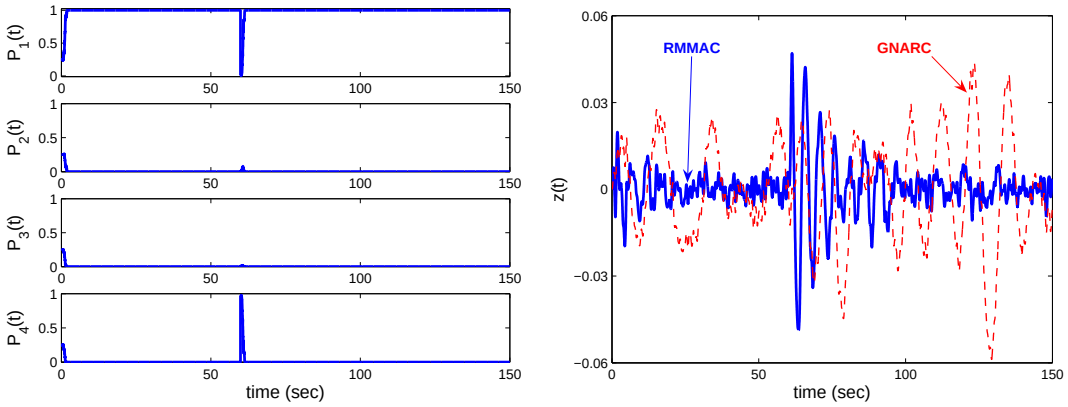
$$P_1(T) = P_2(T) = P_3(T) = 0.01,\ P_4(T) = 0.97; \quad \text{for all T, } 60 < T < 60.1 \text{ secs}$$

This forces the RMMAC system to be unstable at time $t = 60$ for 0.1 secs which corresponds to 10 discrete-time measurements. Fig. 24 shows that the RMMAC rapidly recovers to a stable expected configuration. Fig. 24(a) shows that initially $P_1(t) \to 1$ so that the correct Model #1 is identified. Then it is forced to $P_1(60) = 0.01$. Fig. 24(b) shows the output response in which the RMMAC quickly recovers, following some oscillations induced by the forced instability, re-identifies the correct Model #1 and returns to its predictable superior disturbance-rejection.



(a) Posterior probabilities transients (b) Output response

Figure 23. The RMMAC recovers from forced unstable configuration at $t = 0$, when $k_1 = 1.75$ in Model #1.



(a) Posterior probabilities transients (b) Output response

Figure 24. The RMMAC recovers from a forced unstable configuration at $T = 60$ secs with $k_1 = 1.75$ in Model #1.

Comments on Instability: The RMMAC is designed to operate in a stochastic environment. Thus, it is conceivable that very unlikely events of very small probability can infrequently occur. Such an event may occur by having at the same instant of time a plant noise and sensor noise, say, at the 10+ sigma-level of their associated gaussian distributions and this would lead to an unstable RMMAC mismatch configuration of the type shown in Figs. 23 and 24. In the case of Fig. 24 the same unlikely event must persist for 10 consecutive measurements and since the white noises are independent this event has truly infinitesimal probability of occurrence. Even so, as long as the remainder of the theoretical assumptions are valid, Figs. 23 and 24 demonstrate that the RMMAC can rapidly recover as long as the forced

temporary instability does not make the KF residuals to be completely inconsistent with their precomputed covariances – see eq. (2.4). This fact is supported by numerous similar simulations [2] not shown here.

For the system discussed in this subsection, a *forced instability* starting at $t = 60$ secs and lasting until $t = 62.6$ secs (a total of 260 discrete-time consecutive measurements) caused one out of 10 Monte Carlo runs to be completely unstable, while nine recovered and eventually identified the correct model after a prolonged period of violent oscillations (as expected). A *forced instability* for the time interval $60 \leq t \leq 63.4$ (a total of 340 consecutive measurements) caused all 10 MC RMMAC simulations to be unstable. Clearly, the probability of such events is immensely infinitesimal!

Future adaptive simulations, especially those involving multiple-models, should explicitly report upon such mismatch-instability simulations. After all, the more models we use the higher the performance and loop-gain and of the associated local compensators, i.e. LNARCs. This fact will certainly imply that a large subset of the N local compensators can potentially lead to severe mismatch-unstable combinations.

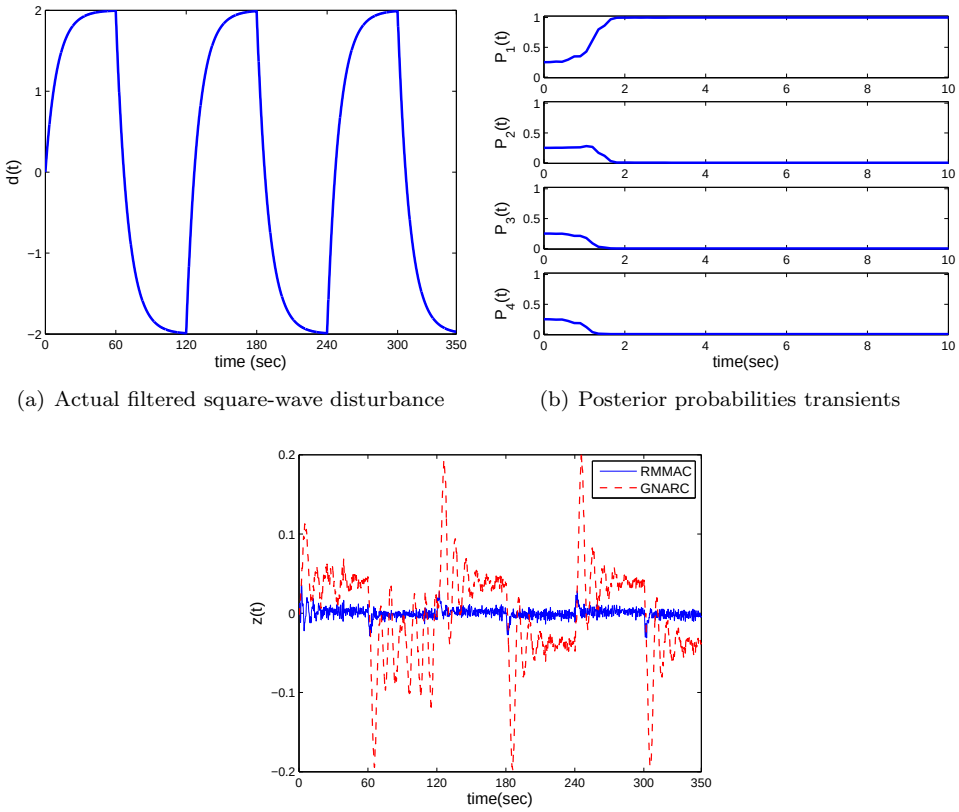
6.7. Mild Violations of the RMMAC Convergence Assumptions

The theory which guarantees the convergence of the posterior probabilities [61, 62] assumes that all MMAE signals are stochastic stationary random processes. With some additional ergodicity conditions, all results presented in Section 6.6 satisfied those assumptions, and we have indeed observed convergence to a nearest probabilistic neighbor. In this section we evaluate the RMMAC performance when we intentionally "mildly violate" some of the above assumptions, because this will always happen for real systems. We shall show that the RMMAC still performs quite well; in a sense it appears "robust" to mild violations of the theoretical assumptions. We evaluated the RMMAC performance over a wide variety of operating conditions, for different values of the uncertain spring stiffness [2]. In all the RMMAC worked well and no instabilities were observed. The following representative cases are presented.

6.7.1. Step Plant Disturbance: In this set of simulations we used a deterministic periodic square-wave disturbance, $\xi(t) = \pm 2.0$, with a period of 60 secs, rather than pure plant white-noise. The sensor noise was white as in Section 6.6. The KFs in the RMMAC were NOT aware of the square-wave ξ disturbance; they continued to use (6.1) to model the disturbance dynamics. The true spring stiffness is $k_1 = 1.75$ in Model #1. Fig. 25 shows the simulation results for one MC run.

Note that after 2 secs the posterior probabilities converge to the correct Model #1, i.e. $P_1(t) \rightarrow 1$ as shown in Fig. 25(a). The RMMAC performance is excellent, as shown in Fig. 25(c).

6.7.2. Sinusoidal Sensor Noise: We also tested "robustness to the assumptions" by using a high frequency sinusoid for the measurement noise, $\theta(t) = 10^{-3}\, sin(10t)$, rather than pure white noise. Fig. 26 shows a representative simulation using the value $k_1 = 0.3$, which is in Model #4. Fig. 26(a) shows that the "correct" probability $P_4(t) \rightarrow 1$ within 5 secs. The improvement in disturbance-rejection is again obvious from Fig. 26(b). Fig. 26(c) shows a comparison of the control signals. Both GNARC and RMMAC control signals have sinusoidal



(a) Actual filtered square-wave disturbance

(b) Posterior probabilities transients

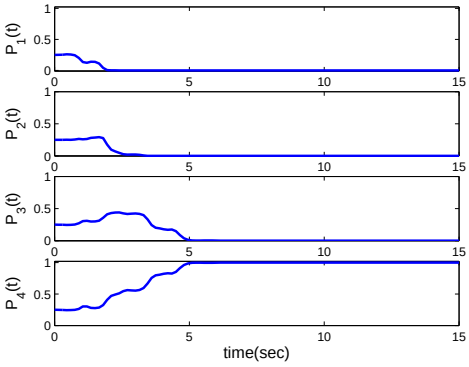
(c) Output response

Figure 25. RMMAC performance when the actual disturbance $d(t)$ is a ± 2 periodic (filtered) square-wave. The true $k_1 = 1.75$ (in Model #1).

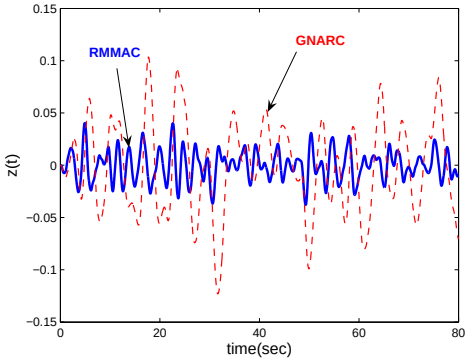
components at steady-state. As expected, the amplitude of the RMMAC control is slightly larger than that of the GNARC since the RMMAC yields improved disturbance-rejection. Only one MC run is shown.

6.7.3. Slow Parameter variation: As mentioned in Section 1, a driving engineering motivation for using adaptive control is the need to deal with "slow" changes in the plant uncertain parameters. In all the numerical simulations presented up to now, we constrained the uncertain parameter to remain constant for all time. Of course, the presence of a time-varying spring stiffness violates the plant LTI assumption, and hence all stationarity and ergodicity assumptions (required to prove the posterior probability convergence results) do not hold. Nevertheless, it is important to understand, for any adaptive system, its behavior and performance in the presence of slow parameter variations.

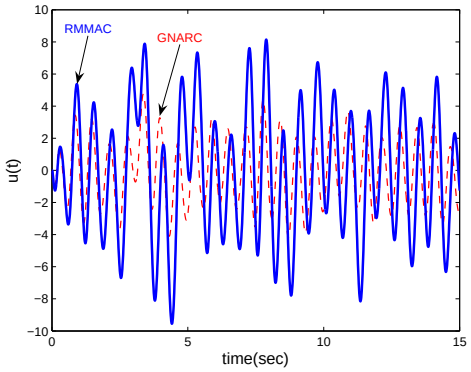
In the following numerical MC simulations, the uncertain spring stiffness is assumed to be



(a) Posterior probabilities transients



(b) Output response



(c) Control signals

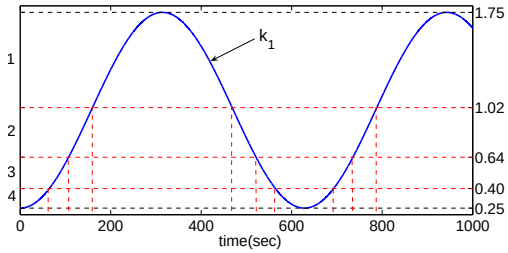
Figure 26. RMMAC robustness to sinusoidal sensor noise, for $k_1 = 0.3$ in Model #4.

sinusoidal with frequency 0.01 rad/sec, i.e.

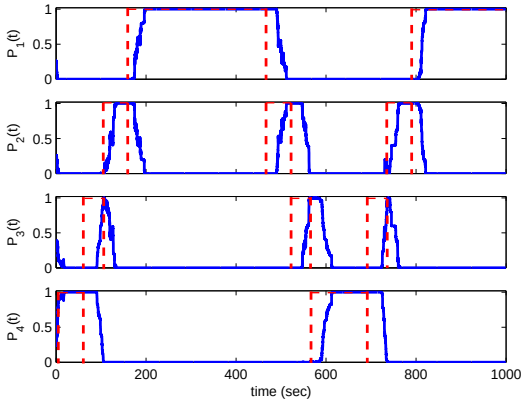
$$k_1(t) = 1 - 0.75 \cos 10^{-2} t \tag{6.19}$$

as shown in Fig. 27(a). The dashed-lines in Fig. 27(b) indicate the times that the spring stiffness crosses the boundaries of the four models of Table II. Fig. 27(b) also shows the dynamic evolution of the four posterior probabilities, while Fig. 27(c) compares the GNARC and RMMAC output responses and demonstrates that the RMMAC continues to work very well.

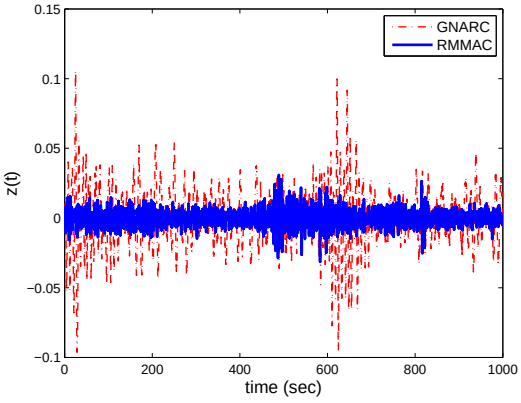
It is tempting to interpret Fig. 27(b) as demonstrating a "transient" in the identification process. This may well be true, but the reader should realize that the signals generated by the time-varying plant cannot be interpreted using "frozen model" reasoning. After all changing the spring stiffness according to (6.19) implies an exogenous energy transfer as a function of time, which is not accounted for in a "frozen" model. This opens up avenues for future research, especially with LPV system-theoretic concepts [71, 72, 73, 74]. Nonetheless, the results of Fig. 27 (and many others not presented herein) demonstrate the ability of the RMMAC in dealing with slowly-varying uncertain parameters. However, we cannot specify how slow is



(a) Sinusoidal spring stiffness, $k_1(t)$



(b) Posterior probabilities transients



(c) Output response

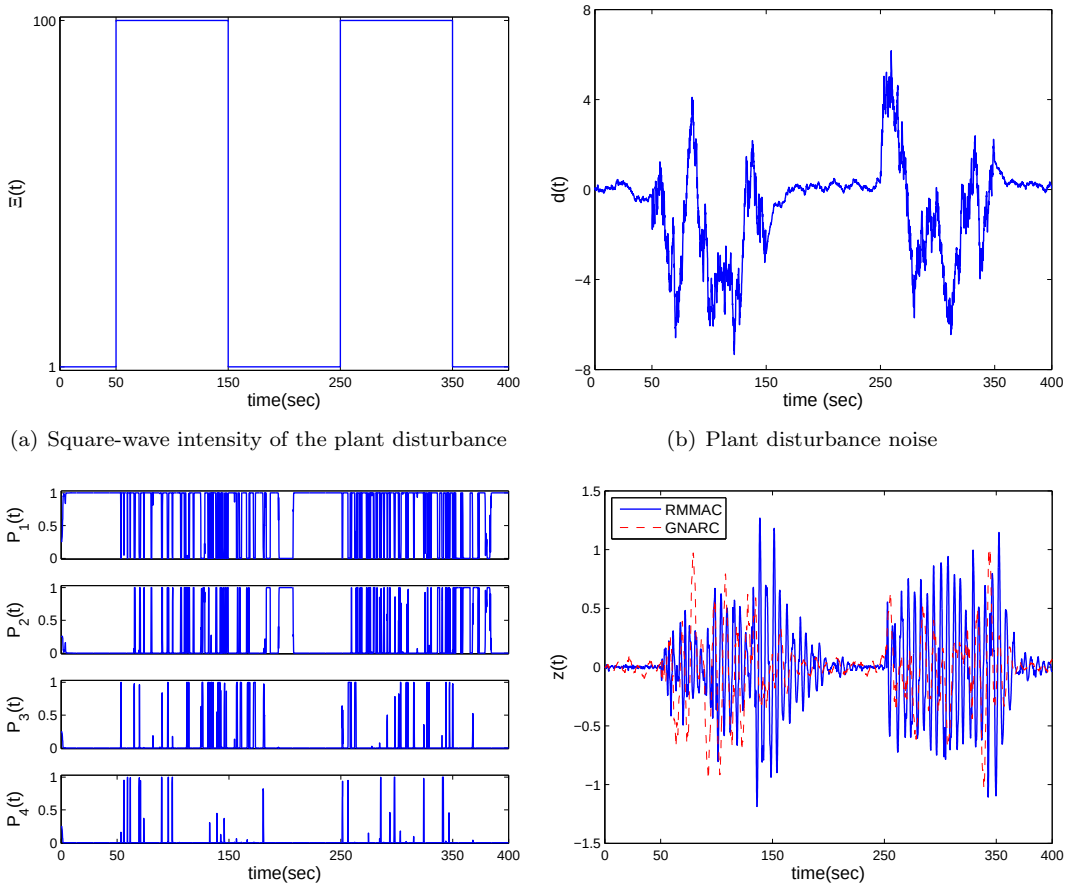
Figure 27. RMMAC responses for the sinusoidal parameter variation of eq. (6.19) using 5 MC simulations.

"slow", and still obtain satisfactory RMMAC performance, without simulations.

6.8. Widely Varying Disturbance Environments and the RMMAC/XI Architecture.

We shall now demonstrate the poor performance of the standard RMMAC when the plant disturbance environment is highly uncertain; this represents a *severe violation* of the underlying assumptions. We shall also demonstrate how the RMMAC/XI architecture, developed in Section 5, overcomes these shortcomings and yields outstanding disturbance-rejection.

First, let us suppose that the N=4 model standard RMMAC, designed for unit intensity, $\Xi = 1$, of the plant white noise $\xi(t)$ (see eq. (6.1)) has been implemented as in the previous simulations of this section. Let us now suppose that the actual plant white noise $\xi(t)$ is generated by the square-wave intensity sequence, shown in Fig. 28(a), which changes from the "normal intensity" $\Xi = 1$ to the "abnormal intensity" $\Xi = 100$. Thus, periodically, every 100 secs the standard RMMAC is subjected to the very large (an unanticipated) disturbance $d(t)$ shown in Fig. 28(b) generated by eq. (6.1). Figure 28(c) shows the evolution of the four posterior probabilities using the value k_1 =1.5 in Model #1. It is evident from Fig. 28(c) that during the time-intervals associated with the "normal" disturbance, $P_1(t) \to 1$ so that the correct model #1 is identified. However, during the time-intervals of the "abnormal" disturbance



(a) Square-wave intensity of the plant disturbance

(b) Plant disturbance noise

(c) Posterior probabilities

(d) The output $z(t)$

Figure 28. The standard RMMAC performance when we have a periodic severe violation of the theoretical assumptions. The KFs are designed for the "normal" intensity $\Xi = 1$, but the actual noise intensity is a periodic square-wave with $\Xi_{act} \in \{1, 100\}$. The true spring stiffness is k_1 =1.5 in Model #1. Only a single Monte Carlo simulation is shown.

all posterior probabilities switch rapidly and exhibit "model-identification confusion." The impact upon the standard RMMAC performance is shown in Fig. 28(d). There is no long-term instability; however, during the "abnormal" time-intervals the disturbance-rejection of the RMMAC is about the same as that of the (nonadaptive) GNARC. Hence, we conclude that this severe violation of the theoretical assumptions drastically degrades the RMMAC performance. Similar behavior was observed in many other "abnormal" simulations [2](not shown here).

We next demonstrate how the use of the RMMAC/XI architecture of Fig. 10 with N=4 alleviates the performance-deterioration noted above. We implement eight KFs. The first four KFs (k=1,2,3,4) are designed for the noise intensity Ξ=1 and are identical to those used before.

The last four KFs (k=5,6,7,8) are designed for the noise intensity Ξ=100; for these the BPM procedure discussed in Section 6.5 was repeated to design KF #5,6,7,8 (results not shown). The four LNARCs remain the same since they do not depend upon the value of the intensity Ξ.

We show simulations for the same spring constant k_1 =1.5 as in Fig. 28. In this case, the true model is Model #1 when Ξ=1 but changes to Model #5 when Ξ=100. Figure 29 shows the results obtained from simulations using the same square-wave Ξ intensity as in Fig. 28(a), repeated in Fig. 29(a). The time-evolution of the *eight* posterior probabilities is shown in Fig. 29(c); note that the posterior probabilities identify the correct models almost immediately, i.e.

$$P_1(t) \rightarrow 1 \; for \; t \in [0, 50] \cup [150, 250] \cup [350, 400]$$
$$P_5(t) \rightarrow 1 \; for \; t \in [50, 150] \cup [250, 350]$$

Fig. 29(b) compares the disturbance rejection of the standard RMMAC (from Fig. 28(d)) with that of the RMMAC/XI. It is clear that in this example the RMMAC/XI yields truly outstanding disturbance-rejection; similar conclusions were arrived at in [2] from numerous other similar simulations.

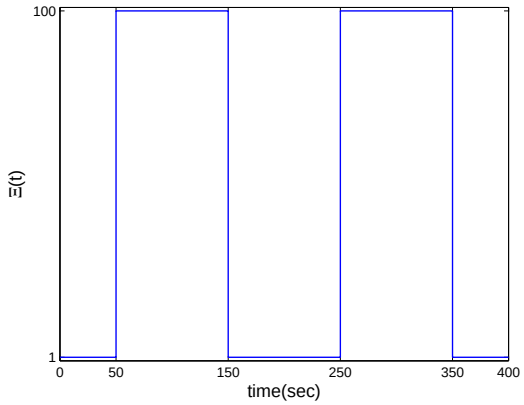
6.9. Discussion of Numerical Results

The stochastic simulation results presented in this paper demonstrate the excellent performance of the RMMAC system. Compared to the "best" non-adaptive design, GNARC, the RMMAC and RMMAC/XI consistently had superior disturbance-rejection. No closed-loop instabilities were noted in thousands of MC simulations, with the exception of forced prolonged mismatch instabilities as noted in Section 6.6.4.

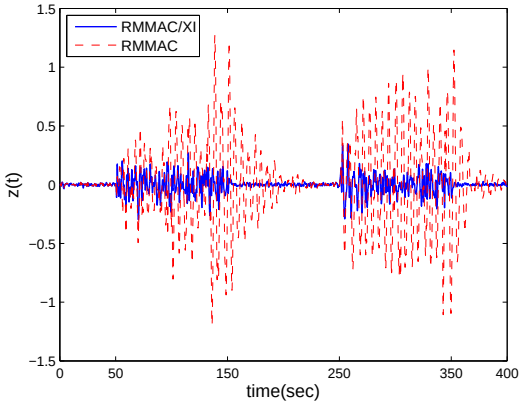
The example illustrated how to predict (and then validate) the potential performance characteristics of the RMMAC. This was done by analyzing the corresponding LTI feedback loops involving the GNARC and each of the four LNARCs so as to generate RMS predictions, frequency-domain visualizations for disturbance-rejection etc. We emphasize that this constructive capability is critical so that *quantitative tradeoffs* can be carried out, before one constructs and tests the full-blown RMMAC design with numerous MC simulations.

The RMMAC performance conclusions reported here validate earlier simulations by the authors. In [27] we analyzed, using a five-model RMMAC, an academic SISO non-minimum phase (NMP) plant with a single uncertain right-half plane zero which poses fundamental limits to superior disturbance-rejection near its frequency; see, for example, [70]. In that example, the GNARC "hedged" for the minimum value of the NMP zero. If the true zero was near its minimum value, the RMMAC performance was essentially the same as for the GNARC (no magic properties). The true value of the RMMAC shows when the true NMP zero is large; then the RMMAC produces truly superior performance vis-à-vis the GNARC. The interested reader may also review the RMMAC results reported in [29] for a different two-cart mass-spring-dashpot non-collocated system with a mass uncertainty. However, the previous results [27] and [29] did *not* use the systematic FNARC based approach for defining the models (in the spirit of Fig. 14) nor the optimized KF design using the BPM (in the spirit of Fig. 18). Several other SISO and MIMO results and examples can be found in the doctoral dissertation of S. Fekri [2].

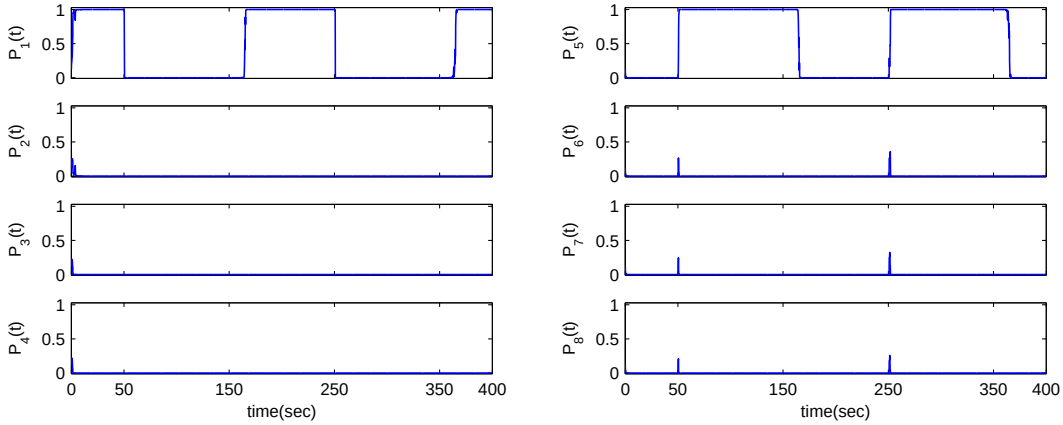
The computation time required by the RMMAC is modest for the numerical example considered in this section. In a single Monte Carlo simulation we carry out, in real-time, the



(a) Square-wave intensity of the plant disturbance



(b) The RMMAC output $z(t)$



(c) Posterior probabilities

Figure 29. The RMMAC/XI performance for varying plant noise intensity. The true spring constant is k_1 =1.5 in Model #1 when Ξ=1 but in Model #5 when Ξ=100. Note that the correct probabilities, $P_1(t)$ and $P_5(t)$, almost instantaneously go to unity whenever Ξ changes. Only a single Monte Carlo run, with the same seed as in Fig. 29, is shown.

required on-line calculations and simulations for the four KFs, the PPE and the four LNARCs using a sampling time of 0.01 secs. These were done *thirteen times faster than real-time* using a Toshiba Satellite Pro laptop with an Intel Pentium M 1.5GHz processor, running under Windows XP Pro (SP2) and MATLAB 7.0.

We also carried out several simulations using the "switched version" of the RMMAC discussed in Section 2.4 [2]. Our experience is that the "switched RMMAC" leads to *larger RMS errors during the identification transient interval* for constant uncertain parameters. After the posterior probability converges, of course both RMMAC versions behave in the same manner. This is the reason that we focused upon the RMMAC architecture of Fig. 4 that utilizes probabilistic weighting of the control signals via eq. (2.3).

Finally, we stress that all mixed-μ computations to determine the GNARC and LNARCs were carried out by the (as yet non-commercially available) software provided to us by Prof. G. Balas [48]. If we had designed these compensators with the commercial version of MATLAB [49], which uses only the D-K scales in the iterative design of the $\mathcal{H}_\infty$ compensators, they would be much more conservative and the performance improvements would be inferior [69].

7. DISCUSSION AND FUTURE RESEARCH DIRECTIONS

We have presented a brief overview of different approaches to adaptive control that utilize multiple-model architectures. Recent research in the area of the CMMAC and RMMAC architectures seems most promising, but we still have no widely accepted solution to the robust adaptive control problem. As a consequence, there is still a great deal of work that needs to be done, both theoretical and applied, to fully understand the similarities and difference among the two architectures and the variants within each one, and suitable extensions.

Any future theoretical research must adopt a design methodology. We strongly believe that the adaptive problem formulation should always include unmodeled dynamics, unknown plant disturbances and sensor noise, in addition to the (slowly-varying) uncertain real parameters. Furthermore, the design methodology should contain explicit performance specifications and performance-robustness should be an integral part of the adaptive problem formulation. Thus, closed loop-stability arguments are not enough. This implies that the "local" compensators, the LNARCs, must be designed to exhibit both local robust-stability and robust-performance, following the ideas presented in Section 3 and as utilized in our RMMAC architectures. In the unfalsified control methodologies, if robust-performance is to be considered, the available results must be significantly extended; most probably, the family of robust compensators available for unfalsified and safe control will also have to be designed using the methodology suggested in Section 3. Recent results in *unfalsified control* [30, 31] claim stability, but performance is not addressed (after all, an open-loop stable highly-uncertain plant remains robustly-stable with no feedback, albeit with lousy performance).

The structure of the multi-controllers used in the SMMAC architecture does not explicitly reflect a robust-performance requirement. Indeed, for the most part, the SMMAC multi-controllers have a common A_c-matrix (so all controllers have the same poles). This was done for the purpose of computational simplicity and "bump less transfer" to avoid control transients. However, such controller structures are not appropriate if performance-robustness is desired. A recent exception [75] uses multi-controllers that have an LQG structure, but they use pole-placement ideas, which are also notorious for their lack-of-robustness properties. On the other hand, the RMMAC architecture deals directly with such robust-performance issues and the dynamic characteristics of the RMMAC compensators are quite different.

We believe that, as stressed in Section 3, one cannot arrive at a systematic procedure for defining the number of models required in any multiple-model scheme without explicit performance specifications. Specification of complexity requires fixing the number, N, of models; however, one still needs to properly calculate their "boundaries." Specification of the required performance, such as in the % FNARC method, would naturally lead to the required number of models and their boundaries. In either case, one needs to utilize the mixed-μ methodology to maximize performance and derive the LNARCs.

We note that for many industrial applications, where "tuned" PID controllers are judged to yield satisfactory performance, designing the LNARCs may be an overkill. Based upon operator experience, one can define the N models in an *ad-hoc* manner as opposed to systematic procedures of Section 3. In such cases, one can still design the KFs, using the BPM methodology of Section 4 and obtain and implement the "identification subsystem" of the resulting RMMAC.

We now consider the "system identification" part of the SMMAC and RMMAC architectures. In the RMMAC architecture the approach is stochastic and relies on the residuals generated by the bank of KFs. These must be designed very carefully so as to meet the probability convergence results, as explained in Section 4 and Appendices I and II. In the SMMAC architectures the approach is deterministic and relies upon the prediction errors generated by a bank of multi-estimators, which are a special class of Luenberger observers. In all cited SMMAC references, with the exception of [75], each estimator uses the same "A_E" matrix, i.e. all observers have identical stable poles. This was done to minimize on-line computation. An integral norm of the prediction errors is used for the system identification function. Research should be done to remove the common "A_E" matrix assumption and use more general observers. Also, the assumption of scalar prediction errors must be removed to make the SMMAC architectures extendable to the MIMO case. In [75] this was done, but the problem of designing suitable deterministic MIMO Luenberger observers, say by eigenstructure assignment, becomes quite complex.

Our experience has shown that the RMMAC can effectively deal with "slowly time-varying" parameters. It may be interesting to examine if one can integrate the RMMAC philosophy and methodology with the evolving theory of *Linear Parameter Varying (LPV)* systems [36, 37, 38, 39, 40, 41, 71, 72, 73, 74, 76].

A weakness of the MMAE algorithm is that the Baram Proximity Measure (BPM) – see Appendix II – cannot be defined or calculated for *open-loop unstable plants*. Thus, if in an RMMAC application the plant is unstable one would not be able to use the ideas of Section 4, which are based on the BPM, to design the KFs. One can still design KFs (and RMMACs) for unstable plants, but one would most likely have to use trial-and-error methods in designing the KFs and ensuring convergence of the posterior probabilities. This defines a key and very important area for future research.

With respect to the RMMAC and RMMAC/XI architectures, if one or more of the LNARCs is unstable (but leads to a stable feedback loop) remains an open question. All the LNARCs that we have designed [2] were stable. However, if the open-loop plant is both unstable and nonminimum phase then some of the LNARCs may be unstable. Since we cannot define the BPM for unstable open-loop plants noted above, this remains a very important future research area for RMMAC architectures.

Finally, we need to stress that in all available SMMAC and RMMAC results we cannot prove – as yet – *global* robust-stability, because a suitable theory is lacking for handling the nonlinear time-varying (stochastic) closed loop dynamics arising from these architectures. Needless to say, any advances in stability theory along these directions would be welcomed. At present, the promise of the SMMAC and RMMAC architectures is in their "local" and "asymptotic" results related to stability and performance. With respect to the RMMAC one can design real-time monitoring systems for the "regularity" of the residuals. If "abnormal" situations are observed (see the discussion in Sections 2 and 6), which may lead to long-term instability, one should be able to design a "fail-safe" logic that disconnects the LNARCs and uses the GNARC until the residuals return to a "regular" behavior and the LNARCs are reconnected.

Equally important to the theoretical research investigations would be to fairly compare the different adaptive methods using a variety of common "test-bed examples", with identical performance requirements.

8. CONCLUDING REMARKS

We have witnessed the development and evolution of a variety of novel multiple-model architectures for robust adaptive control during the past decade. In order to compare them in a fair manner we have suggested that all should utilize the available results and software associated with mixed-μ synthesis. If we explicitly state the specifications for required stability-robustness and performance-robustness for the adaptive closed-loop system, then we can evolve the present architectures so that they can be confidently used in real applications.

We view our RMMAC architecture and design methodology as a potential contribution to adaptive feedback engineering system design. As we have remarked, we can not prove global stability results and we only have numerous simulation studies that demonstrate its excellent performance. Thus, a reader may ask an important question: *why should one pay attention to the RMMAC procedure in the absence of solid theoretical guarantees?*

The best way to view our perspective is by an analogy. Thousands of real engineering systems are operating using methods that also lack a solid theoretical foundation, but are based upon common-sense integration of rigorous results. We list just three, all of them involving nonlinear systems:

(a) *gain-scheduling* feedback control system designs (with no global stability guarantees),

(b) surveillance, tracking and other estimation systems that utilize the *extended Kalman filter* (which may diverge), and

(c) surveillance, tracking systems and other estimation systems that are based on the *sum-of-gaussian* multiple model estimation method, when the extended Kalman filter performance is not good enough (but they still may not always work well).

None of the above "practical" design methods is backed by a complete body of solid theory. Their extensive engineering utility is based upon exhaustive simulations, engineering insight and know-how to inevitably "tune" certain parameters in the algorithms. However, all utilize reasonable extensions of theoretical results valid for linear systems, e.g. the Kalman filter. The gain-scheduling method also pieces together linear feedback control results in a common sense manner. Since almost all real applications involve nonlinear systems the engineering utility of (a), (b) and (c) is extensive. Our perspective is that the highly nonlinear proposed RMMAC method is akin to the above. Maybe we are nearing the end of the search for the adaptive "Holy Grail."

ACKNOWLEDGEMENT

We thank Profs. B.D.O. Anderson and J. Hespanha for their frequent suggestions, discussions and criticisms throughout the course of our research. We also are grateful to Prof. G.J. Balas for providing us with the latest non-commercial version of the mixed-μ software [48] and Prof. Y. Baram for providing us a copy of his Ph.D. thesis [61]. We are also indebted to the four anonymous reviewers whose comments and criticisms greatly improved the final version of this paper.

APPENDIX I. SUMMARY OF THE MMAE FILTER

In this appendix, the equations of the multiple model adaptive estimation (MMAE) algorithm, shown in Fig. 1, and some of its properties are summarized. Here we shall restrict our attention to linear systems driven by white Gaussian inputs having time-invariant statistics. It is also assumed that the system has attained steady-state, i.e. that all signals of interest are stationary.

Consider the discrete-time (unknown) system

$$\begin{aligned} x(n+1) &= F_* x(n) + J_* u(n) + G_* w(n) \\ y(n) &= H_* x(n) + v(n) \end{aligned} \tag{A.1}$$

where $\{u(n)\}$ is the deterministic (control) input and $\{w(n)\}$ and $\{v(n)\}$ are zero-mean white Gaussian sequences, mutually independent with

$$\begin{aligned} E\{w(n)w(n)^T\} &= Q_* \\ E\{v(n)v(n)^T\} &= R_* \end{aligned} \tag{A.2}$$

Let us assume that N discrete-time stochastic LTI models are given that include disturbance dynamics (if any). These models blend a finite set of families of models

$$\wp = \{\mathcal{M} : \mathcal{M}_k = (F_k, J_k, G_k, H_k, Q_k, R_k)\,; k = (1, 2, \ldots, N)\} \tag{A.3}$$

The k-th model is then described by the state space dynamics

$$\begin{aligned} x_k(n+1) &= F_k x_k(n) + J_k u(n) + G_k w(n) \\ y_k(n) &= H_k x_k(n) + v(n) \end{aligned} \tag{A.4}$$

where the time index is $n = 0, 1, 2, \ldots$ and the model index is $k = 1, 2, \ldots, N$.

It is assumed that the true system that generates the data is one of the models in (A.4). The set of past controls, $u(0), u(1), u(2), \ldots, u(n-1)$, and the set of past noisy measurements, including the one at the *present* time n, $y(1), y(2), \ldots, y(n-1), y(n)$, are also known at time n. We want to determine the true steady-state conditional mean of the present state vector, $x(n)$, i.e. as $n \to \infty$

$$\hat{x}(n|n) = E\big\{x(n)|\underbrace{u(0), u(1), \ldots, u(n-1); y(1), \ldots, y(n-1), y(n)}_{Y(n)}\big\}$$

and the true steady-state conditional covariance matrix of $x(n)$, i.e.

$$\Sigma(n|n) = E\big[\big(x(n) - \hat{x}(n|n)\big)\big(x(n) - \hat{x}(n|n)\big)'|Y(n)\big] \tag{A.5}$$

The MMAE filter shown in Fig. 1 is driven by the sequence of past controls and noisy sensor measurements while it generates both a state-estimate vector and a corresponding error-covariance matrix. Let us first establish some notation for the discrete-time steady-state Kalman filter (KF).

Predict-cycle:

$$\begin{aligned} \hat{x}_k(n+1|n) &= F_k \hat{x}_k(n) + J_k u(n) \\ \hat{y}_k(n+1|n) &= H_k \hat{x}_k(n+1|n) \end{aligned} \tag{A.6}$$

Update-cycle:

$$\hat{x}_k(n+1|n+1) = \hat{x}_k(n+1|n) + K_k r_k(n+1) \tag{A.7}$$

The residual $r_k(\cdot)$, residual covariance matrix S_k, and the constant steady-state KF gain matrix, K_k, are respectively defined as follows.

Residual:

$$r_k(n+1) = y(n+1) - \hat{y}_k(n+1|n) \tag{A.8}$$

Residual covariance:

$$S_k = cov[r_k(n+1); r_k(n+1)] = H_k \Sigma_k^p H_k^T + R_k \tag{A.9}$$

KF gain:

$$K_k = \Sigma_k^p H_k^T S_k^{-1} \tag{A.10}$$

The constant steady-state covariance equations are
Predict-cycle covariance Σ_k^p :

$$\Sigma_k^p = F_k \Sigma_k A_k^T + G_k Q_k G_k^T \tag{A.11}$$

Update-cycle covariance Σ_k :

$$\Sigma_k = \Sigma_k^p - \Sigma_k^p H_k^T S_k^{-1} H_k \Sigma_k^p \tag{A.12}$$

We stress that both the MMAE state-estimate and state-covariance matrix (A.5) represent true conditional state estimate and its conditional covariance at steady-state [54, 57]. This is because, one can explicitly calculate the conditional probability density function $p\left(x(n)|Y(n)\right)$, which turns out to be a weighted sum of gaussian densities, where the weights are found from the posterior probability evaluator (PPE); see Fig. 1.

The problem reduces to a combination of a hypothesis-testing problem and a state-estimation problem. The fact that one of the N models is the true one is modeled by a hypothesis random variable that must belong to a discrete set of hypothesis $\{\mathcal{H}_1, \mathcal{H}_2, \ldots, \mathcal{H}_N\}$. It turns out that online generation of the posterior conditional probabilities determines which hypothesis is true. Let suppose that H indicates the hypothesis random variable (scalar) which can attain only one of N possible values,

$$\mathcal{H} \in \{\mathcal{H}_1, \mathcal{H}_2, \ldots, \mathcal{H}_N\} \tag{A.13}$$

The event $\mathcal{H} = \mathcal{H}_k$ means that the k-th system is the true one, i.e. the one that is generating the data. The prior probabilities at initial time $n = 0$, $P_k(0) \equiv Prob(\mathcal{H} = \mathcal{H}_k)$, are assumed known (typically $P_k(0) = \frac{1}{N}$), and

$$P_k(0) \geq 0, \ \sum_{k=1}^{N} P_k(0) = 1 \tag{A.14}$$

The posterior probabilities, $P_k(n) = Prob(\mathcal{H} = \mathcal{H}_k|Y(n))$, must also satisfy

$$P_k(n) \geq 0, \ \sum_{k=1}^{N} P_k(n) = 1 \tag{A.15}$$

and can be calculated on-line by the PPE in the MMAE system. It turns out that the conditional PDF, $p\left(y(n+1)|u(n), \mathcal{H}_k, Y(n)\right)$, is Gaussian with mean

$$E\left\{y(n+1)|u(n), \mathcal{H}_k, Y(n)\right\} = H_k \, \hat{x}_k(n+1|n) \tag{A.16}$$

and steady-state covariance

$$cov\left[y(n+1); y(n+1)|u(n), \mathcal{H}_k, Y(n)\right] = H_k \Sigma_k^p H_k^T + R_k \triangleq S_k \tag{A.17}$$

Furthermore,

$$p\left(y(n+1)|u(n), \mathcal{H}_k, Y(n)\right) = \frac{1}{(2\pi)^{m/2}\sqrt{\det S_k}} \cdot e^{-\frac{1}{2} r_k^T (n+1) \, S_k^{-1} \, r_k(n+1)} \tag{A.18}$$

By using Bayes rule we deduce that

$$P_k(n+1) = \frac{p\big(y(n+1) \,|\, \mathcal{H}_k, u(n), Y(n)\big)}{\sum_{j=1}^{N} P_j(n) p\left(y(n+1) \,|\, \mathcal{H}_j, u(n), Y(n)\right)} \cdot P_k(n) \tag{A.19}$$

For notational simplicity, define

$$w_j(n+1) = r_j'(n+1)\, S_j^{-1}\, r_j(n+1) \tag{A.20}$$

$$\beta_j = \left((2\pi)^{m/2}\sqrt{\det S_j}\right)^{-1} \tag{A.21}$$

where m is the number of measurements. Then, from (A.19)–(A.22) the posterior probabilities can be computed on-line by the PPE using the recursive formula

$$P_k(n+1) = \frac{\beta_k\, e^{-\frac{1}{2}w_k(n+1)}}{\sum_{j=1}^{N} \beta_j\, e^{-\frac{1}{2}w_j(n+1)} P_j(n)} \cdot P_k(n) \tag{A.22}$$

where $P_k(0)$ are the prior model probabilities as in (A.14).
Thus, at steady-state the MMAE generates the state estimate (exact conditional mean) by

$$\hat{x}(n|n) = \sum_{k=1}^{N} P_k(n)\, \hat{x}_k(n|n) \tag{A.23}$$

and the exact conditional covariance matrix

$$\Sigma(n|n) = \sum_{k=1}^{N} P_k(n) \cdot \left[\Sigma_k(n|n) + \left(\hat{x}_k(n|n) - \hat{x}(n|n)\right)\left(\hat{x}_k(n|n) - \hat{x}(n|n)\right)'\right] \tag{A.24}$$

We stress that the above results hold when the true (unknown) plant is assumed to belong to the model set of (A.4).

APPENDIX II. BARAM PROXIMITY MEASURE (BPM)

The purpose of this appendix is to summarize the underlying results from probability and estimation theory that yield the identification procedure which converges to a model in the set which is closest to the true model in an information-theoretic sense, as proved by [61]. The posterior probabilities will converge if certain ergodicity and stationarity assumptions of residuals (innovation signals) are true. These conditions are briefly discussed in this appendix; see [61] for more details.

Stationarity and Ergodicity

In this section we provide definitions and convergence results for ergodic sequences used in the paper. It is not intended to provide an elaborate presentation of the ergodicity concept. For a precise development of ergodicity theory the reader is referred to, e.g. [77, 78].

Definitions:

1. Consider a probability space (Ω, U, P). A transformation T from Ω to U is said to be *measure preserving* if

$$P(T^{-1}A) = P(A)$$

for all $A \in U$.

2. A stochastic sequence $\{x(n)\}$ on (Ω, U, P) is said to converge *almost everywhere* (a.e.) or almost surely (a.s.) to a random variable x on (Ω, U, P) if

$$\lim_{n\to\infty} x(n) = x \quad a.e.$$

3. Given a measure preserving transformation T, a U-measurable event A is said to be *invariant* if

$$T^{-1}A = A$$

4. Let $\{x(n)\}$ be a stochastic sequence on (Ω, U, P) with values in (R^ℓ, B^ℓ), where (R^ℓ) is the l-dimensional Euclidean space and (B^ℓ) is the σ algebra of Borel sets of R_l; see [79, Chap. 3]. Let B^ℓ_∞ be the σ-algebra of Borel sets of R^ℓ_∞ where $R^\ell_\infty = R^\ell \times R^\ell \times \cdots$. Then $\{x(n)\}$ is said to be *stationary* if for each $\kappa \geq 1$ and for every $C \in B^\ell_\infty$

$$P\left[\{x(1), x(2), \ldots, x(n)\} \in C\right] = P\left[\{x(\kappa+1), x(\kappa+2), \ldots, x(\kappa+n)\} \in C\right] \tag{B.1}$$

5. The covariance matrix defined by

$$R(\kappa) \equiv E\{x(n)\, x^T(n+\kappa)\}$$

plays a fundamental role in the study of stationary processes in the wide sense [77]. A zero mean stationary Gaussian process is *ergodic* if and only if [80, pp. 257–260]; [77, pp. 494]

$$\lim_{n\to\infty} \frac{1}{n+1} \sum_{\kappa=0}^{n} |\mathcal{R}(\kappa)|^2 = 0 \tag{B.2}$$

where $|\mathcal{R}(\kappa)|$ denotes the determinant of $\mathcal{R}(\kappa)$.

Calculating the Baram Proximity Measure (BPM) [61]

Consider a bank of N discrete Kalman filters, and a set of N models for the system of (A.1) described by

$$\begin{aligned} x_j(n+1) &= F_j x_j(n) + G_j w(n) \\ y_j(n) &= H_j x_j(n) + v(n) \end{aligned} \tag{B.3}$$

in the absence of deterministic inputs but with the assumption of a stationary processes. Note that these results can be extended to the case where the system (B.3) is driven by an additional deterministic (known) stationary random sequence..

Let $\mathcal{M}^* = \{* \cup \wp\}$ denotes the model set that also includes the true model, denoted by $*$.

For each $i, j \in \wp$ let

$$S_j \equiv E\left\{[y(n) - \hat{y}_j(n)][y(n) - \hat{y}_j(n)]^T | \mathcal{H}_i = \mathcal{H}_j\right\}$$

denote the residual covariance matrix generated by the j-th Kalman filter and

$$\Gamma^i_j \equiv E\left\{[y(n) - \hat{y}_j(n)][y(n) - \hat{y}_j(n)]^T | \mathcal{H}_i \neq \mathcal{H}_j\right\}$$

according to $\mathcal{M}_j$, when $\mathcal{M}_i$ is the correct model.

We shall use the following condition [61, pp. 77] in addition to the other stationarity and ergodicity conditions.

Condition B1 (C.B1). For each $j \in \mathcal{M}^*$, the residual covariance S_j exists and has a finite positive definite value.

A sufficient condition for $B1$ is that each model corresponding to $j \in \mathcal{M}^*$ is detectable and controllable. For each $j \in \mathcal{M}^*$, S_j is generated by the discrete Kalman filter $\#j$ corresponding to the model $\mathcal{M}_j = (F_j, G_j, H_j, Q_j, R_j)$.

Assuming that index 'i' denotes the true parameter in $\mathcal{M}^*$, the augmented dynamic equation generating simultaneously the state $x(n)$ and its updated estimate generated by the j-th Kalman filter, $\hat{x}_j(n)$, is

$$\begin{bmatrix} x_i(n+1) \\ \hat{x}_j(n+1) \end{bmatrix} = \begin{bmatrix} F_i & 0 \\ F_j K_j H_i & F_j(I - K_j H_j) \end{bmatrix} \begin{bmatrix} x_i(n) \\ \hat{x}_j(n) \end{bmatrix} + \begin{bmatrix} G_i & 0 \\ 0 & F_j K_j \end{bmatrix} \begin{bmatrix} w(n) \\ v(n) \end{bmatrix} \tag{B.4}$$

where K_j is the (steady-state) Kalman filter (KF) gain corresponding to the model $\mathcal{M}_j$. The KF gain matrix is given by

$$K_j = \Sigma_j H_j^T (H_j \Sigma_j H_j^T + R_j)^{-1}$$

where Σ_j denote the prediction error covariance matrix according to M_j.

For notational purposes, define:

$$\begin{aligned} F_j^i &\equiv \begin{bmatrix} F_i & 0 \\ F_j K_j H_i & F_j(I - K_j H_j) \end{bmatrix} \\ G_j^i &\equiv \begin{bmatrix} G_i & 0 \\ 0 & F_j K_j \end{bmatrix} \\ Q^i &\equiv \begin{bmatrix} Q_i & 0 \\ 0 & R_i \end{bmatrix} \\ H_j^i &\equiv \begin{bmatrix} H_i & | & -H_j \end{bmatrix} \end{aligned}$$

Then the matrix

$$\Psi_j^i(n+1) \equiv E\left\{ \begin{bmatrix} x_i(n+1) \\ \hat{x}_j(n+1) \end{bmatrix} \begin{bmatrix} x_i(n+1) \\ \hat{x}_j(n+1) \end{bmatrix}^T \right\}$$

is generated by the dynamic Lyapunov equation

$$\Psi_j^i(n+1) = F_j^i \Psi_j^i(n) {F_j^i}^T + G_j^i Q^i {G_j^i}^T \tag{B.5}$$

The (steady-state) limit of the Lyapunov equation of (B.5) is computed by:

$$\Psi_j^i = \lim_{n \to \infty} \Psi_j^i(n) \tag{B.6}$$

It exists and is finite if and only if the augmented state matrix F_j^i has all its eigenvalues inside the unit circle, i.e. the spectral radius satisfies

$$\rho(F_j^i) < 1 \tag{B.7}$$

This is the case if for each $j \in \mathcal{M}^*$, F_j has all its eigenvalues inside the unit circle and (F_j, H_j) is observable.

Therefore, to calculate the limit matrix Ψ_j^i of (B.5) we simply solve, using MATLAB, the discrete-time algebraic Lyapunov equation

$$\Psi_j^i = F_j^i \Psi_j^i {F_j^i}^T + G_j^i Q^i {G_j^i}^T \tag{B.8}$$

It can be shown that

$$\Gamma_j^i = H_j^i \Psi_j^i {H_j^i}^T + R_i \tag{B.9}$$

where Ψ_j^i is the steady-state solution of (B.5).

It can also be shown that the state covariance matrix can be calculated as follows.

$$\mathcal{R}(\kappa) = \begin{cases} H_j^i \Psi_j^i {H_j^i}^T + R_i = \Gamma_j^i & ; \kappa = 0 \\ H_j^i \Psi_j^i \left({F_j^i}^T \right)^\kappa {H_j^i}^T & ; \kappa > 0 \end{cases} \tag{B.10}$$

This is shown in [61, pp. 80–81]. However, note that the stability and observability of the models are only sufficient, and not necessary. In fact, [61, Theorem 5.1] has proved ergodicity of the state residuals $\{x(n) - \hat{x}_j(n)\}$, which is not necessary, to show the ergodicity of $\{y(n) - \hat{y}_j(n)\}$. Thus, in the sequel we shall directly use the following assumption.

Condition B2 (C.B2). For each $j \in \mathcal{M}^*$ the residual sequence $\{y(n) - \hat{y}_j(n)\}$ is ergodic.

The ergodicity condition of (C.B2) is important and must be checked. If it is not satisfied, one might use a "fake-white-plant-noise" as mentioned in Section 2.4; see [67, Chap. 7] for more details.

The conditional probability density function corresponding to Model #j ($j \in \mathcal{M}^*$) is

$$p_j(y(n)|Y(n-1)) = [(2\pi)^m |S_j|]^{-\frac{1}{2}} \cdot exp\{-\frac{1}{2}[y(n) - \hat{y}_j(n)]^T {S_j}^{-1} [y(n) - \hat{y}_j(n)]\} \tag{B.11}$$

where m is the dimension of $y(n)$, i.e. the number of measurements.

For each $j, k \in \mathcal{M}^*$ the mean information in $y(n)$ favoring "model" k against j is defined by [61, 62, 63]

$$\bar{I}_n(k;j) = E \log \frac{p_k\big(y(n)|Y(n-1), \mathcal{H} = \mathcal{H}_k\big)}{p_j\big(y(n)|Y(n-1), \mathcal{H} = \mathcal{H}_j\big)}$$

and the distance measure is defined as

$$d_n(k;j) = \left|\bar{I}_n(k;j)\right|$$

For each $j \in \mathcal{M}^*$, using $*$ to denote the true plant in $\mathcal{M}^*$, we have

$$E \log p_j\big(y(n)|Y(n-1), \mathcal{H} = \mathcal{H}_j\big) = -\frac{m}{2} \log 2\pi - \frac{1}{2} \log |S_j| - \frac{1}{2} tr\left\{S_j^{-1} \Gamma_j^*\right\} \tag{B.12}$$

and for each pair $j, k \in \mathcal{M}^*$

$$\bar{I}_n(k;j) = \bar{I}_n(j;k) = \frac{1}{2} \log |S_j| + \frac{1}{2} tr\left\{S_j^{-1} \Gamma_j^*\right\} - \frac{1}{2} \log |S_k| - \frac{1}{2} tr\left\{S_k^{-1} \Gamma_k^*\right\} \tag{B.13}$$

The Baram proximity measure (BPM) of the j-th filter denoted by L_j^i is generated by

$$L_j^i \equiv \log |S_j| + tr\left\{{S_j}^{-1} \Gamma_j^i\right\} \quad ; i, j \in \mathcal{M}^* \tag{B.14}$$

Therefore,

$$\bar{I}_n(k;j) = \frac{1}{2}\left(L_j^* - L_k^*\right) \quad ; i, j \in \mathcal{M} \tag{B.15}$$

Also,

$$\bar{I}_n(*;j) \geq 0 \quad ; j \in \mathcal{M} \tag{B.16}$$

It is easy to check that

$$d(*;j) \geq d(*;k)$$

if and only if

$$L_j^* \geq L_k^*$$

for which the following condition is essential.

Condition B3 (C.B3). There exists some parameter $k \in \mathcal{M}$ such that

$$L_k^* < L_j^* \quad \text{for all } j \in \mathcal{M} \; ; \; j \neq k \tag{B.17}$$

If the convergence conditions hold for the MMAE to converge, i.e. eq. (B.2) and conditions (C.B1)–(C.B3), then it will converge to the j-th filter (corresponding to the appropriate plant model) governed by

$$L_j^* = \min_i \{L_j^i\} \quad ; i = 1, \ldots, N$$

where L_j^i is the BPM of the models as in (B.14).

In conclusion, the Baram proximity measure (BPM), is an appropriate distance metric between the true model and each of the other models that form the bank of Kalman filters.

Remarks

The application of deterministic inputs, rather than stationary random sequences, to the dynamic system, for the purpose of identification and their optimal selection, is not addressed in this paper. Generally speaking, the convergence analysis in the presence of deterministic inputs requires a more elaborate analysis, a topic for future research.

Ref. [61, Chap. 6] discusses a case that any deterministic input sequence, that satisfies a suitable assumption of [61, Theorem 6.3], will provide convergence in the mean of the identification procedures at a certain rate; see [61] and [81] for more details.

Ref. [66] uses the prediction error parameter estimate obtained by minimizing a scalar function of the matrix

$$\sum_{\tau=0}^{t}[R^{1/2}(\tau)\, r(\tau;p)][R^{1/2}(\tau)\, r(\tau;p)]^T \tag{B.18}$$

where $R(\tau)$ is some positive definite weighting matrix, and shows that under general conditions of bounded fourth moments of the residuals $r(t;p)$, by searching over models leading to stable Kalman filters and overall system stability, this prediction error estimate converges into the set of models that yield the same output prediction as the true system in the following sense.

$$\lim_{t\to\infty} \inf \frac{1}{t+1} \sum_{\tau=0}^{t} |\hat{y}(\tau;p_0) - \hat{y}(\tau;p)|^2 = 0 \tag{B.19}$$

which is similar to the ergodicity condition (B.2). This set depends in general on the input signal and will be contained in the set of all models with the same input-output relation as the true model, if the input is general enough to excite all modes of the system. It is shown in [66] that a sufficient condition would be for $u(t)$ to be independent of the process noise and persistently exciting. The input $u(t)$ is persistently exciting if, for all M, there exists $\delta(M)$ and $N_0(M)$ such that

$$\delta I < \frac{1}{N} \sum_{1}^{N} u_M(t) u'_M(t) < \frac{1}{\delta} I$$

for $N > N_0$ and where

$$u'_M(t) \equiv [u'(t) \dots u'(t-M)].$$

REFERENCES

[1] M. Athans, S. Fekri and A. Pascoal, "Issues on robust adaptive feedback control," in *Preprints 16th IFAC World Congress, Invited plenary paper*, Prague, Czech Republic, July 2005, pp. 9–39.

[2] S. Fekri, "Robust adaptive MIMO control using multiple-model hypothesis testing and mixed-μ synthesis," Ph.D. dissertation, Instituto Superior Técnico, Lisbon, Portugal, 2005.

[3] I. Landau, *Adaptive Control: The Model Reference Approach.* NY, USA: Marcel Dekker, 1979.

[4] K.S. Narendra and A. Annaswamy, *Stable Adaptive Systems.* NJ, USA: Prentice-Hall, 1988.

[5] S. Sastry and M. Bodson, *Adaptive Control: Stability, Convergence and Robustness.* NJ, USA: Englewood Cliffs, Prentice-Hall, 1989.

[6] K. Åström and B. Wittenmark, *Adaptive Control*, 2nd ed. MA, USA: Addison-Wesley, 1995.

[7] P. Ioannou and J. Sun, *Robust Adaptive Control.* NJ, USA: Prentice-Hall, 1996.

[8] C. Rohrs *et al.*, "Robustness of continuous-time adaptive control algorithms in the presence of unmodeled dynamics," *IEEE Trans. on Automatic Control*, vol. 30, no. 9, pp. 881–889, Sep. 1985.

[9] B.D.O. Anderson *et al.*, *Stability of Adaptive Systems: Passivity and Averaging Analysis.* MIT Press, 1986.

[10] E. Mosca *et al.*, "Designing predictors for MIMO switching supervisory control," *Int. J. of Adaptive Control and Signal Processing*, vol. 15, pp. 265–286, July 2001.

[11] A. Morse, "Supervisory control of families of linear set-point controllers – part 1: Exact matching," *IEEE Trans. on Automatic Control*, vol. 41, no. 10, pp. 1413–1431, Oct. 1996.

[12] ——, "Supervisory control of families of linear set-point controllers – part 2: Robustness," *IEEE Trans. on Automatic Control*, vol. 42, no. 11, pp. 1500–1515, Nov. 1997.

[13] ——, *A bound for the disturbance - to - tracking - error gain of a supervised set point control system*, ser. *Perspectives in Control Theory and Applications,* (D. Normand-Cyrot, ed.). Berlin: Springer-Verlag, 1998, pp. 23–41.

[14] ——, "Analysis of a supervised set-point control containing a compact continuum of finite-dimensional linear controllers," in *Proc. MTNS*, Belgium, 2004.

[15] K.S. Narendra and J. Balakrishnan, "Adaptive control using multiple models," *IEEE Trans. on Automatic Control*, vol. 42, no. 2, pp. 171–187, Feb. 1997.

[16] B.D.O. Anderson *et al.*, "Multiple model adaptive control. part 1: Finite controller coverings," *Int. J. of Robust and Nonlinear Control*, vol. 10, no. 11-12, pp. 909–929, 2000.

[17] ——, "Multiple model adaptive control with safe switching," *Int. J. of Adaptive Control and Signal Processing*, vol. 15, no. 5, pp. 445–470, Aug. 2001.

[18] J. Hespanha *et al.*, "Multiple model adaptive control – part 2: switching," *Int. J. of Robust and Nonlinear Control*, vol. 11, no. 5, pp. 479–496, Apr. 2001.

[19] D. Willner, "Observation and control of partially unknown systems," Ph.D. dissertation, MIT, Cambridge, MA, USA, June 1973.

[20] C. Greene, "An analysis of the multiple model adaptive control algorithm," Ph.D. dissertation, MIT, Cambridge, MA, USA, 1978.

[21] M. Athans *et al.*, "The stochastic control of the F-8C aircraft using the multiple model adaptive control (MMAC) method – part I: Equilibrium flight," *IEEE Trans. on Automatic Control*, vol. 22, no. 5, pp. 768–780, Oct. 1977.

[22] H. Shomber, "An extended analysis of the multiple model adaptive control algorithm," Ph.D. dissertation, MIT, Cambridge, MA, USA, 1980.

[23] C. Greene and A. Willsky, "An analysis of the multiple model adaptive control algorithm," in *Proc. of the IEEE Conf. on Decision and Control*, 1980, pp. 1142–1145.

[24] G. Schiller and P. Maybeck, "Control of a large space structure using MMAE/MMAC techniques," *IEEE Trans. on Aerospace and Electronic System*, vol. 33, no. 4, pp. 1122–1130, 1997.

[25] P. Maybeck and J. Griffin, "MMAE/ MMAC control for bending with multiple uncertain parameters," *IEEE Trans. on Aerospace and Electronic System*, vol. 33, no. 3, pp. 903–912, 1997.

[26] K. Schott and B. Bequette, *Multiple Model Adaptive Control*, ser. *Multiple Model Approaches to Modeling and Control,* (R. Murray-Smith and T.A. Johnsen eds.). London: Taylor & Francis, 1997, ch. 11, pp. 269–292.

[27] S. Fekri, M. Athans, and A. Pascoal, "A new robust adaptive control method using multiple-models," in *Proc. 12th IEEE Mediterranean Conference on Control and Automation (MED'04)*, Kusadasi, Turkey, June 2004.

[28] ——, "RMMAC: A novel robust adaptive control scheme – Part I: Architecture," in *Proc. of the IEEE Conf. on Decision and Control*, Paradise Island, Bahamas, Dec. 2004, pp. 1134–1139.

[29] ——, "RMMAC: A novel robust adaptive control scheme – Part II: Performance evaluation," in *Proc. of the IEEE Conf. on Decision and Control*, Paradise Island, Bahamas, Dec. 2004, pp. 1140–1145.

[30] C. Wang *et al.*, "Cost-detectability and stability of adaptive control systems," in *Proc. of the joint IEEE Conf. on Decision and Control / European Control Conf.*, Seville, Spain, Dec. 2005.

[31] M. Stefanovic *et al.*, "Stability and convergence in adaptive systems," in *Proc. of the American Control Conf.*, Boston, MA, 2004.

[32] R. Wang *et al.*, "Unfalsified direct adaptive control using multiple controllers," in *Proc. AIAA GNC Conf.*, no. AIAA-2004-5223, Providence, RI, 2004.

[33] M. Safonov and T.-C. Tsao, *The Unfalsified Control Concept: A Direct Path From Experiment To Controller*, ser. *Feedback Control, Nonlinear Systems, and Complexity,* (B.A. Francis and A.R. Tannenbaum, eds.). Springer-Verlag, 1995, pp. 196–214.

[34] ——, "The unfalsified control concept and learning," *IEEE Trans. on Automatic Control*, vol. 42, no. 6, pp. 843–847, June 1997.

[35] M. Jun and M. Safonov, "Automatic PID tuning: An application of unfalsified control," in *Proc. IEEE CCA/CACSD*, 1999, pp. 328–333.

[36] J. Shamma, "Analysis and design of gain scheduled control systems," Ph.D. dissertation, Dep. Mech. Eng., M.I.T., Cambridge, 1988.

[37] J. Shamma and M. Athans, "Analysis of gain scheduled control for nonlinear plants," *IEEE Trans. on Automatic Control*, vol. 35, no. 8, pp. 898–907, Aug. 1990.

[38] W. Rugh, "Analytical framework for gain scheduling," *IEEE Control Systems Magazine*, vol. 11, no. 1, pp. 79–84, Jan. 1991.

[39] R. Nichols *et al.*, "Gain scheduling for $\mathcal{H}_\infty$ controllers: A flight control example," *IEEE Trans. on Control Systems Technology*, vol. 1, no. 2, pp. 69–79, June 1993.

[40] D. Stilwell and W. Rugh, "Interpolation of observer state feedback controllers for gain scheduling," *IEEE Trans. on Automatic Control*, vol. 44, no. 6, pp. 1225–1229, June 1999.

[41] F. Blanchini, "The gain scheduling and the robust state feedback stabilization problems," *IEEE Trans. on Automatic Control*, vol. 45, no. 11, pp. 2061–2070, Nov. 2000.

[42] S. Fekri, M. Athans, and A. Pascoal, "A two-input two-output robust multiple model adaptive control (RMMAC) case study," in *Proc. of the American Control Conf.*, Minneapolis, MN, USA, June 2006.

[43] P.M. Young *et al.*, "Practical computation of the mixed-μ problem," in *Proc. of the American Control Conf.*, Chicago, IL, June 1992, pp. 2190–2194.

[44] ——, "Computing bounds for the mixed-μ problem," *Int. J. of Robust and Nonlinear Control*, vol. 5, pp. 573–590, 1995.

[45] P.M. Young, "Controller design with mixed uncertainties," in *Proc. of the American Control Conf.*, Baltimore, MD, June 1994, pp. 2333–2337.

[46] K. Zhou *et al.*, *Robust and Optimal Control.* NJ, USA: Englewood Cliffs, Prentice-Hall, 1996.

[47] K. Zhou and J.C. Doyle, *Essentials of Robust Control.* NJ, USA: Englewood Cliffs, Prentice-Hall, 1998.

[48] G. Balas, Private communication re mixed-μ software, 2003.

[49] G. Balas *et al.*, *μ-Analysis and Synthesis Toolbox of* MATLAB*, User's Guide.* The MathWorks Inc., Jan. 2001.

[50] S. Skogestad and I. Postlethwaite, *Multivariable Feedback Control – Analysis and Design*, 2nd ed. John Wiley and Sons, 2005.

[51] R.L. Kosut and B.D.O. Anderson, "Adaptive control via finite modeling and robust control," *IFAC Workshop on Robust Adaptive Control*, pp. 91–95, Aug. 1988.

[52] D. Magill, "Optimal adaptive estimation of sampled stochastic processes," *IEEE Trans. on Automatic Control*, vol. 10, no. 4, pp. 434–439, Oct. 1965.

[53] D. Lainiotis, "Optimal adaptive estimation: Structure and parameter adaptation," *IEEE Trans. on Automatic Control*, vol. 16, no. 2, pp. 160–170, Apr. 1971.

[54] M. Athans and C.B. Chang, "Adaptive estimation and parameter identification using multiple model estimation algorithms," MIT Lincoln Lab., Lexington, MA, USA, Tech. Rep. 28, June 1976.

[55] R. Hawkes and J. Moore, "Performance of bayesian parameter estimators for linear signal models," *IEEE Trans. on Automatic Control*, vol. 21, no. 4, pp. 523–527, Aug. 1976.

[56] C.B. Chang and M. Athans, "State estimation for discrete systems with switching parameters," *IEEE Trans. on Aerospace and Electronic System*, vol. 14, no. 3, pp. 418–425, May 1978.

[57] B.D.O. Anderson and J.B. Moore, *Optimal Filtering.* NJ, USA: Englewood Cliffs, Prentice-Hall, 1979.

[58] H. Sorenson and D. Alspach, "Recursive baysian estimation using gaussian sums," *Automatica*, vol. 7, pp. 465–479, July 1971.

[59] D. Alspach and H. Sorenson, "Nonlinear baysian estimation using gaussian sum approximations," *IEEE Trans. on Automatic Control*, vol. AC-17, pp. 439–448, Aug. 1972.

[60] A. Gelb, *Applied Optimal Estimation.* MA, USA: MIT Press, 1974.

[61] Y. Baram, "Information, consistent estimation and dynamic system identification," Ph.D. dissertation, MIT, Cambridge, MA, USA, Nov. 1976.

[62] Y. Baram and N.R. Sandell, "An information theoretic approach to dynamical systems modeling and identification," *IEEE Trans. on Automatic Control*, vol. 23, no. 1, pp. 61–66, Feb. 1978.

[63] ——, "Consistent estimation on finite parameter sets with application to linear systems identification," *IEEE Trans. on Automatic Control*, vol. 23, no. 3, pp. 451–454, June 1978.

[64] H. Kwakernaak and R. Sivan, *Linear Optimal Control Systems.* NY, USA: John Wiley, 1972.

[65] B.D.O. Anderson and J.B. Moore, *Optimal Control - Linear Quadratic Methods.* NJ, USA: Englewood Cliffs, Prentice-Hall, 1990.

[66] L. Ljung, *On the Consistency of Prediction Error Identification Methods*, ser. *System Identification: Advances and Case Studies,* (R.K. Mehra and D.G. Lainiotis, eds.). New York: Academic Press, 1976, pp. 121–164.

[67] M. Grewal *et al.*, *Global Positioning Systems, Inertial Navigation, and Integration.* John Wiley & Sons, Inc., 2001.

[68] D. Barros, S. Fekri, and M. Athans, "Robust mixed-μ synthesis performance for a mass-spring system with stiffness uncertainty," in *Proc. of the IEEE International Symposium on Intelligent Control, 2005, Mediterranean Conference on Control and Automation*, Limassol, Cyprus, June 2005, pp. 743–748.

[69] J. Vasconcelos *et al.*, "Uncertainty and performance tradeoffs in robust feedback control: A mimo case study," in *Proc. of the IEEE Conf. on Decision and Control*, (submitted), Dec. 2006.

[70] K. Åström, "Limitations on control system performance," *European Journal of Control*, vol. 6, no. 1, pp. 1–19, 2000.

[71] J. Bokor and G. Balas, "Linear parameter varying systems: A geometric theory and applications," in *Preprints 16^{th} IFAC World Congress*, Prague, Czech Republic, July 2005, pp. 69–79.

[72] X.-D. Sun and I. Postlethwaite, "Affine LPV modeling and its use in gain-scheduled helicopter control," *UKACC Int. Conf. on Control (Conf. Publ. No. 455)*, vol. 2, pp. 1504–1509, Sep. 1998.

[73] G.I. Bara *et al.*, "State estimation for affine LPV systems," *Proc. of the IEEE Conf. on Decision and Control*, vol. 5, pp. 4565–4570, 2000.

[74] I. Szszi *et al.*, "LPV detection filter design for a Boeing 747-100/200 aircraft," *AIAA Journal of Guidance, Dynamics and Control*, vol. 18, no. 3, pp. 461–470, 2005.

[75] M. Corradini *et al.*, "Robust stabilization of multivariable uncertain plants via switching control," *IEEE Trans. on Automatic Control*, vol. 49, no. 1, pp. 107– 114, Jan. 2004.

[76] C. Mehendale and K. Grigoriadis, "A new approach to LPV gain-scheduling design and implementation," in *Proc. of the IEEE Conf. on Decision and Control*, vol. 3, Paradise Island, Bahamas, Dec. 2004, pp. 2942–2947.

[77] J. Doob, *Stochastic Processes.* New York: John Wiley & Sons, Inc., 1953.

[78] P. Halmos, *Lectures on Ergodic Theory.* New York: Chelsea Publ. Co., 1956.

[79] R. Gray, *Probability, Random Processes, and Ergodic Properties.* Springer Verlag, 2001.

[80] U. Grenander, *Stochastic Processes and Statistical Inference.* Arkiv für Matematik, 1950, vol. 1.

[81] K. Yared, "Maximum likelihood identification of state space models for linear dynamic systems," Ph.D. dissertation, MIT, Cambridge, MA, USA, 1979.