

ISSUES ON ROBUST ADAPTIVE FEEDBACK CONTROL

Michael Athans^{*,1}
Sajjad Fekri^{*,2}
Antonio Pascoal^{*}

** Institute for Systems and Robotics,
Instituto Superior Técnico, Lisbon, Portugal
email: {athans,sfekri,antonio@isr.ist.utl.pt}
Tel: (+351) 218 418 054*

Abstract: We discuss recent progress in the field of robust adaptive control with special emphasis on methodologies that use multiple-model architectures. We emphasize that the selection of the number of models, estimators and compensators in such architectures must be based on precise definition of the robust performance requirements. We illustrate some of the concepts and outstanding issues for a new methodology that blends robust nonadaptive mixed μ -synthesis and stochastic hypothesis-testing concepts leading to the so-called RMMAC architecture. A numerical example is used to illustrate the RMMAC design methodology.

Copyright © 2005 IFAC

Keywords: 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 be very promising, but still require a great deal of theoretical and pragmatic research to arrive at the “Holy Grail” of adaptive control. Thus, *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 greater complexity than that of the best possible non-adaptive controller. The word “robust” refers to the desire that the adaptive control system remains stable and *also meets the posed performance specifica-*

¹ M. Athans is also Professor (emeritus), Dept. of EECS, MIT, Cambridge, MA, USA.

² This work was supported in part by the Portuguese FCT POSI programme under framework QCA III. The second author benefited from PhD grant Ref. SFRH/BD/6199/2001 of the FCT (Portuguese Foundation for Science and Technology).

tions for all-possible “legal” parameter values and unmodeled dynamics.

In order to place our remarks in a proper perspective 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 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 discussion 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* uncertain real parameters. 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 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 for time-varying parameters as well.

The current accepted concept of a robust (non-adaptive) 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. At present,

robust synthesis can not deal with slowly-varying parameters, from a theoretical perspective.

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 (Landau, 1979; Narendra and Annaswamy, 1988; Sastry and Bodson, 1989; Åström and Wittenmark, 1995; Ioannou and Sun, 1996). In the MRAC method the emphasis was on proving global convergence to the uncertain real parameter and while using deterministic Lyapunov (hyperstability) arguments for inferring closed-loop stability. However, the assumptions required for stability and convergence 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 (Rohrs *et al.*, 1985). 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 problem involved *multiple-model techniques* which, in principle, are 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 one of the multiple-model approaches, the identification of the most likely model is carried out

by a “supervisor” which switches different controllers, based primarily on deterministic concepts (Morse, 1996, 1997, 1998, 2004; Narendra and Balakrishnan, 1997; Anderson *et al.*, 2000, 2001; Hespanha *et al.*, 2001). 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 was more likely. In the latter approach the controllers could be designed either by classical LQG methods (Willner, 1973; Greene, 1975; Athans *et al.*, 1977; Shomber, 1980; Greene and Willsky, 1980; Schiller and Maybeck, 1997; Maybeck and Griffin, 1997; Schott and Bequette, 1997) or by more sophisticated methods (Fekri *et al.*, 2004*a,b,c*; Fekri, 2005). In the latter approach one can deal with MIMO designs. A different, more direct approach, called *unfalsified control*, also merits mention (Safonov and Tsao, 1995, 1997; Jun and Safonov, 1999); although we shall not discuss it in this paper. We also 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 is.

1.2 Our Design Philosophy

In this paper we shall focus upon “robust performance” requirements on the adaptive system implemented by one of the available multiple-model methods. We follow the recent research on “Robust Multiple-Model Adaptive Control (RM-MAC),” (Fekri *et al.*, 2004*a,b,c*; Fekri, 2005), which will be discussed in 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 Doyle *et al.*, is often called the *mixed- μ* design method. 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 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 (Kosut and Anderson, 1988; Anderson *et al.*, 2000; Fekri *et al.*, 2004*a,b,c*; Fekri, 2005).

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 can be bounded *a priori* only in the frequency domain. Therefore, the adaptive design must *explicitly* reflect these frequency-domain bounds upon 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 (Young *et al.*, 1992, 1995; Young, 1994; Zhou *et al.*, 1996; Zhou, 1998; Balas, 2003; Balas *et al.*, 2004); and associated MATLAB software (Balas, 2003; Balas *et al.*, 2004).

(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 emphasise 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 the best “*global non-adaptive robust compensator (GNARC)*” for the entire (large) uncertain real-parameter set and 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 then provides a yardstick (lower-bound) for performance, so that any performance improvements by more complex adaptive designs can be quantified. It is self-evident that the GNARC must be 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 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 from 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 upon 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. 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. 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 present numerical results, using the RMMAC, for a non-trivial dynamic control problem. Section 6 discusses the commonalties and differences of

the supervisory *switching multiple model adaptive control (SMMAC)* and *RMMAC architectures* and we summarize our conclusions in Section 7.

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. Finally, we discuss the architecture associated with *robust multiple-model adaptive control (RMMAC)*.

2.1 Multiple-Model Adaptive Estimation (MMAE)

One of the first uses of multiple-models was motivated by the need for accurate stochastic state-estimation for dynamic systems subject to significant parameter uncertainty. In such problems the estimation accuracy provided by standard Kalman filters was not adequate. For some early references regarding MMAE see (Magill, 1965; Lainiotis, 1971; Athans and Chang, 1976; Hawkes and Moore, 1976; Chang and Athans, 1978). We remark that MMAE algorithm was also referred to as “partitioned estimation”, especially in the research by Lainiotis.

Fig. 1 shows the architecture of the MMAE system. It is assumed that a 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 noise. If there is no parameter uncertainty in the plant, then the Kalman filter (KF) is the optimal state-estimation algorithm; see, for example, (Gelb, 1974; Anderson and Moore, 1979). Moreover, under the usual linear-gaussian assumptions, the KF state-estimate is the true conditional mean of the state. If the plant has an uncertain real-parameter vector, p , one can imagine that it is “close” to one of a *finite* discrete parameter set, $P_D = \{p_1, p_2, \dots, p_N\}$. 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, $k=1, 2, \dots, N$. It turns out that, if indeed the true plant parameter is identical to one of its discrete values – and this is modeled as 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*

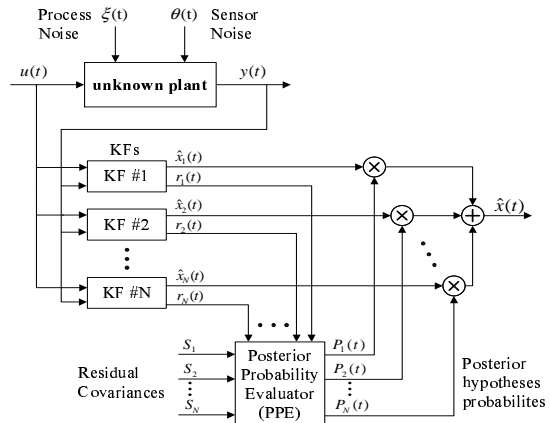


Fig. 1. The MMAE architecture for adaptive estimation.

mean of the state and one can calculate the *true conditional covariance matrix*; see, for example, (Athans and Chang, 1976); (Anderson and Moore, 1979, Chap. 10).

Appendix I summarizes the notation and formulas associated with the MMAE algorithm

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 , $k = 1, 2, \dots, N$, 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. $P_k(t) = \text{Prob}\{H = H_k | Y(t)\}$, $k = 1, 2, \dots, N$. 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 , $P_j(t) \rightarrow 1$, 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 $t \rightarrow \infty$. 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)”, (see Baram, 1976; Baram and Sandell, 1978*b,a*), as discussed in Appendix II or the Kullback information metric (see Anderson and Moore, 1979, 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 (Anderson and Moore, 1979) 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 employ several sensors, 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 the classical MMAC algorithms which simply integrated design concepts from Linear-Quadratic-Gaussian (LQG) control system design (Kwakernaak and Sivan, 1972; Anderson and Moore, 1990) with the MMAE architecture. The classical MMAC architecture (CMMAC) is shown in Fig. 2. Each “local” state-estimate of the (steady-state) KF, $\hat{x}_k(t|t)$, $k = 1, 2, \dots, N$, is multiplied by the associated linear-quadratic (LQ) optimal control-gain matrix, G_k , to generate a “local” control signal $u_k(t)$ by

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

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

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

The CMMAC algorithm of Fig. 2 represented a purely *ad-hoc* approach to adaptive control. *It is not optimal* in a stochastic LQG sense (Willner, 1973). However, several adaptations, extensions and simulations have been reported (Greene,

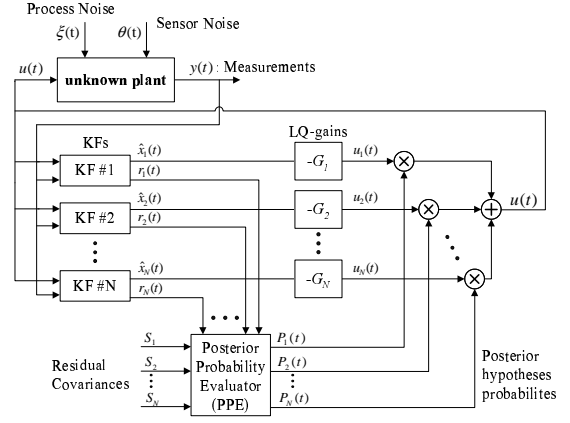


Fig. 2. The CMMAC algorithm where the “local” KF state estimates get multiplied by “local” LQ gains to form “local” controls which, in turn, generate a probabilistically weighted “global” control.

1975; Shomber, 1980; Athans *et al.*, 1977; Greene and Willsky, 1980; Schiller and Maybeck, 1997; Maybeck and Griffin, 1997; Schott and Bequette, 1997). 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 widely adopted at present.

It is also noteworthy to mention that a “switching” version of the CMMAC can be 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. 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, more deterministic, approach to adaptive control using multiple models has been initiated (Morse, 1996, 1997, 1998, 2004; Anderson *et al.*, 2000, 2001; Hespanha *et al.*, 2001) and research along these lines is still in progress. The methodology is called *Supervisory Switching Multiple-Model Adaptive Control (SMMAC)*. The architecture of the SMMAC algorithms is shown in Fig. 3 (which is an

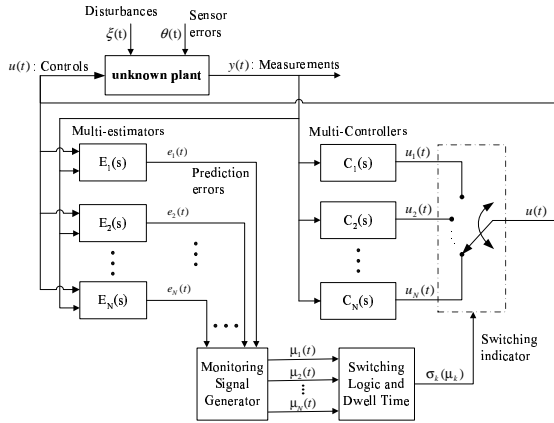


Fig. 3. The SMMAC architecture.

adaptation of Fig. 1 in (Hespanha *et al.*, 2001), so that comparisons with the CMMAC and RMMAC become easier). We remark that in SMMAC architectures there exists a “separation” between identification and control (unlike the CMMAC).

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. 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 CMMAC and RMMAC. The presence of unmodeled dynamics is also considered, although the bound on the unmodeled dynamics is simply an H_∞ 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.

In reference to Fig. 3, the SMMAC employs a finite number of stable deterministic estimators (Luenberger observers), called *multi-estimators* and denoted by $E_k(s)$, $k = 1, 2, \dots, N$, designed for a grid of distinct 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 $e_k(t) = y_k(t) - y(t)$, $k = 1, 2, \dots, N$. 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*, $C_k(s)$, $k = 1, 2, \dots, N$, has been designed so that each provides *satisfactory*

stable feedback performance for at least one discrete parameter $p_k \in P_D$. The basic idea is to use the monitoring signals $\mu_k(t)$ to “switch-in” the suitable controller. This is accomplished 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 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 CMMAC that 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 present the newest multiple-model architecture which we call RMMAC, to emphasize the fact that both stability-robustness and performance-robustness are addressed from the start. Preliminary results on RMMAC can be found in (Fekri *et al.*, 2004b,c) and a more complete treatment is available in Fekri’s Ph.D. thesis (Fekri, 2005). We note that the RMMAC architecture has a “separation” between identification and control, like the SMMAC and unlike the CMMAC.

The RMMAC architecture is shown in Fig. 4. It represents a blend of the CMMAC, Fig. 2, and of the SMMAC, Fig. 3.

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. *However, unlike CMMAC the “local” KF state-estimates (the $\hat{x}_k(t|t)$ in Fig. 2) are not used in gener-*

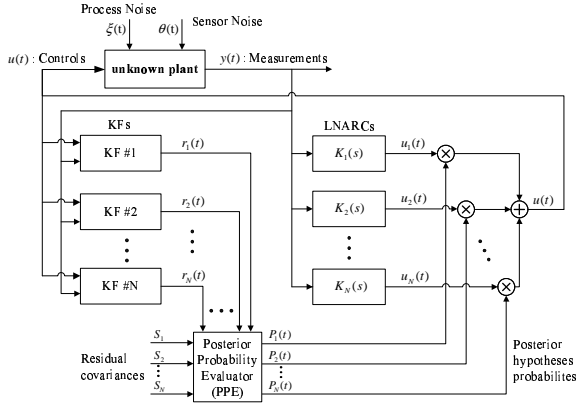


Fig. 4. The RMMAC architecture.

ating the control signals. Only the KF residuals, $r_k(t)$, $k = 1, 2, \dots, N$, 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 of the posterior probabilities is identical to that in the MMAE or CMMAC (see Appendix I). A crucial difference is that the “nominal” KF design uses explicitly 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, $K_j(s)$, $j = 1, 2, \dots, N$, in Fig. 4, which are designed using the state-of-the-art in robust mixed- μ synthesis (Young *et al.*, 1992; Young, 1994; Young *et al.*, 1995; Zhou *et al.*, 1996; Balas *et al.*, 2004; Balas, 2003). These compensators, $K_j(s)$; $j = 1, 2, \dots, N$, referred to as the “local non-adaptive robust compensators (LNARC)”, are designed so as to guarantee “local” stability- and performance-robustness. 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 $P_j(t)$

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

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” can 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) (Baram, 1976; Baram and Sandell, 1978*b,a*), to ensure that the RMMAC operates in a superior manner, so as to ensure correct asymp-

totic 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.

2.5 Discussion

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

- One must design a set of N multi-controllers or dynamic compensators.
- One must design a set of N Kalman filters or multi-estimators (observers).
- 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 systematic procedure by which, starting from a compact parameter set $p \in P$, to define a finite set, N , of discrete values (models), $P_D = \{p_1, p_2, \dots, p_N\} \subset P$, 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.

We conclude this section with a few comments. 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. Even in the simpler CMMAC, involving LQG controllers, attempts to prove global stability were not successful (Greene, 1975; Shomber, 1980; Greene and Willsky, 1980). 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

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 H_∞ 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 H_∞ 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 H_∞ 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. A recent application (Barros *et al.*, 2005) illustrates this uncertainty/performance tradeoff in a clear manner.

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 to 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 straight forward, but complicates the exposition. What we want to stress relates to the 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

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

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, for each independent real uncertain parameter we must have a lower- and upper-bound (real-valued).

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

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

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 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. The GNARC is computed via mixed- μ synthesis and it takes into account the frequency-domain bounds on unmodeled dynamics, and 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 to some initial value and exercises the MATLAB software which, after a sequence of so-called *DG-K* iterations, determines a compensator $K(s)$ and generates an upper-bound $\mu_{ub}(\omega)$. If

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

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*. It should be evident, that in order to find the “best” GNARC we must maximize the performance-parameter A_p until the μ -upper bound is just below unity, say $\mu_{ub}(\omega) \approx 0.995 \quad \forall \omega$.

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 1.

We believe that such a thorough understanding of the non-adaptive GNARC is essential prior to making a 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,

Table 1.
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.

become acceptable if more controls and/or sensors are introduced 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 will be used to 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, $N \rightarrow \infty$, in any multiple-model adaptive scheme. Thus, we understand *what is the best possible performance if we knew the real parameter exactly*.

In practice, to determine the FNARC one uses a dense grid of parameters p_j , $j \rightarrow \infty$, 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 (Balas *et al.*, 2004), 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, $\mu_{ub}^c(\omega)$ is just below unity for all frequencies, say $\mu_{ub}^c(\omega) \approx 0.995 \quad \forall \omega$, to be consistent with the GNARC upper-bound employed.

One can then analyze each FNARC design using the six techniques outlined in Table 1. Our experience indicates that detailed analyses, especially at the corners of the hyper-parallelepiped P_π , provide useful insights regarding the impact of the subset of the uncertain real parameters that determine the need for sophisticated adaptive control.

3.4 The Potential Benefit of Adaptive Control

Recall that we had initially posed the following question: *do we need adaptive control?* The GNARC and FNARC results provide the designer with the tool 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, at worse, 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 GNARC provides the lower-bound upon expected performance, while the (infinite-model) adaptive FNARC provides the performance upper-bound. This information is critical in deciding whether to implement an adaptive control system.

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 parameter was near its upper-bound, i.e. $p \approx p_U$.

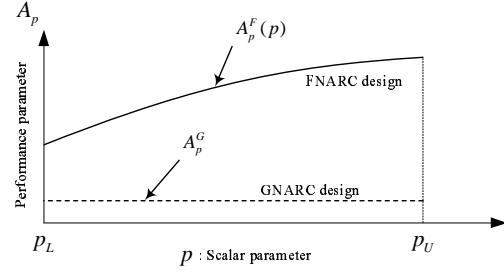


Fig. 5. Hypothetical comparison of the performance parameter, A_p , for the GNARC and FNARC for the case of a scalar uncertain real parameter, p , $p_L \leq p \leq p_U$.

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. (Åström, 2000)). Such a non-minimum phase system has been analyzed, using the RMMAC, in (Fekri *et al.*, 2004a).

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 numerous to construct the two hypersurfaces along the lines suggested by Fig. 5. However, the same philosophy still applies.

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. 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.

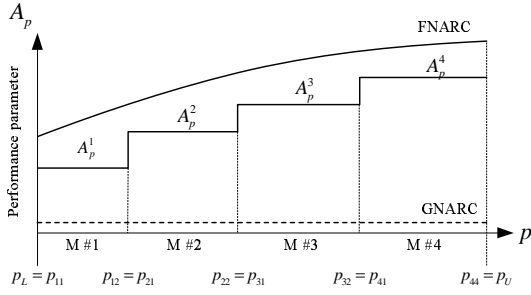


Fig. 6. Illustration of a “brute-force” selection of four models. An *ad-hoc* decision is made on the number of models, $N=4$, and the specification of the “boundary”, $[p_{kL}, p_{kU}]$, $k = 1, 2, 3, 4$, for each model $M\#k$. The A_p^k denote the maximized value of the performance parameter.

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

The model upper and lower bounds must be found by trial-and-error.

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, we obtain 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 , $k=1,2,3,4$, plotted 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 1.

The RMMAC numerical results presented in (Fekri *et al.*, 2004a,c) 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, although this was not done in their numerical simulations.

3.5.2. A More Systematic Approach to Model Selection and Definition: The % FNARC method.

The numerical simulations for the RMMAC design in Section 5 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 $\Omega_\alpha = \{p : \alpha \leq p \leq p_U\}$. 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 $A_p^{\alpha \max}$. As long as $A_p^{\alpha \max} > (X\%) \cdot A_p^F(\alpha)$, 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 $A_p^{\alpha^* \max} = (X\%) \cdot A_p^F(\alpha^*)$. 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 $\Omega_1 = \{p : \alpha^* \leq p \leq p_U\}$; 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 $\Omega_\beta = \{p : \beta \leq p \leq \alpha^*\}$. For each value of β , we use the mixed- μ software to design the best robust controller by maximizing the performance parameter A_p denoted by $A_p^{\beta \max}$. As long as $A_p^{\beta \max} > (X\%) \cdot A_p^F(\beta)$, where $A_p^F(\beta)$ is the parameter value of the FNARC at $p = \beta$, then β is decreased until at $\beta = \beta^*$ we have $A_p^{\beta^* \max} = (X\%) \cdot A_p^F(\beta^*)$. 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

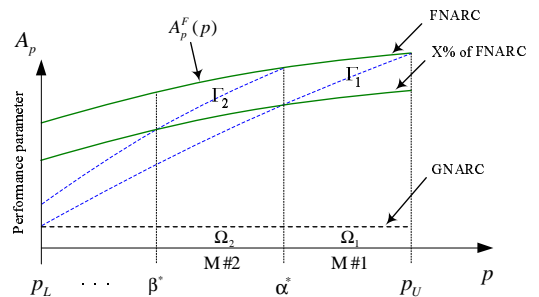


Fig. 7. Visualization of the % FNARC model definition process.

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 $\Gamma_1, \Gamma_2, \dots$ are also surfaces). Unfortunately, the intersection of these surfaces *does not* occur along rectangular (or parallelepiped) parameter subsets. However, the mixed- μ software requires rectangular (or parallelepiped) constraints for uncertain parameter sets. Suggested modifications to the % FNARC concept are discussed in (Fekri, 2005).

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 KALMAN FILTERS

4.1 Introduction

In this section we discuss 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 (Baram, 1976; Baram and Sandell, 1978*b,a*) which will imply that the

PPE will lead to the correct model identification. 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 must be 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 $\Omega_k, k = 1, 2, \dots, N$, such that

$$\bigcup_{k=1}^N \Omega_k = P_\pi \quad (4.1)$$

and each Ω_k defines the 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.

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 terms of the convergence of the posterior probabilities. The correct way is to select the p_k^* using the Baram proximity measure.

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 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 a “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 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 $P_1(t) \rightarrow 1, P_2(t) \rightarrow 0$, so that we have convergence to Model #1 (defined by KF #1) occurs, even though the Euclidean distances would indicate the opposite ($|p_A - p_1^*| > |p_A - p_2^*|$).

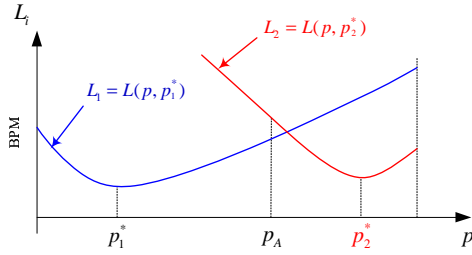


Fig. 8. Illustration of Baram proximity measures (BPM).

Fundamental convergence result: In (Baram, 1976) this result was generalized and proved for an arbitrary number of KFs designed for the nominal values $p_1^*, p_2^*, \dots, p_N^*$. If the BPM satisfies the inequality

$$L(p, p_j^*) < L(p, p_k^*) \quad \forall k \neq j = 1, 2, \dots, N \quad (4.2)$$

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

$$\lim_{t \rightarrow \infty} P_j(t) \rightarrow 1 \quad (4.3)$$

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 $\Omega_1, \Omega_2, \dots, \Omega_N$ which define the models $M\#1, M\#2, \dots, M\#N$. The question now is: *how do we select the nominal values $p_1^*, p_2^*, \dots, p_N^*$ 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 $p \in \Omega_j \Rightarrow P_j(t) \rightarrow 1$ *a.s.* This, is the method we use in the numerical RMMAC simulations of Section 5.

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 nominal KF optimization design points in an optimal manner (Fekri, 2005).

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 must “robustify” the KF to be tolerant of such “errors”. The time-honored engineering practice of using suitable “fake plant white noise” is a very useful tool. We have found (Fekri, 2005) that judicious use of “fake plant white noise” (which causes the KF gains to increase and pay more attention to the measurements) 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 parameter changes “slowly” as a function of time.

It is also important to note that similar probability convergence results for the MMAE can be found in (Anderson and Moore, 1979, 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 studies the deep relationship of the BPM and the Kullback metric, especially with respect to posterior probability convergence behavior.

5. 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.

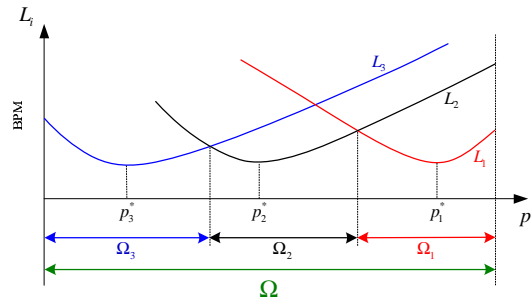


Fig. 9. Optimizing the KF nominal design points using the BPMs.

In this section we test and evaluate the disturbance-rejection performance of the RMMAC feedback system as compared to the “best” non-adaptive design obtained via mixed- μ robust synthesis. We follow the step-by-step design methodology of Section 3 and 4.

We present the outcome of several Monte Carlo (MC) simulations, which demonstrate the actual 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 robust non-adaptive one, thereby confirming the positive aspects of this adaptive control method.

5.1 The Two-Cart Example Dynamics

The RMMAC was tested and evaluated using the two-cart mass-spring-damper (MSD) system, shown in Fig. 10. A different topology of the two-cart system was analysed by RMMAC in (Fekri *et al.*, 2004c). The system in Fig. 10 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 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, as follows:

$$d(s) = \underbrace{\frac{\alpha}{s + \alpha}}_{W_d(s)} \xi(s) \quad (5.1)$$

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

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) + L\xi(t) \\ y(t) &= Cx(t) + \theta(t) \end{aligned} \quad (5.2)$$

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

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

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

$$L^T = [0 \ 0 \ 0 \ 0 \ \alpha]$$

The following parameters in (5.2) are fixed and known:

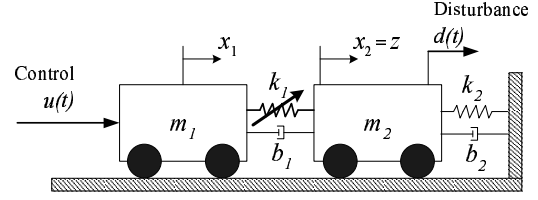


Fig. 10. The two-cart system. The spring-constant k_1 is uncertain.

$$m_1 = m_2 = 1, \ k_2 = 0.15, \ b_1 = b_2 = 0.1, \ \alpha = 0.1 \quad (5.3)$$

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

$$\Omega = \{k_1 : 0.25 \leq k_1 \leq 1.75\} \quad (5.4)$$

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

$$z(t) \equiv x_2(t) \quad (5.5)$$

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

$$\begin{aligned} y(t) &\equiv x_2(t) + \theta(t) \\ E\{\theta(t)\} &= 0, \ E\{\theta(t)\theta(\tau)\} = 10^{-6} \delta(t - \tau) \end{aligned} \quad (5.6)$$

The desired disturbance-rejection requires that the effects of $d(t)$ and $\theta(t)$ be minimized 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 through 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.

$$\tau \leq 0.05 \text{ sec} \quad (5.7)$$

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

$$W_{unmod}(s) = \frac{2.1s}{s + 40} \quad (5.8)$$

5.2 Designing 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 parameter uncertainty of (5.4). 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 H_2 or H_∞ designs, we use the following frequency domain weights on the control and measurement noise,

$$\text{Control weight: } W_u(s) = \frac{10(s+10)}{s+10^3} \quad (5.9)$$

$$\text{Meas. noise weight: } W_n = 10^{-3} \text{ (constant)}$$

These, together with the unmodeled dynamics weight (5.8), limit the bandwidth of the closed-loop system by penalizing large high-frequency control signals. The weights (5.8) and (5.9) are incorporated in the definition of the *nominal generalized plant*, together with (5.2) with $k_1 = 1.0$. The weights (5.1), (5.8), and (5.9) will not change in any of the subsequent designs.

To carry out the mixed- μ synthesis, the parameter uncertainty of (5.4) is represented by

$$0.25 \leq k_1 \leq 1.75 \Rightarrow k_1 = 1.0 + 0.75 \delta_{k_1}; \quad |\delta_{k_1}| \leq 1 \quad (5.10)$$

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

$$W_p(s) = A_p \frac{0.1}{s+0.1} \quad (5.11)$$

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. Notice that performance weight $W_p(s)$ penalizes 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 limited by the required guarantees on robust-stability and -performance inherent in the mixed- μ synthesis methodology.

Fig. 11 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

$$\tilde{z}(s) = W_p(s) z(s) = A_p \left(\frac{0.1}{s+0.1} \right) z(s) \quad (5.12)$$

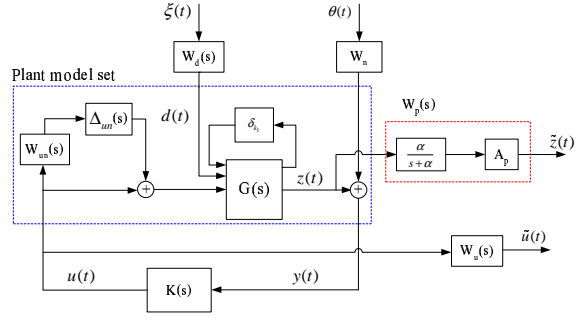


Fig. 11. MSD system with weights for mixed- μ synthesis.

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.

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

$$\tilde{u}(s) = W_u(s) u(s) = 10 \left(\frac{s+10}{s+10^3} \right) u(s) \quad (5.13)$$

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

Using the mixed- μ software the performance parameter A_p in (5.11) is increased as much as possible until the upper-bound on the mixed- μ , $\mu_{ub}(\omega)$, satisfied the inequality

$$\mu_{ub}(\omega) \leq 1, \quad \forall \omega \quad (5.14)$$

which is only a *sufficient condition* for both stability- and performance-robustness. The largest value of the performance parameter A_p in (5.11) thus determined was

$$A_p^G = 50.75 \text{ (with } \mu_{ub} \approx 0.995) \quad (5.15)$$

which leads to the GNARC design, the “best” LTI non-adaptive compensator $K(s)$ that guarantees stability- and performance-robustness for the entire parameter interval (5.4). 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.

5.3 Designing Local Non-Adaptive Robust Compensators (LNARCs)

Following the procedure presented in Section 3, the plots of the (optimized) performance parameter A_p for the GNARC design and the FNARC designs (requiring an infinite number of models) is shown in Fig. 12. This figure shows that there is a potential 20-fold improvement in performance by using adaptive control.

Remark: In this example, the GNARC and FNARC look flat for all values of k_1 . Actually,

Table 2. Best GNARC and LNARCs

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”

the FNARC get 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 (Fekri, 2005).

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

Fig. 13 shows how four covering models are selected to be adequate for constructing the RM-MAC using performance level of 70%.

As illustrated in Fig. 13 for the specified performance level of 70%, the large parameter uncertainty interval of (5.4) is subdivided into four subintervals as summarized in Table 2.

Explanation. We now provide a bit more detail on how the four models of Fig. 13 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 to calculate the maximum performance parameter A_p^b which results in the dashed curve, labeled Γ_1 in Fig. 13. 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. 13. When the curve Γ_2 intersects the 70% FNARC curve we stop. In this example,

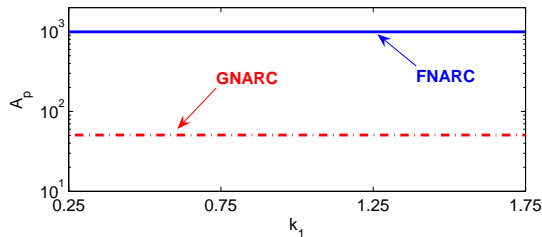


Fig. 12. Best GNARC and FNARC performance gains for spring-stiffness uncertainty subintervals, as in (5.4).

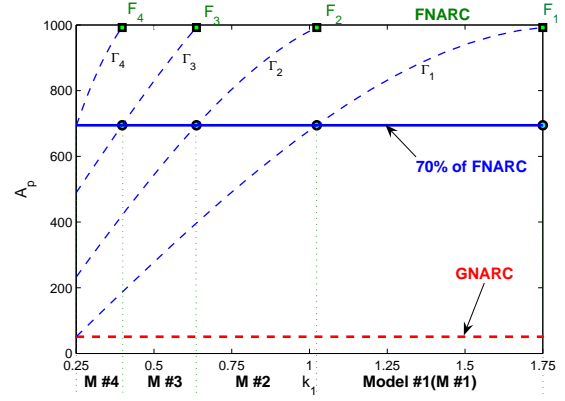


Fig. 13. Best local performance gains using mixed- μ using 70% of the best FNARC performance designed for local uncertainty bounds.

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. 13 and the four models summarized in Table 2.

As a result, the whole initial uncertainty set of (5.4) is covered by four models using 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 2.

In fact the above model selection procedure using mixed- μ synthesis also generates four “local” robust compensators, $K_1(s), \dots, K_4(s)$, designed for each subinterval defined in Table 2; these are referred to as LNARCs. In the mixed- μ synthesis, the weights (5.1), (5.8) and (5.9) were the same as in the GNARC design of Section 5.2. However, for each LNARC design, the performance parameter A_p in (5.11) is increased until the mixed- μ bound of (5.14) is achieved, and their optimized values are shown in the last column of Table 2.

Fig. 14 compares the GNARC design with the four LNARCs, namely $K_1(s), \dots, K_4(s)$ by looking at their Bode magnitude plots.

Note 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 discussed below.

We emphasize that each individual LNARC closed-loop design has guaranteed performance- and stability-robustness over its associated parameter subinterval of Table 2.

Remark: In the design philosophy adopted in this paper we have stressed that the adaptive controller complexity, as measured by the number of models in the CMMAC, SMMAC and RMMAC should be the natural by-product of the perfor-

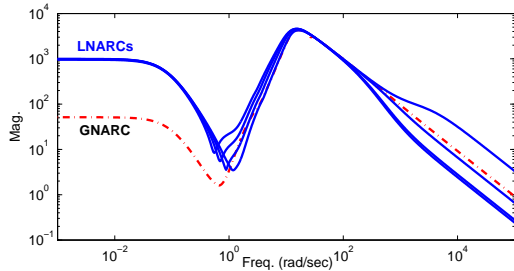


Fig. 14. Frequency-domain characteristics (Bode plot) of the GNARC and the four LNARCs.

mance 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 2.

If we are willing to have somewhat inferior performance and select, say, $X=50\%$ then the procedure outlined results in only two models. If we wish to have much better performance and select, say, $X=90\%$, then the outlined procedure yields nine models. Clearly, as we demand better and better performance we must increase the controller complexity, and this agrees with engineering intuition.

5.4 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 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, $K_1(s)$, \dots , $K_4(s)$, in addition to the calculation of the four posterior probabilities, $P_1(s)$, \dots , $P_4(s)$, 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 a specific LNARC is used. *After the*

probability convergence, the RMMAC essentially operates as an LTI stochastic feedback system!

In the spirit of Table 1, this allows us to calculate two key transfer functions for disturbance-rejection and control signal characteristics

Disturbance - rejection:

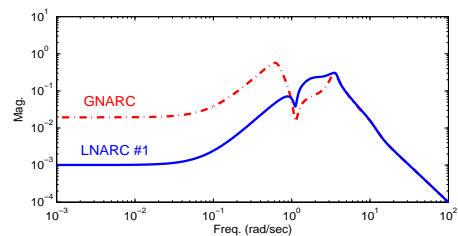
$$M_{\xi z}(s) \equiv \frac{z(s)}{\xi(s)} \quad \text{or} \quad M_{dz}(s) \equiv \frac{z(s)}{d(s)} \quad (5.16)$$

Control - signal:

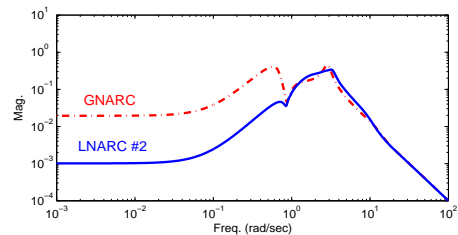
$$M_{\xi u}(s) \equiv \frac{u(s)}{\xi(s)} \quad \text{or} \quad M_{du}(s) \equiv \frac{u(s)}{d(s)}$$

for different values of the unknown spring stiffness of (5.4), for the GNARC and for each LNARC design.

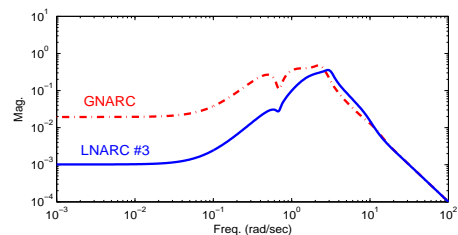
Fig. 15 illustrates the above using the actual spring constants indicated in (5.17) quantifying



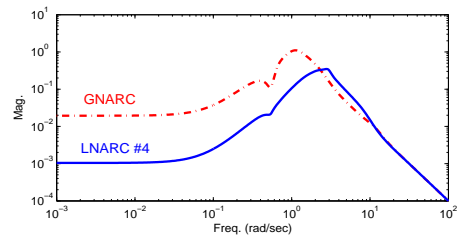
(a) GNARC and LNARC #1 for $k_1 = 1.2$



(b) GNARC and LNARC #2 for $k_1 = 0.76$



(c) GNARC and LNARC #3 for $k_1 = 0.44$



(d) GNARC and LNARC #4 for $k_1 = 0.34$

Fig. 15. Potential improvement from RMMAC visualized by Bode plots of disturbance-rejection transfer function $|M_{\xi z}(j\omega)|$.

the potential RMMAC improvement in disturbance-rejection.

Similar plots can be made for other values of the uncertain spring stiffness. Fig. 15 predicts that the RMMAC has the potential to significantly improve disturbance-rejection; these predictions will be validated in the sequel, on the order of $\frac{1}{A_p}$.

Moreover, other similar plots, such as control signal characteristics, can be made for other values of the uncertain spring constant. Other transfer functions could also be computed (not shown) from disturbance $\xi(t)$ and the sensor noise $\theta(t)$ to the control $u(t)$.

Fig. 16 evaluates the potential performance improvement of using RMMAC by using stochastic metrics, namely by comparing the 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. The graphs of Fig. 16 vividly suggest that RMMAC has the potential of decreasing the output RMS by a factor of 2–5 over the GNARC system. Note that the potential improvement in output RMS requires controls with higher RMS values, as expected.

Finally, it is important to construct what we call the “stability-mismatch” table shown in Table 3.

The interpretation of Table 3 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 # i ?* The diagonal entries in this table are always robustly-stable, by construction. Examining the first row in Table 3 it is observed that for Model #1, i.e. for all $k_1 \in [1.20, 1.75]$, if

Table 3. Mismatched-Model 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 \equiv always stable
U \equiv always unstable
CU \equiv conditionally unstable

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 (5.14) is “only a sufficient condition for robust stability” 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 LNARC #4 for all $k_1 \in [0.25, 0.73]$. Of course, performance-robustness is only guaranteed for the subintervals defined in Table 2.

5.5 Designing the MMAE and RMMAC System

For this example we follow the process 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 is how to select the nominal value of the uncertain spring stiffness, k_1 , denoted by k_{1i}^* , $i = 1, 2, 3, 4$, 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 2.* Rather, as explained in Section 4, the KF design points, k_{1i}^* , $i = 1, 2, 3, 4$, must be determined by an optimization process so that the Baram proximity measures (BPMs) agree at the boundaries of the sets Ω_j . The outcome of this optimization process is shown in Fig. 17 using the optimized KF nominal values.

In Fig. 17 the curves show the BPM for any value $0.25 \leq k_1 \leq 1.75$ from each of the four optimized KFs. This process is required so that for any $k_1 \in \Omega_j$, the corresponding posterior probability $P_j \rightarrow 1$, $P_k \rightarrow 0 \forall k \neq j$. The specific nominal KF numerical values for this example are obtained by iteration and are

$$\mathcal{K}_1^* = [1.2 \ 0.76 \ 0.45 \ 0.34] \quad (5.17)$$

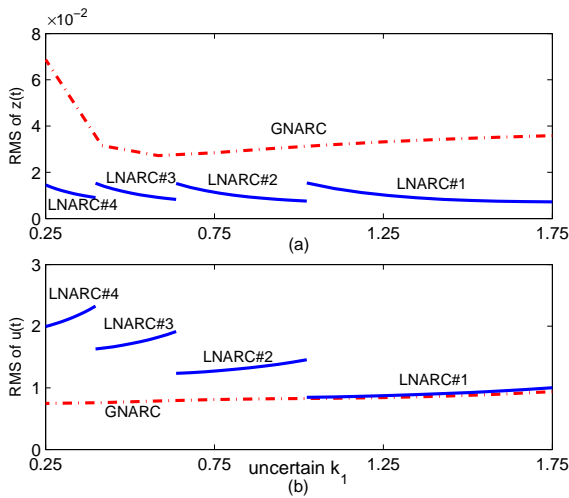


Fig. 16. Predicted potential RMS performance of the RMMAC vs GNARC. There is no sensor noise for these plots. (a): Output RMS comparisons from $\xi(t)$ to $z(t)$. (b): Control RMS comparisons from $\xi(t)$ to $u(t)$.

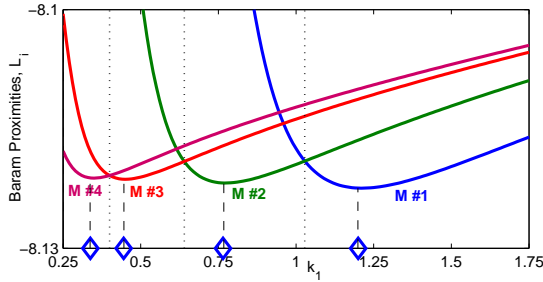


Fig. 17. Baram proximity measures (BPM) for the optimized four KF.

where $k_{1i}^* = \mathcal{K}_1^*[i]$ is the nominal spring constant k_1 used in Model # i associated with the KF # i .

The GNARC and LNARC compensators are designed in continuous time, but the simulation was implemented in discrete time using a zero-order hold with a sampling time of $T_s = 0.01$ secs. The KF in the MMAE algorithm were designed in discrete-time using the sampling interval T_s ; all continuous-time dynamics (plant, weights, etc) were transformed to their discrete-time equivalents. In addition, the correct variances of the discrete-time white noise sequences, $\xi(\cdot)$ and $\theta(\cdot)$, were calculated and used to design the four KF and the posterior probability evaluator (PPE); these discrete-time numerical values were used in all Monte Carlo (MC) simulations in the sequel.

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

$$u(t) = \sum_{j=1}^4 P_j(t) u_j(t) \quad (5.18)$$

where the $u_j(t)$ are the “local” controls generated by each LNARC, $K_j(s)$, as designed in Section 5.3.

5.6 RMMAC Stochastic Simulations and Performance Evaluation

Unless stated otherwise:

(a) all simulations use a stochastic disturbance and white measurement noise generated according to (5.1). The true system includes an actual (but unmodeled) time-delay of 0.01 secs in the control channel.

(b) all initial model probabilities are initialized to be $P_k(0) = 0.25$ ($k = 1, \dots, 4$) at $t = 0$ secs.

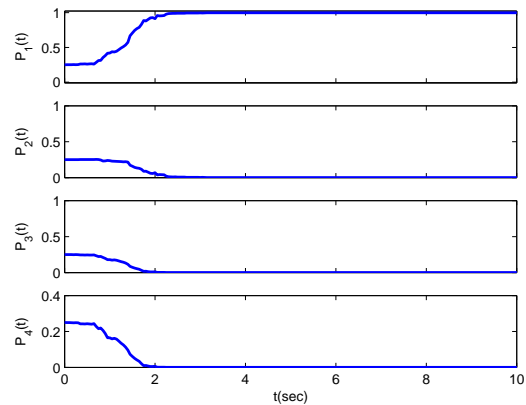
(c) we present numerical averages for 5 MC simulations.

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 (Fekri, 2005).

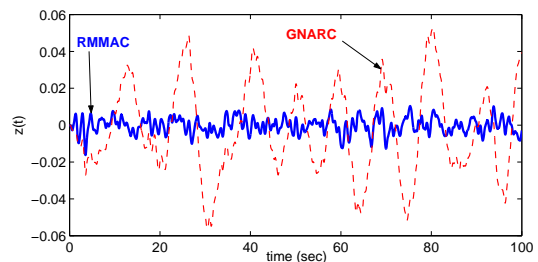
5.6.1. “Easy” Identification, I: The dynamic evolution of the four posterior probabilities when the true $k_1 = 1.65$, well inside the Model #1 subinterval, and the corresponding outputs for the RMMAC and the GNARC systems are shown in Fig. 18. The correct model (Model #1) is identified quickly in about 2 secs. The improvement in disturbance-rejection by the RMMAC is evident as shown in 18(b).

5.6.2. “Easy” Identification, II: Again, 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. 19(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. 19(b).

5.6.3. “Harder” Identification: When the actual spring constant is near the boundary between two models, it takes longer (more data) to resolve

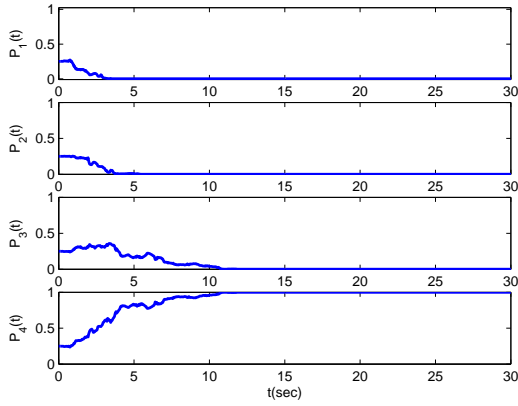


(a) RMMAC Posterior probabilities

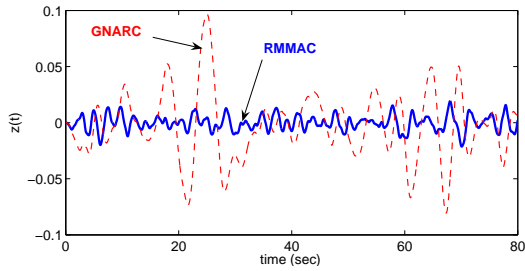


(b) Output performance comparisons

Fig. 18. Simulation results for $k_1 = 1.65$.



(a) RMMAC Posterior probabilities



(b) Output performance comparisons

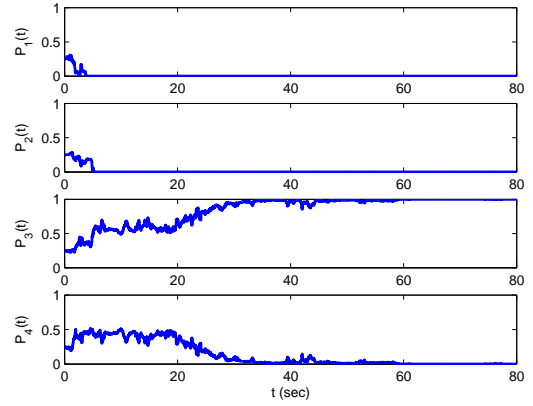
Fig. 19. Simulation results for $k_1 = 0.3$.

the true hypothesis. In this example, $k_1 = 0.405$ is selected which belongs to Model #3 but is also “close” to Model #4, see Fig. 17. The probabilities vs time as well as output comparisons are shown in Fig. 20. It takes about 50 secs to resolve the ambiguity between Models #3 and #4.

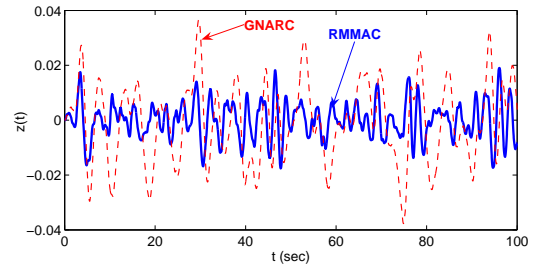
Fig. 20(a) shows that the probabilistic weighing of the control according to (5.18) persists for about 50 secs. However, as evidenced by Fig. 20(b), there is no significant degradation of the RMMAC performance as compared to the GNARC. In Fig. 21 the performance of the feedback system is shown when we fix $P_3(t) = 1, \forall t = 0, 1, 2, \dots$ (i.e. when we have “perfect” identification from the start) with that of the RMMAC and compare the output response. As we can see, the probabilistic averaging results in insignificant performance deterioration.

5.6.4. Initial Mismatch Instability: In Table 3 the mismatch-stability properties of the LNARC designs are summarized. We next evaluate the RMMAC response *when we force it to be unstable at time $t = 0$* .

Fig. 22 illustrates a typical result selected from several different MC simulations. In Fig. 22 the true value of k_1 is 1.75 in Model #1; its nearest probabilistic neighbor is KF #1. From Table 3 we know that if we use LNARC #4, $K_4(s)$, with



(a) RMMAC Posterior probabilities



(b) Output performance comparisons

Fig. 20. Simulation results for $k_1 = 0.405$ (in Model #3).

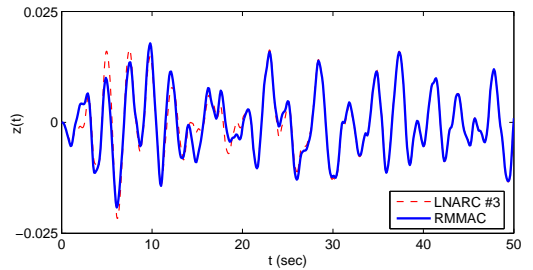
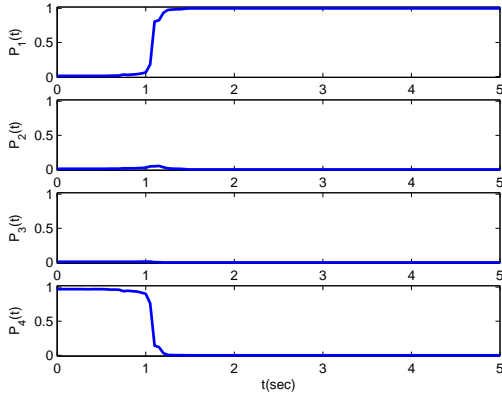


Fig. 21. LNARC #3 and RMMAC performance output comparisons for $k_1 = 0.405$.

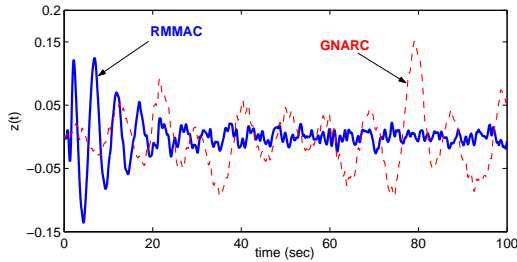
Model #1 we have an unstable closed-loop system. To force this initial instability, the initial values of the probability vector are selected to be

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

so that initially, at least at $t = 0$, the RMMAC system is forced to be unstable. However, as illustrated in Fig. 22, the RMMAC rapidly recovers to a stable configuration. Fig. 22(a) shows that the “correct probability” $P_1(t) \rightarrow 1$ 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. 22(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.



(a) Posterior probabilities transients



(b) Output response

Fig. 22. The RMMAC recovers from an initial unstable configuration at $t = 0$.

Fig. 23 illustrates another mismatch-instability result, similar to the above case with $k_1 = 1.75$. This test was suggested by Prof. B.D.O. Anderson. To force this instability the probabilities vector at time $T = 60$ secs are forced to be

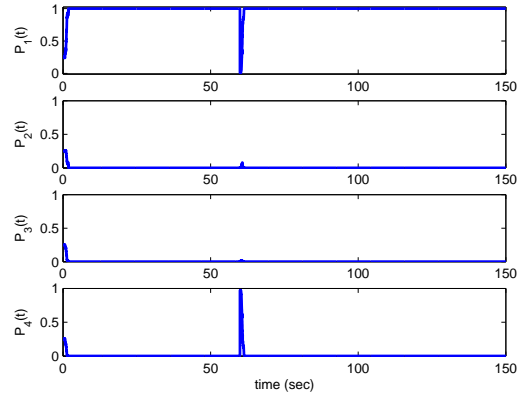
$$P_1(T) = P_2(T) = P_3(T) = 0.01, P_4(T) = 0.97$$

so that the RMMAC system is forced to be unstable at time $t = T$ for 0.1 secs. Fig. 23 shows that the RMMAC rapidly recovers to a stable expected configuration. Fig. 23(a) shows that the “correct probability” $P_1(t) \rightarrow 1$ has been forced to $P_1(60) = 0.01$. Fig. 23(b) shows the output response in which the RMMAC quickly recovers and returns to its predictable superior disturbance-rejection.

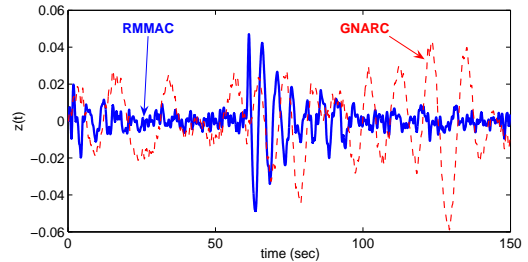
Remark: Provided that all convergence assumptions hold, (Baram, 1976), we observed in all our simulations that a temporary instability of the RMMAC will cause all signals to grow. As a consequence, the signal-to-noise ratio will increase, and it will be reflected in the size of the residuals; this, in turn, appears to force the PPE to adjust the posterior probabilities so that a stable configuration is re-established.

5.7 Violating RMMAC Convergence Assumptions

The theory which guarantees the convergence of the posterior probabilities almost-surely (Baram,



(a) Posterior probabilities transients



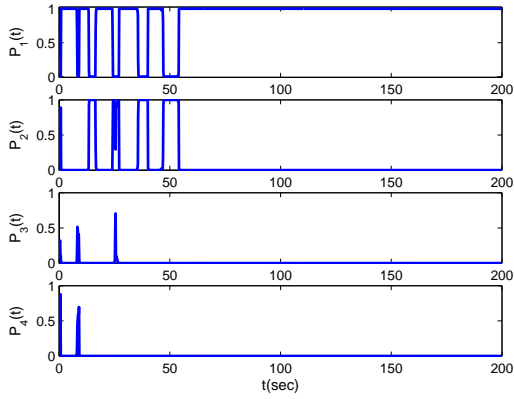
(b) Output response

Fig. 23. The RMMAC recovers from a forced unstable configuration at $T = 60$ secs.

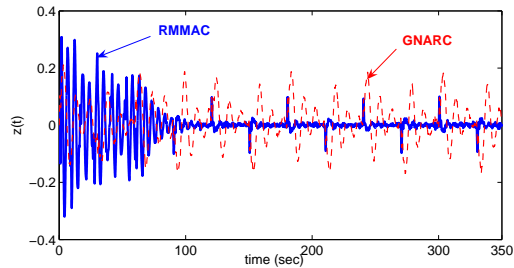
1976; Baram and Sandell, 1978b) assumes that all MMAE signals are stochastic stationary random processes. With some additional ergodicity conditions, all results presented in Section 5.6 satisfied those assumptions, and we have indeed observed convergence to a nearest probabilistic neighbor. In this section we evaluate the RMMAC performance when we intentionally violate some of the above assumptions. We shall show that the RMMAC still performs quite well; in a sense it appears “robust” to violating the theoretical assumptions. We evaluated the RMMAC performance over a wide variety of operating conditions, for different values of the uncertain spring stiffness. In all the RMMAC worked well and no instabilities were observed. The following representative cases are presented.

5.7.1. Step Disturbance: In this set of simulations we used a deterministic periodic square-wave disturbance, $d(t) = \pm 1.0$, with a period of 60 secs. The sensor noise was white as in Section 5.6. The KFs in the RMMAC were NOT aware of the square-wave disturbance; they continued to use (5.1) to model the disturbance dynamics. The true spring stiffness is $k_1 = 1.75$. Fig. 24 shows the simulation result for one MC run.

Note that after one period (60 secs) of the disturbance square-wave the probabilities converge to



(a) Posterior probabilities transients



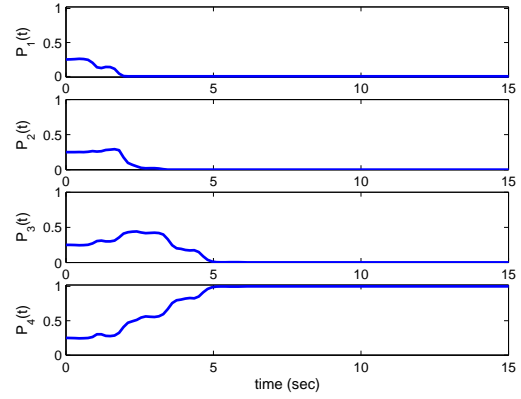
(b) Output response

Fig. 24. RMMAC performance for a square-wave disturbance.

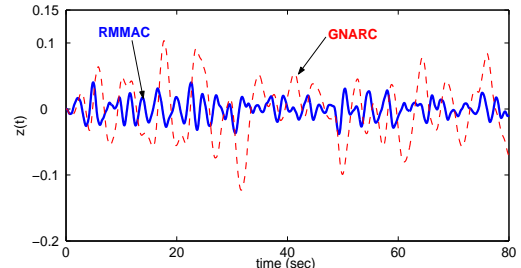
the correct Model #1, i.e. $P_1(t) \rightarrow 1$ as shown in Fig. 24(a). The RMMAC exhibits an oscillatory behavior during this time period (Fig. 24(b)) which dies out after the probability $P_1(t)$ converges to unity; afterwards the RMMAC performance is very good.

5.7.2. Sinusoidal Sensor Noise: We also tested “robustness to the assumptions” by using a high frequency sinusoid for the measurement noise, as $\theta(t) = 10^{-3} \sin(10t)$, rather than pure white noise. Fig. 25 shows a representative simulation using the value $k_1 = 0.3$, which is “close” to Model #4. Fig. 25(a) shows that the “correct” probability $P_4(t) \rightarrow 1$ within 5 secs. The improvement in disturbance-rejection is obvious from Fig. 25(b). Fig. 25(c) shows a comparison of the control signals. Both GNARC and RMMAC control signals have sinusoidal components at steady-state. Nonetheless, as expected, the amplitude of the RMMAC control is slightly larger than that of the GNARC (this comes as no surprise) since RMMAC yields improved disturbance-rejection. Only one MC run is shown.

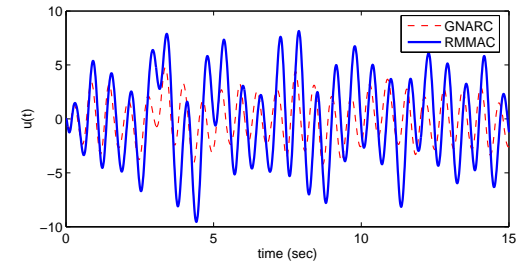
5.7.3. Slow Parameter variation: As mentioned in Section 1, the driving engineering motivation for using adaptive control was the need to deal with “slow” changes in the plant uncertain param-



(a) Posterior probabilities transients



(b) Output response



(c) Control signals

Fig. 25. RMMAC robustness to the sinusoidal sensor noise.

eters. In all the numerical simulations presented up to now in this section, 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 sinusoidal with frequency 0.01 rad/sec, i.e.

$$k_1(t) = 1 - 0.75 \cos 10^{-2}t \quad (5.19)$$

as shown in Fig. 26(a). The dashed-lines indicate the times that the spring crosses the boundaries of the four models of Table 2. Fig. 26(b) shows the

5.8 Discussion of Numerical Results

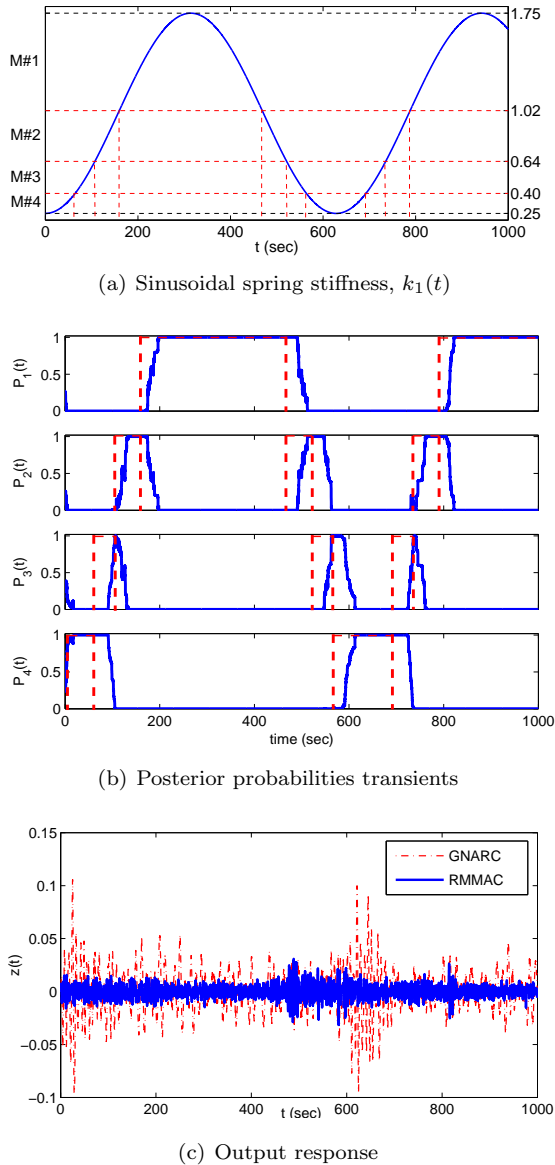


Fig. 26. RMMAC responses for the sinusoidal parameter variations, as in (5.19).

dynamic evolution of the four posterior probabilities, while Fig. 26(c) compares the GNARC and RMMAC performance output responses, which shows that the RMMAC continues to work quite well.

It is tempting to interpret Fig. 26(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 (5.19) 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. Nonetheless, the results of Fig. 26 (and many others not presented herein) demonstrate the ability of the RMMAC in dealing with slowly-varying uncertain parameters.

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 consistently had superior disturbance-rejection. No closed-loop instabilities were noted in thousands of MC simulations.

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 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 (Fekri *et al.*, 2004a) 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, (Åström, 2000). 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 (Fekri *et al.*, 2004c) for a different two-cart mass-spring-dashpot non-collocated system with a mass uncertainty. However, the previous results (Fekri *et al.*, 2004a) and (Fekri *et al.*, 2004c) did *not* use the systematic FNARC based approach for defining the models (in the spirit of Fig. 13) nor the optimized KF design using the BPM (in the spirit of Fig. 17). Several other SISO and MIMO results and examples can be found in the doctoral dissertation of S. Fekri (Fekri, 2005).

6. RESEARCH NEEDS

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 theoret-

ical 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 using mixed μ -synthesis, following the ideas presented in Section 3.

The structure of the multi-controllers used in the SMMAC architecture does not 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 (Corradini *et al.*, 2004) 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 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 (Corradini *et al.*, 2004), each estimator uses the same “ A_E ” matrix, i.e. all observers have identical 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 (Corradini *et al.*, 2004) this was done, but the problem of designing suitable deterministic MIMO Luenberger observers, say by eigenstructure assignment, becomes quite complex.

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

7. CONCLUSIONS

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.

ACKNOWLEDGEMENTS

We thank Profs. B.D.O. Anderson and J. Hespanha for their suggestions, discussions and criticisms. We also are grateful to Prof. G.J. Balas for providing us with the latest version of the mixed- μ software (Balas, 2003) and Prof. Y. Baram for his help to provide us a copy of his Ph.D. thesis (Baram, 1976).

Appendix I.
SUMMARY OF THE MMAE FILTER

In this appendix, the equations of the multiple model adaptive estimation (MMAE) algorithm, as depicted 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 statistic. 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

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

where $\{u(n)\}$ is the deterministic (control) input and $\{w(n)\}$ and $\{v(n)\}$ are zero-mean white Gaussian sequences, mutually uncorrelated and uncorrelated with $x(0)$ with

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

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

$$\begin{aligned} \varphi &= \{\mathcal{M} : \mathcal{M}_k = (F_k, J_k, G_k, H_k, Q_k, R_k); \\ & \quad k = (1, 2, \dots, N)\} \end{aligned} \quad (\text{A.3})$$

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

$$\begin{aligned} x_k(n+1) &= F_kx_k(n) + J_ku(n) + G_kw(n) \\ y_k(n) &= H_kx_k(n) + v(n) \end{aligned} \quad (\text{A.4})$$

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

It is assumed that the true system that generates the data is one of the models in (A.4).

The set of past controls, $u(0), u(1), u(2), \dots, u(n-1)$, and the set of past noisy measurements, including the one at the *present* time n , $y(1), y(2), \dots, y(n-1), y(n)$, 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 $n \rightarrow \infty$

$$\hat{x}(n|n) = E\{x(n) | \underbrace{u(0), u(1), \dots, u(n-1); y(1), \dots, y(n-1), y(n)}_{Y(n)}\} \quad (\text{A.5})$$

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

$$\Sigma(n|n) = E[(x(n) - \hat{x}(n|n))(x(n) - \hat{x}(n|n))' | Y(n)] \quad (\text{A.6})$$

The MMAE filter shown in Fig. 1 is driven by the sequence of past controls and noisy sensor measurements while 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:

$$\begin{aligned} \hat{x}_k(n+1|n) &= F_k\hat{x}_k(n) + J_ku(n) \\ \hat{y}_k(n+1|n) &= H_k\hat{x}_k(n+1|n) \end{aligned} \quad (\text{A.7})$$

Update-cycle:

$$\hat{x}_k(n+1|n+1) = \hat{x}_k(n+1|n) + K_k r_k(n+1) \quad (\text{A.8})$$

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:

$$r_k(n+1) = y(n+1) - \hat{y}_k(n+1|n) \quad (\text{A.9})$$

Residual covariance:

$$S_k = \text{cov}[r_k(n+1); r_k(n+1)] = H_k \Sigma_k^p H_k^T + R_k \quad (\text{A.10})$$

KF gain:

$$K_k = \Sigma_k^p H_k^T S_k^{-1} \quad (\text{A.11})$$

The constant steady-state covariance equations are

Predict-cycle covariance Σ_k^p :

$$\Sigma_k^p = F_k \Sigma_k A_k^T + G_k Q_k G_k^T \quad (\text{A.12})$$

Update-cycle covariance Σ_k :

$$\Sigma_k = \Sigma_k^p - \Sigma_k^p H_k^T S_k^{-1} H_k \Sigma_k^p \quad (\text{A.13})$$

We stress that both the MMAE state-estimate (A.5) and state-covariance matrix (A.6) represent true conditional state estimate and its conditional covariance at steady-state (Athans and Chang, 1976; Anderson and Moore, 1979). This is because, one can explicitly calculate the conditional probability density function $p(x(n)|Y(n))$, 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 $\{\mathcal{H}_1, \mathcal{H}_2, \dots, \mathcal{H}_N\}$. It turns out that on-line 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,

$$\mathcal{H} \in \{\mathcal{H}_1, \mathcal{H}_2, \dots, \mathcal{H}_N\} \quad (\text{A.14})$$

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 $n = 0$, $P_k(0) \equiv \text{Prob}(\mathcal{H} = \mathcal{H}_k)$, are assumed known (typically $P_k(0) = \frac{1}{N}$), and

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

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

$$P_k(n) \geq 0, \sum_{k=1}^N P_k(n) = 1 \quad (\text{A.16})$$

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

$$E\{y(n+1)|u(n), \mathcal{H}_k, Y(n)\} = H_k \hat{x}_k(n+1|n) \quad (\text{A.17})$$

and steady-state covariance

$$\text{cov}[y(n+1); y(n+1)|u(n), \mathcal{H}_k, Y(n)] = H_k \Sigma_k^p H_k^T + R_k \triangleq S_k \quad (\text{A.18})$$

Furthermore,

$$p(y(n+1)|u(n), \mathcal{H}_k, Y(n)) = \frac{1}{(2\pi)^{m/2} \sqrt{\det S_k}} \cdot e^{-\frac{1}{2} r_k^T(n+1) S_k^{-1} r_k(n+1)} \quad (\text{A.19})$$

By using Bayes rule we deduce that

$$P_k(n+1) = \frac{p(y(n+1) | \mathcal{H}_k, u(n), Y(n))}{\sum_{j=1}^N P_j(n) p(y(n+1) | \mathcal{H}_j, u(n), Y(n))} \cdot P_k(n) \quad (\text{A.20})$$

For notational simplicity, define

$$w_j(n+1) = r_j'(n+1) S_j^{-1} r_j(n+1) \quad (\text{A.21})$$

$$\beta_j = ((2\pi)^{m/2} \sqrt{\det S_j})^{-1} \quad (\text{A.22})$$

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

$$P_k(n+1) = \frac{\beta_k e^{-\frac{1}{2} w_k(n+1)}}{\sum_{j=1}^N \beta_j e^{-\frac{1}{2} w_j(n+1)} P_j(n)} \cdot P_k(n) \quad (\text{A.23})$$

where $P_k(0)$ are the prior model probabilities as in (A.15).

Thus, at steady-state the MMAE generates the state estimate (exact conditional mean) by

$$\hat{x}(n|n) = \sum_{k=1}^N P_k(n) \hat{x}_k(n|n) \quad (\text{A.24})$$

and the exact conditional covariance matrix

$$\Sigma(n|n) = \sum_{k=1}^N P_k(n) \cdot [\Sigma_k(n|n) + (\hat{x}_k(n|n) - \hat{x}(n|n))(\hat{x}_k(n|n) - \hat{x}(n|n))'] \quad (\text{A.25})$$

A brief review on Multiple-Model Adaptive Estimation (MMAE) is given that was restricted to LTI systems driven by white Gaussian inputs having time-invariant statistics. Again it is noted that the above results hold in the sense that the true (unknown) plant is assumed to belong to the model set of (A.4).

Appendix II.

BARAM PROXIMITY MEASURE (BPM)

The main 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 metric sense, as proved by (Baram, 1976). 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 also (Baram, 1976) for more details.

II.1 Stationarity and Ergodicity

The purpose of this section is to provide definitions and convergence results for ergodic sequences used in the paper. It is not intended to provide an elaborate presentation of the concept of ergodicity. For a precise development of ergodicity theory the reader is referred to, e.g. (Doob, 1953; Halmos, 1956).

II.2 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

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

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 (Gray, 2001, Chap. 3). Let B_∞^ℓ be the σ -algebra of Borel sets of R_∞^ℓ where $R_\infty^\ell = R^\ell \times R^\ell \times \dots$. Then $\{x(n)\}$ is said to be *stationary* if for each $\kappa \geq 1$ and for every $C \in B_\infty^\ell$

$$P[\{x(1), x(2), \dots, x(n)\} \in C] = P[\{x(\kappa+1), x(\kappa+2), \dots, x(\kappa+n)\} \in C] \quad (\text{B.1})$$

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 (Doob, 1953). A zero mean stationary Gaussian process is *ergodic* if and only if (Gernander, 1959, pp.257–260); (Doob, 1953, pp.494)

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

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

II.3 Calculating the BPM

Based on a bank of N discrete Kalman filters, a set of N models for the system of (A.1) can be described by

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

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

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

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

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

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

$$I_j^i \equiv E\{[y(n) - \hat{y}_j(n)][y(n) - \hat{y}_j(n)]^T | \mathcal{H}_i \neq \mathcal{H}_j\}$$

denote the residual covariance matrix according to \mathcal{M}_j , when \mathcal{M}_i is the correct model.

We shall use the following condition (Baram, 1976, 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 (C.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 dynamic equation generating simultaneously the state $x(n)$ and its updated estimation by the j -th Kalman filter, $\hat{x}_j(n)$, is

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

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 \mathcal{M}_j .

For notational purposes, the following are used.

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

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 Lyapunov equation

$$\Psi_j^i(n+1) = F_j^i \Psi_j^i(n) F_j^{iT} + G_j^i Q_j^i G_j^{iT} \quad (\text{B.5})$$

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

$$\Psi_j^i = \lim_{n \rightarrow \infty} \Psi_j^i(n) \quad (\text{B.6})$$

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

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

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

$$\Psi_j^i = F_j^i \Psi_j^i F_j^{iT} + G_j^i Q_j^i G_j^{iT} \quad (\text{B.8})$$

It can be shown that

$$\Gamma_j^i = H_j^i \Psi_j^i H_j^{iT} + R_i \quad (\text{B.9})$$

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.

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

This is shown in (Baram, 1976, pp.80–81). However, note that the stability and observability of the models are only sufficient, and not necessary. In fact, (Baram, 1976, 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 (Grewal *et al.*, 2001, Chap. 7) for more details.

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

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

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 k against j is defined by (Baram, 78a, b)

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

and the distance measure is

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

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

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

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

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

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

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

Therefore,

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

Also,

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

It is easy to check that

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

if and only if

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

in which the following condition is essential.

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

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

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

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

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

In conclusion, the Baram proximity measure (BPM), can be an appropriate distance metric in the stochastic systems between the true model and each of the adopted models utilizing the corresponding bank of Kalman filters.

II.4 Remarks

The application of deterministic inputs to dynamic systems for the purpose of identification and their optimal selection, covering the reference tracking problems, is not addressed in this paper. Generally speaking, the convergence analysis in the presence of deterministic inputs requires a more elaborate analysis that is suggested for future research.

(Baram, 1976, Chap. 6) discusses a case that any deterministic input sequence, that satisfies a suitable assumption of (Baram, 1976, Theorem 6.3), will provide convergence in the mean of the identification procedures at a certain rate. see (Baram, 1976) and (Yared, 1979) for more details.

(Ljung, 1976) uses the prediction error parameter estimate obtained by minimizing a scalar function of the matrix

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

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 give the same output prediction as the true system in the following sense.

$$\liminf_{t \rightarrow \infty} \frac{1}{t+1} \sum_{\tau=0}^t |\hat{y}(\tau; p_0) - \hat{y}(\tau; p)|^2 = 0 \quad (\text{B.19})$$

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 same input-output relation as true model, if the input is general enough to excite all modes of the system. It is shown in (Ljung, 1976) 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

- Anderson, B.D.O. and J.B. Moore (1979). *Optimal Filtering*. Prentice Hall.
- Anderson, B.D.O. and J.B. Moore (1990). *Optimal Control - Linear Quadratic Methods*. Prentice Hall. NJ, USA.
- Anderson, B.D.O. *et al.* (2000). Multiple model adaptive control. part 1: Finite controller coverings. *Int. J. of Robust and Nonlinear Control* **10**(11-12), 909–929.
- Anderson, B.D.O. *et al.* (2001). Multiple model adaptive control with safe switching. *Int. J. of Adaptive Control and Signal Processing* **15**(5), 445–470.
- Åström, K.J. (2000). Limitations on control system performance. *European Journal of Control* **6**(1), 1–19.
- Åström, K.J. and B. Wittenmark (1995). *Adaptive Control*. 2nd ed., Addison-Wesley. MA, USA.
- Athans, M. *et al.* (1977). 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* **22**(5), 768–780.
- Athans, M. and C.B. Chang (1976). Adaptive estimation and parameter identification using multiple model estimation algorithms. Technical Report 1976-28. MIT Lincoln Lab., Lexington, MA, USA.
- Balas, G.J. (2003). Private communication re mixed- μ software.
- Balas, G.J. *et al.* (2004). *μ -Analysis and Synthesis Toolbox For Use with MATLAB, Users Guide*. 4th ed., The MathWorks Inc.
- Baram, Y. (1976). Information, consistent estimation and dynamic system identification. PhD thesis. MIT. Cambridge, MA, USA.
- Baram, Y. and N.R. Sandell (1978a). Consistent estimation on finite parameter sets with application to linear systems identification. *IEEE Trans. on Automatic Control* **23**(3), 451–454.
- Baram, Y. and N.R. Sandell (1978b). An information theoretic approach to dynamical systems modeling and identification. *IEEE Trans. on Automatic Control* **23**(1), 61–66.
- Barros, D. *et al.* (2005). Robust mixed- μ synthesis performance for a mass-spring benchmark system with stiffness uncertainty. *Proc. 13th IEEE Mediterranean Conference on Control and Automation (MED'05)*. Limassol, Cyprus.
- Chang, C. and M. Athans (1978). State estimation for discrete systems with switching parameters. *IEEE Trans. on Aerospace and Electronic System* **14**(3), 418–425.
- Corradini, M.L. *et al.* (2004). Robust stabilization of multivariable uncertain plants via switching control. *IEEE Trans. on Automatic Control* **49**(1), 107–114.
- Doob, J.L. (1953). *Stochastic Processes*. John Wiley & Sons, Inc., New York.
- Fekri, S. *et al.* (2004a). A new robust adaptive control method using multiple-models. *Proc. 12th IEEE Mediterranean Conference on Control and Automation (MED'04)*. Kusadasi, Turkey.
- Fekri, S. *et al.* (2004b). RMMAC: A novel robust adaptive control scheme – Part I: Architecture. *Proc. of the IEEE Conf. on Decision and Control*. Paradise Island, Bahamas.
- Fekri, S. *et al.* (2004c). RMMAC: A novel robust adaptive control scheme – Part II: Performance evaluation. *Proc. of the IEEE Conf. on Decision and Control*. Paradise Island, Bahamas.
- Fekri, S. (2005). Robust adaptive MIMO control using multiple-model hypothesis testing and mixed- μ synthesis. PhD thesis. Instituto Superior Técnico. Lisbon, Portugal.
- Gelb, A. (1974). *Applied Optimal Estimation*. MIT Press. MA, USA.
- Gernander, U. (1959). *Stochastic Processes and Statistical Inference*. Vol. 1. Arkiv für Mathematik.
- Gray, R.M. (2001). *Probability, Random Processes, and Ergodic Properties*. Springer Verlag.

- Greene, C.S. (1975). An analysis of the multiple model adaptive control algorithm. PhD thesis. MIT. Cambridge, MA, USA.
- Greene, C.S. and A.S. Willsky (1980). An analysis of the multiple model adaptive control algorithm. *Proc. of the IEEE Conf. on Decision and Control*. pp. 1142–1145.
- Grewal, M.S. *et al.* (2001). *Global Positioning Systems, Inertial Navigation, and Integration*. John Wiley & Sons, Inc.
- Halmos, P.R. (1956). *Lectures on Ergodic Theory*. Chelsea Publ. Co., New York.
- Hawkes, R.M. and J. Moore (1976). Performance of bayesian parameter estimators for linear signal models. *IEEE Trans. on Automatic Control* **21**(4), 523–527.
- Hespanha, J.P. *et al.* (2001). Multiple model adaptive control. part 2: switching. *Int. J. of Robust and Nonlinear Control* **11**(5), 479–496.
- Ioannou, P. and J. Sun (1996). *Robust Adaptive Control*. Prentice Hall. NJ, USA.
- Jun, M. and M.G. Safonov (1999). Automatic PID tuning: An application of unfalsified control. *Proc. IEEE CCA/CACSD*. pp. 328–333.
- Kosut, R.L. and B.D.O. Anderson (1988). Adaptive control via finite modelling and robust control. *IFAC Workshop on Robust Adaptive Control* pp. 91–95.
- Kwakernaak, H. and R. Sivan (1972). *Linear Optimal Control Systems*. John Wiley. NY, USA.
- Lainiotis, D.G. (1971). Optimal adaptive estimation: Structure and parameter adaptation. *IEEE Trans. on Automatic Control* **AC-16**(2), 160–170.
- Landau, I.D. (1979). *Adaptive Control: The Model Reference Approach*. Marcel Dekker. NY, USA.
- Ljung, L. (1976). On the Consistency of Prediction Error Identification Methods. pp. 121–164. in *System Identification: Advances and Case Studies*, (R.K. Mehra and D.G. Lainiotis, eds.). Academic Press. New York.
- Magill, D.T. (1965). Optimal adaptive estimation of sampled stochastic processes. *IEEE Trans. on Automatic Control* **AC-10**(4), 434–439.
- Maybeck, P.S. and J. Griffin (1997). MMAE/MMAC control for bending with multiple uncertain parameters. *IEEE Trans. on Aerospace and Electronic System* **33**(3), 903–912.
- Morse, A.S. (1996). Supervisory control of families of linear set-point controllers – part 1: Exact matching. *IEEE Trans. on Automatic Control* **41**(10), 1413–1431.
- Morse, A.S. (1997). Supervisory control of families of linear set-point controllers – part 2: Robustness. *IEEE Trans. on Automatic Control* **42**(11), 1500–1515.
- Morse, A.S. (1998). A bound for the disturbance - to - tracking - error gain of a supervised set point control system. pp. 23–41. in *Perspectives in Control Theory and Applications*, (D. Normand-Cyrot, ed.). Springer-Verlag. Berlin.
- Morse, A.S. (2004). Analysis of a supervised set-point control containing a compact continuum of finite-dimensional linear controllers. *Proc. MTNS*. Belgium.
- Narendra, K.S. and A. Annaswamy (1988). *Stable Adaptive Systems*. Prentice Hall. NJ, USA.
- Narendra, K.S. and J. Balakrishnan (1997). Adaptive control using multiple models. *IEEE Trans. on Automatic Control* **42**(2), 171–187.
- Rohrs, C.E. *et al.* (1985). Robustness of continuous-time adaptive control algorithms in the presence of unmodeled dynamics. *IEEE Trans. on Automatic Control* **30**(9), 881–889.
- Safonov, M.G. and T.-C. Tsao (1995). The Unfalsified Control Concept: A Direct Path From Experiment To Controller. pp. 196–214. in *Feedback Control, Nonlinear Systems, and Complexity*, (B.A. Francis and A.R. Tannenbaum, eds.). Springer-Verlag.
- Safonov, M.G. and T.-C. Tsao (1997). The unfalsified control concept and learning. *IEEE Trans. on Automatic Control* **42**(6), 843–847.
- Sastry, S. and M. Bodson (1989). *Adaptive Control: Stability, Convergence and Robustness*. Prentice-Hall. NJ, USA.
- Schiller, G. and P.S. Maybeck (1997). Control of a large space structure using MMAE/MMAC techniques. *IEEE Trans. on Aerospace and Electronic System* **33**(4), 1122–1130.
- Schott, K.D. and B.W. Bequette (1997). Multiple Model Adaptive Control. pp. 269–292. in *Multiple Model Approaches to Modelling and Control*, (R. Murray-Smith and T.A. Johnsen eds.). Taylor & Francis. London.
- Shomber, H.R. (1980). An extended analysis of the multiple model adaptive control algorithm. PhD thesis. MIT. Cambridge, MA, USA.
- Willner, D. (1973). Observation and control of partially unknown systems. PhD thesis. MIT. Cambridge, MA, USA.
- Yared, K.I. (1979). Maximum likelihood identification of state space models for linear dynamic systems. PhD thesis. MIT. Cambridge, MA, USA.
- Young, P.M. *et al.* (1992). Practical computation of the mixed- μ problem. *Proc. of the American Control Conf.*. Chicago, IL. pp. 2190–2194.
- Young, P.M. *et al.* (1995). Computing bounds for the mixed- μ problem. *Int. J. of Robust and Nonlinear Control* **5**, 573–590.
- Young, P.M. (1994). Controller design with mixed uncertainties. *Proc. of the American Control Conf.*. Baltimore, MD. pp. 2333–2337.
- Zhou, K. *et al.* (1996). *Robust and Optimal Control*. Prentice-Hall. NJ, USA.
- Zhou, K. (1998). *Essential of Robust Control*. Prentice-Hall. NJ, USA.