Robust Multiple Model Adaptive Control (RMMAC): A Case Study

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

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

SUMMARY

We evaluate the performance of the RMMAC methodology by considering a mass-spring-dashpot (MSD) system subject to high-frequency disturbances that strongly excite all its lightly-damped oscillatory modes. The results demonstrate the superior performance of the RMMAC and its variant RMMAC/XI architecture for a much more difficult adaptive control problem than that designed and analyzed in Ref. [1]. Copyright © 2006 John Wiley & Sons, Ltd.

KEY WORDS: Adaptive Control, Robust Adaptive Control, Multiple-Model Architectures, Hypothesis Testing, Robust Feedback Control Synthesis

1. Introduction

The RMMAC architecture and its variants have been discussed in Refs. [1], [2] and references therein. We assume that the reader is familiar with the general concepts and the mass-spring-dashpot (MSD) example presented in [1].

In this paper we examine the same physical MSD system considered in [1] and shown in Figure 1. The results presented in the sequel consider a (colored) stochastic disturbance force, $d(t)$, whose power spectral density has a higher bandwidth than that considered in [1] and which strongly excites all the lightly-damped modes of the MSD system. Furthermore, in this paper, the posed performance requirements demand *superior disturbance-rejection* regarding the position $z(t)$ of mass m_2 over a wider frequency range as well. These changes will result in a system with much higher bandwidth than that considered in [1] and will also increase the complexity of the RMMAC designs.

By comparing the results of these two RMMAC designs, that is, those in [1] against those described in the sequel, we wish to point out several key concepts that are important in

*S. Fekri is now Research Associate in Department of Engineering, Control & Instrumentation Research Group, University of Leicester, Leicester, UK. E-mail: sf111@le.ac.uk, Tel: +44-116-2522874, Fax: +44-116-2522619 (Correspondences to: University Road, Leicester LE1 7RH, UK).

§M. Athans is also Professor (emeritus) of the Dept. of EE&CS, MIT, Cambridge, MA, USA.

This paper was recommended for publication in revised form by Associate Editor XXX. YYYYYYY.

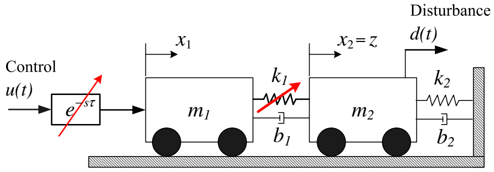


Figure 1. The MSD two-cart system with a spring whose stiffness k_1 is uncertain and which includes an unmodeled bounded time-delay, τ, in the control channel.

designing either robust *non-adaptive* controllers or RMMAC-based systems. These include:

(a) Changes in the exogenous disturbances power spectral densities and associated frequency-dependent performance weights drastically impact and change both the non-adaptive and the adaptive designs.

(b) The potential performance improvements of the adaptive RMMAC design over the best non-adaptive are also critically dependent upon the choices in (a) as well as on the frequency-domain specifications for superior disturbance-rejection.

(c) Conditions (a) and (b) also drastically impact the complexity of the RMMAC designs and their performance.

For these reasons, we re-iterate our philosophy that in designing adaptive controllers, *robust-stability considerations are not enough.* In order to evaluate the benefits of any adaptive feedback system *explicit robust-performance requirements must also be specified;* these also critically depend upon modeling of the exogenous signal characteristics (in our case, disturbance dynamics and sensor noise models).

In Section 2 we discuss the physical MSD system, frequency-domain bounds on its unmodeled dynamics (bounded time-delay), the lower- and upper-bounds upon the uncertain spring stiffness, the modeling of the disturbance force, the sensor noise intensity, and define the explicit performance requirements for robust disturbance-rejection and for limiting the high-frequency control behavior. In Section 3 we follow the step-by-step procedures described in [1] to design all compensators using the mixed-μ design robust synthesis [3, 4, 5, 6, 7] method and associated MATLAB software [8, 9]. Next, we derive the models for the RMMAC architecture based upon our posed explicit adaptive performance specifications. Moreover, we present the predicted improvements in disturbance-rejection by the RMMAC scheme. In Section 4 we discuss the design the bank of the Kalman Filters (KFs) and the Posterior Probability Evaluator (PPE) required to completely define the RMMAC architecture. In Section 5 we present several representative stochastic simulation results to illustrate the superior disturbance-rejection of the RMMAC as compared to the best non-adaptive design. The simulation results illustrate the RMMAC performance when the theoretical assumptions are not violated as well as when they are “mildly violated”. In Section 6 we discuss the poor performance of the RMMAC when we present situations in which the theoretical assumptions are ”severely violated” such as enforced temporary instability and time-variation of the spring constant. We discuss the poor performance of the RMMAC when the disturbance environment is highly uncertain as well – an important engineering consideration – and demonstrate how the variant RMMAC/XI architecture presented in [1] results in superior performance as well. Section 7 summarizes our conclusions.

2. The MSD System and Performance Specifications

In this section we present the equations of the MSD system in Figure 1 and its uncertainties and present the models for the exogenous signals and the performance requirements for both robust-stability and robust-performance as required by the mixed-μ design methodology.

The system in Figure 1 includes a colored stochastic-process disturbance force $d(t)$, acting on mass m_2, and additive sensor noise on the single 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 modeled by 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:

$$E\{\xi(t)\} = 0,\ E\{\xi(t)\xi(\tau)\} = \Xi\delta(t-\tau),\ \Xi = 1 \tag{2.1}$$

$$d(s) = W_d(s)\,\xi(s) \tag{2.2}$$

and the "disturbance dynamics" are

$$W_d(s) = \frac{3}{s+3} \tag{2.3}$$

where the frequency range $\omega \leq 3$ rad/sec is where the disturbance has most of its power. In [1] the disturbance $W_d(s)$ was $W_d(s) = 0.1/(s+0.1)$ reflecting a lower bandwidth disturbance. The current disturbance strongly excites all of the lightly-damped modes and will result in a more complex high-bandwidth design.

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

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) + L\xi(t) \\ y(t) &= Cx(t) + \theta(t) \end{aligned} \tag{2.4}$$

where the state vector is

$$x^T(t) = [x_1(t)\ x_2(t)\ \dot{x}_1(t)\ \dot{x}_2(t)\ d(t)] \tag{2.5}$$

and

$$\begin{aligned} A &= \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ -\frac{k_1}{m_1} & \frac{k_1}{m_1} & -\frac{b_1}{m_1} & \frac{b_1}{m_1} & 0 \\ \frac{k_1}{m_2} & -\frac{(k_1+k_2)}{m_2} & \frac{b_1}{m_2} & -\frac{(b_1+b_2)}{m_2} & \frac{1}{m_2} \\ 0 & 0 & 0 & 0 & -3 \end{bmatrix} \\ B^T &= \begin{bmatrix} 0 & 0 & \frac{1}{m_1} & 0 & 0 \end{bmatrix}; C = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \end{bmatrix} \\ L^T &= \begin{bmatrix} 0 & 0 & 0 & 0 & 3 \end{bmatrix} \end{aligned} \tag{2.6}$$

The following parameters in eq. (2.6) are fixed and known (and have the same values as in [1]):

$$m_1 = m_2 = 1,\ k_2 = 0.15,\ b_1 = b_2 = 0.1 \tag{2.7}$$

The lower- and upper-bound for the uncertain spring constant, k_1, are, as in Ref. [1],

$$\Omega = \{k_1 : 0.25 \leq k_1 \leq 1.75\} \tag{2.8}$$

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

$$z(t) \equiv x_2(t) \tag{2.9}$$

All feedback loops utilize a single noisy measurement $y(t)$, as in [1], the position of mass m_2, that includes additive white sensor noise $\theta(t)$, independent of $\xi(t)$, defined by

$$y(t) \equiv x_2(t) + \theta(t) \tag{2.10}$$

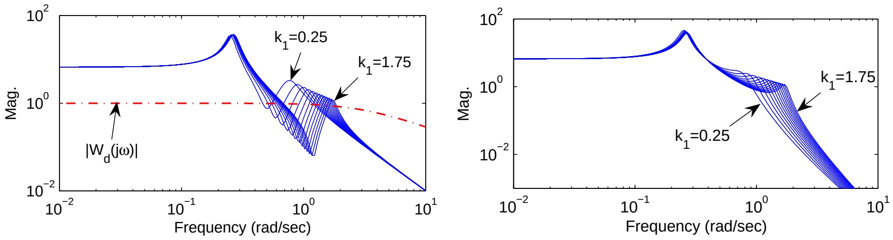
$$E\{\theta(t)\} = 0,\ E\{\theta(t)\theta(\tau)\} = 10^{-6}\,\delta(t-\tau) \tag{2.11}$$

By comparing eqs. (2.1) and (2.11) we note that the intensity of the sensor noise is much smaller than that of the process noise.

The desired disturbance-rejection requires that the effects of $d(t)$ (primarily) and also $\theta(t)$ be minimized so that $z(t) \approx 0$ reflecting our desire for superior disturbance-rejection.

As in [1], the control problem is hard even if the spring constant k_1 is known. We deal with a *non-collocated actuator* problem because the control is not applied directly to the mass m_2 whose position is to be regulated. Clearly, the control problem becomes even harder in
our adaptive design, because the control $u(t)$ is applied through the uncertain spring and
unmodel
much force is exerted through the uncertain
spring k

The o
SD system, from $d(t)$ to $z(t)$ and also from
control u
e shown in Figure 2 for different values of k_1;
see eq. (
ost noticeable in the higher frequency mode.
Figure 2
h shows that all modes are strongly excited
by $d(t)$.



(a) From disturbance $d(t)$ to output $z(t)$ (b) From control $u(t)$ to output $z(t)$

Figure 2. Bode plot of the open-loop MSD system for different values of spring constant k_1. Figure 2(a) also shows the disturbance low-pass filter $|W_d(j\omega)|$ of eq. (2.3) which demonstrates that the disturbance strongly excites all modes.

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 secs, i.e.

$$\tau \leq 0.05 \text{ secs} \tag{2.12}$$

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

$$W_{un}(s) = \frac{2.45s}{s+40} \tag{2.13}$$

This frequency-domain bound on the unmodeled time-delay is shown in Figure 3, and is the same as that used in [1].

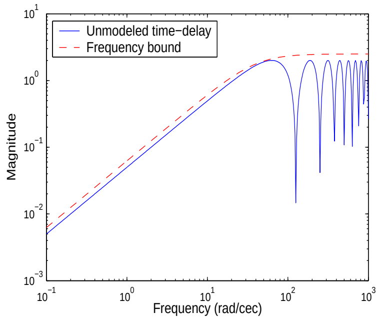
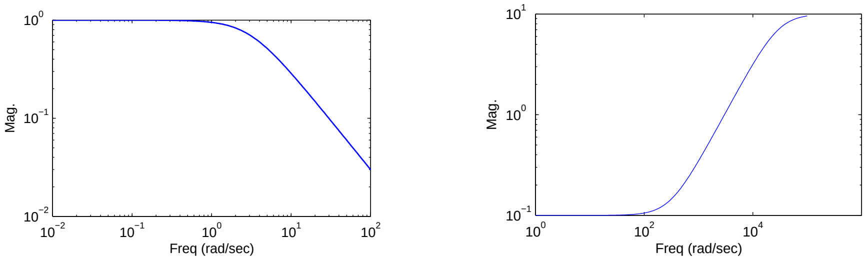


Figure 3. Frequency response of unmodeled time-delay dynamics for $\tau = 0.05$ secs and its frequency-domain upper bound $|W_{un}(j\omega)|$ of eq. (2.13).

We also select a control frequency weight, denoted by $W_u(s)$, in order to penalize the control $u(t)$ differently in different frequency regions. This is used to limit the bandwidth of the closed-loop system. The $W_u(s)$ used in this design is

$$W_u(s) = \frac{10(s+300)}{s+30000} \tag{2.14}$$

so that we allow larger controls in lower frequencies and we penalize for large controls at much higher frequencies. The weight $W_u(s)$ in eq. (2.14) is different than that used in [1], since we deal here with a higher bandwidth design, and its frequency response is shown in Figure 4(b). In other words, the control weight (2.14) is a higher ”high-pass filter” from that in [1]. This is necessary to allow reasonable transition of the loop transfer functions from high-gain to low-gain.



(a) The 1st order disturbance modeling low-pass filter, $W_d(s)$

(b) Control weight $W_u(s)$, a high-pass filter that penalizes the system for using large controls at high frequencies

Figure 4. Frequency-weighted functions used in this design.

In order to carry out the mixed-μ synthesis for designing the “best possible” non-adaptive feedback system, we also need to select performance weights to reflect the desired frequency-dependent performance objective. The transfer function of the performance weight upon the output $z(t)$ is defined as

$$W_p(s) = A_p \, \frac{3}{s+3} \tag{2.15}$$

which reflects our performance specification for good disturbance-rejection in the frequency range $\omega \leq 3$ rad/sec, where the disturbance $d(t)$ has most of its power. Notice, from eqs. (2.3) and (2.15), that the "corner frequency", $\omega = 3$ rad/sec, for both the disturbance dynamics, $W_d(s)$, and the performance weight, $W_p(s)$, is the same. Thus, we are interested in superior disturbance-rejection precisely in the *same* frequency band as the disturbance. Difference choices would yield different results.

We also select, as in [1], a constant noise weight

$$W_n = 10^{-3} \tag{2.16}$$

consistent with eq. (2.11).

We emphasize that the dynamics of both $W_p(s)$ and $W_d(s)$, eqs. (2.15) and (2.3), are the same except for the "gain" A_p in eq. (2.15). In fact, the performance weight $W_p(s)$ penalizes the output error in the same frequency range as the disturbance dynamics $W_d(s)$, while the gain parameter A_p in $W_p(s)$ specifies our desired level of disturbance-rejection. The larger A_p, the greater the penalty on the effect of the disturbances on the position of mass m_2. 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.

In summary, the only differences between the design parameters used in this paper versus those used in the design in [1] are:

(a). In [1] the disturbance weight was $W_d(s) = \frac{0.1}{s+0.1}$ rather than that of eq. (2.3).

(b). In [1] the control weight was $W_u(s) = \frac{10(s+10)}{s+1000}$ rather than that of eq. (2.14).

(c) In [1] the performance weight was $W_p(s) = A_p \frac{0.1}{s+0.1}$ rather than that of eq. (2.15).

Figure 5 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.

3. Designing The Robust Dynamic Compensators

Consider the RMMAC architecture, [1,2] as shown in Figure 6. In this section we determine the number N and design each of the N "local non adaptive robust compensators (LNARCs)" that comprise the *control subsystem* in Figure 2

Following the procedure of [1], we must first determine what is the *best* "global non adaptive robust compensator (GNARC)" for the MSD system in Section 1 and the associated frequency weights. We stress that the GNARC represents the *best non adaptive design which guarantees stability- and performance-robustness* in the presence of both the unmodeled dynamics – the bounds on the unmodeled dynamics of eq. (2.13) – and the real-valued spring uncertainty of eq. (2.8). The GNARC is determined through iterative utilization of the mixed-μ D,G-K

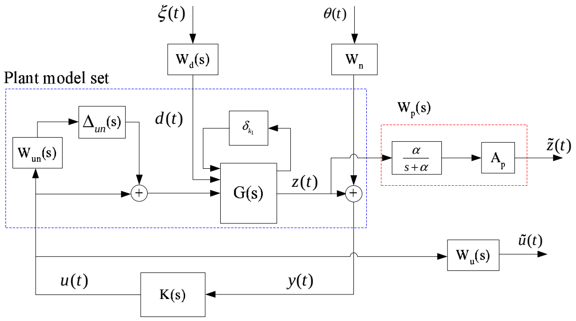


Figure 5. The MSD system with weights for mixed-μ synthesis.

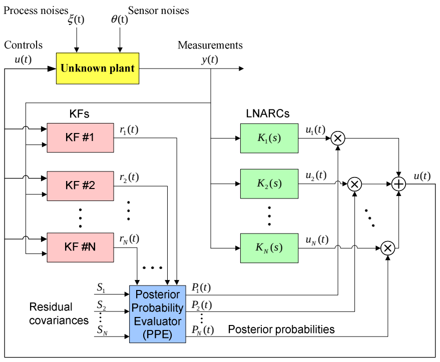


Figure 6. The RMMAC architecture.

algorithms* [9] until the performance parameter A_p in eq. (2.15) is maximized. Appendix A shows some of the design choices we have made in the D,G-K iterations.

The largest performance parameter A_p in eq. (2.15) by using the GNARC with the entire uncertainty bound of eq. (2.8) was determined to be

$$A^G_{p\,\max} = 0.79 \quad (\text{with } \mu_{ub} \approx 0.995) \tag{3.1}$$

Next, we determine what we call the *best* "fixed non adaptive robust compensator (FNARC)" [1,2]. In this case, we assume we know, i.e. "fix", k_1 in the interval (2.8) and we determine the best robust compensator by again maximizing the performance parameter A_p in eq. (2.15); all other frequency weights remain the same including the bound on unmodeled dynamics defined by eq. (2.13). The procedure is repeated for every k_1 in the interval (2.8). We stress that the FNARC defines an *upper bound* on the potential adaptive performance and it yields an infinite number of models in the RMMAC architecture of Figure 6, [1].

*We remark, as in [1], that these algorithms are not as yet available in the commercially available version of the MATLAB software [8]

The outcome of the optimized FNARC and GNARC performance parameter for this example is shown in Figure 7.

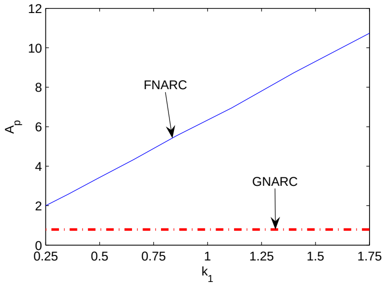


Figure 7. Best GNARC and FNARC performance parameter A_p for all $k_1 \in \Omega = [0.25, 1.75]$ and any unmodeled dynamics defined by the magnitude of the transfer function (2.13).

The reader should compare Figure 7 with Figure 13 in [1] to note the drastic changes in both the GNARC and FNARC performance caused by the change of the specifications to demand better disturbance-rejection over a wider frequency range. Thus, the MSD design of this paper is much harder than that in [1] and this is reflected to the reduction of the best possible potential robust-performance (disturbance-rejection).

Next we describe how to determine the number of models in the RMMAC architecture of Figure 6. As explained in [1], this requires an additional design specification for the desired performance of the RMMAC system. As in [1], we shall demand that the RMMAC should meet or exceed X% of the FNARC performance.

Following the search algorithm, described in [1], to find the required N “models” and associated ”local non adaptive robust compensators” (LNARCs) using the design specification of $X = 66\%$ of the FNARC for the adaptive design, we obtain Figure 8 that illustrates the results (In [1] we used the value X=70%).

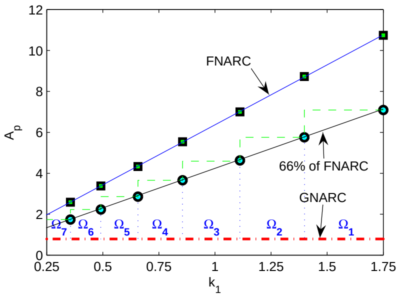


Figure 8. Definition of $N = 7$ required models and LNARCs resulting from the requirement that the RMMAC must guarantee at least 66% of the upper-bound (FNARC) potential performance.

The outcome of Figure 8 is summarized in Table I. As a result, the large uncertainty set Ω of eq. (2.8) yields seven “models”, or subintervals Ω_k, and seven local compensators LNARCs (In [1] we obtained five models using x=70%). Clearly, the reduction in parameter uncertainty

Table I. GNARC and LNARC Performance

Compensator	Ω	A_p
GNARC	$\Omega = [0.25, 1.75]$	0.79
LNARC #1	$\Omega_1 = [1.40, 1.75]$	7.10
LNARC #2	$\Omega_2 = [1.11, 1.40]$	5.76
LNARC #3	$\Omega_3 = [0.86, 1.11]$	4.60
LNARC #4	$\Omega_4 = [0.66, 0.86]$	3.66
LNARC #5	$\Omega_5 = [0.49, 0.66]$	2.86
LNARC #6	$\Omega_6 = [0.36, 0.49]$	2.23
LNARC #7	$\Omega_7 = [0.25, 0.36]$	1.74

allows larger performance gains A_p for designing the LNARCs, resulting into guaranteed *both* stability- and performance-robustness over the subintervals Ω_k of Table I.

Note from Table I that we should expect better disturbance-rejection by the RMMAC for higher values of the spring stiffness k_1. For example, for $k_1 \in \Omega_1$, i.e. a stiff spring, we should expect about $7.10/0.79 \cong 9$ times better disturbance-rejection at low frequencies, while for $k_1 \in \Omega_7$, i.e. for a soft spring, we should only expect about $1.74/0.79 \cong 2.2$ times better disturbance-rejection[†].

We stress that in the MSD design reported in [1] we imposed an adaptive performance specification of 70% of the FNARC and this led to the requirement of *five models* and associated LNARCs. In this paper, we demand a (slightly smaller) performance of 66% of the FNARC that, in turn, yields seven models and associated LNARCs. This illustrates how changes in performance specifications impact the complexity of the RMMAC architecture.

As a result, the GNARC and seven LNARCs, $K_1(s), \ldots, K_7(s)$ are designed for each subinterval shown in Table I. These LNARCs guarantee 66% of the FNARC performance and a performance improvement of 2.2 to 9 times better than that of the best GNARC design. If we require better performance than 66%, we will need more than seven LNARCs. If we wish less than 66% FNARC performance we will need fewer than seven LNARCs.

Figure 9 compares the frequency response of the GNARC with that of the seven LNARCs. Note that at low frequencies (up to approximately 1 r/s), the LNARCs generate a loop-gain about 3.3 to 11 times larger than that of the GNARC. This is why we can obtain superior disturbance-rejection.

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; it follows that a specific LNARC is used in the feedback loop. After the probability convergence, the RMMAC essentially operates as an LTI stochastic feedback system which allows us to calculate the key transfer function for disturbance-rejection

$$\text{Disturbance-rejection transfer function:} \qquad M_{dz}(s) \equiv \frac{z(s)}{d(s)} \tag{3.2}$$

for different values of the unknown spring constant, $k_1 \in [0.25, 1.75]$, for the GNARC and for

[†] We remark that throughout this paper we use the (somewhat abusive) notation M #k and subinterval Ω_k to mean the same thing.

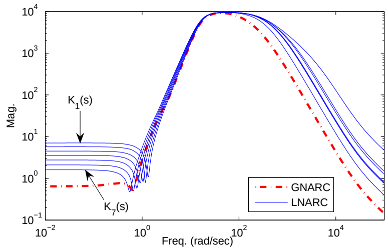
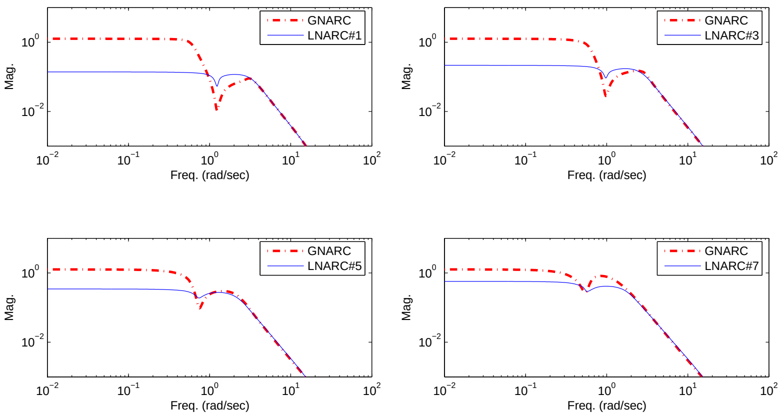


Figure 9. Frequency-domain characteristics of the GNARC and LNARCs compensators.

each LNARC design.

Figure 10 illustrates the above using different values of k_1 quantifying the potential RMMAC improvement in *disturbance-rejection*. It is stressed that these results are consistent with the $1/A_p$ levels (as given in Table I) at low frequencies. Since these plots involve the actual plant, the unmodeled dynamics (time-delay) are set to zero.

Figure 10 predicts that the RMMAC has the potential to significantly improve disturbance-rejection; these predictions will be validated in the sequel. Other transfer functions could also be computed (not shown), e.g. from the white noise $\xi(t)$ and the sensor noise $\theta(t)$ to the control $u(t)$.

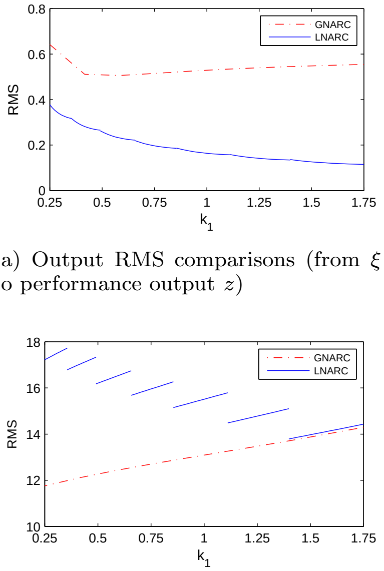


(c) $k_1 = 0.55 \in \Omega_5$

(d) $k_1 = 0.3 \in \Omega_7$

Figure 10. RMMAC potential improvement via plots of the disturbance-rejection transfer function $|M_{dz}(j\omega)|$, for different values of k_1.

Figure 11 evaluates the potential performance improvement of the RMMAC using stochastic metrics, namely by comparing the RMS errors of the output z and the control u, for different values of k_1 due to $\xi(t)$ only. They are in fact the RMS comparisons of GNARC with LNARCs, in the closed-loop transfer function from disturbance input ξ to the output performance z and to the control signal. These RMS results are readily computed by solving standard covariance algebraic Lyapunov equations for stochastic LTI systems [10]. The graphs of Figure 11(a) clearly suggest that RMMAC has the potential of decreasing the output RMS by a factor of



(a) Output RMS comparisons (from ξ to performance output z)

(b) Control RMS comparisons (from ξ to control u)

Figure 11. Predicted potential RMS performance of the RMMAC vs GNARC due to $\xi(t)$ only.

1.7 to 4.8 over that achievable by the GNARC system.

We note that the mixed-μ synthesis generates $\mathcal{H}_\infty$ compensators that, unlike $\mathcal{H}_2$ compensators, do not minimize RMS values. Nonetheless, stochastic RMS comparisons of the type shown in Figure 10 are useful for engineering insights.

Next, we discuss the important "Mismatch Model/LNARC" concepts summarized in Table II. From Figure 8 we note that the seven different LNARCs have significantly different low-frequency gains (so that they yield the improvements in disturbance- rejection shown in Figure 9 and 10). For this reason, we should expect that certain Model/LNARC combinations might lead to instability. Table II explains what are the effects upon closed-loop stability if LNARC #j, j=1,2,. . . ,7, is used when the true model is M # i, i=1,2,. . . ,7. We stress that the calculations leading to Table II did not include unmodeled dynamics.

The diagonal entries in Table II, by construction, guarantee robust closed-loop stability (S); indeed they also guarantee performance-robustness. However, we also have closed-loop stability (S) for Model/LNARC combinations "near" the main diagonal. This is because mixed-μ is only a sufficient condition for both stability- and performance-robustness and hence the resulting LNARC compensators have a wider stability margin. The entries marked (CU) means that whether or not we have a stable "mismatched" feedback combination depends on the specific value of k_1 in the associated model. Certain mismatched combinations are always unstable (U).

The information contained in Table II can be explained by considering the open-loop Bode plots of the MSD system of Figure 2(b) and those of the LNARCs in Figure 8. For example, looking at the last column of Table II we see that LNARC #7 always yields closed-loop stability

Table II. Mismatched Model/LNARC Effects

	LNARC						
M #	#1	#2	#3	#4	#5	#6	#7
1	S	S	S	S	S	S	S
2	S	S	S	S	S	S	S
3	S	S	S	S	S	S	S
4	CU	S	S	S	S	S	S
5	U	CU	S	S	S	S	S
6	U	U	CU	S	S	S	S
7	U	U	U	CU	S	S	S

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

for all seven models, i.e. for all values of k_1 in eq. (2.8). This is because the LNARC K$_7$(s) has the lowest gain of all LNARCs as evidenced by Figure 8.

On the other hand, looking at all unstable (U) combinations in the "southwest corner" of Table II we see, from. Figure 8, that they involve the high-gain LNARCs #1, #2, and #3 in conjunction with the "soft spring" higher-gain open-loop dynamics in Figure 2(b). In the later case, the feedback combination of these two high-gain dynamic systems leads to instability.

We shall use the insights provided by Table II to test the RMMAC system of Figure 2, in the stochastic simulations of Section 5, when we "force" the RMMAC to an unstable Model/LNARC combination and observe whether or not it recovers from such forced instability.

This completes the design of the seven LNARCs, which comprise the *"control subsystem"* of the RMMAC architecture of Figure 2.

4. Designing The Bank Of Kalman Filters

We next turn our attention to the design of the "identification subsystem" of the RMMAC architecture of Figure 2 that requires the design of seven Kalman Filters (KFs) and of the subsequent *"Posterior Probability Evaluator (PPE)"*. As indicated in [1], the correct design of the KFs is critical for asymptotic convergence of the RMMAC algorithm to the correct model.

The KFs in Figure 2 and the PPE are designed in discrete-time. Thus, all continuous-time equations are transformed into discrete-time using a sampling interval of $T_s = 0.001$ seconds (including the calculation of the correct covariances for the *discrete-time* plant white noise sequence $\xi(\cdot)$ and measurement noise $\theta(\cdot)$). All subsequent stochastic simulations use the same sampling interval as well.

The number N of models has been derived (see Table I) and each of the N KFs must use the information contained in eqs. (2.1) through (2.7), (2.10) and (2.11). Once more, unmodeled dynamics are ignored in the KF designs. The only remaining parameter that must be specified to design KF #j, associated with Model #j, j=1,2,. . . ,7, is the correct numerical value of the stiffness parameter k_1^j; $j = 1, 2, \ldots, 7$, to be used in eq. (2.6). Following the procedure

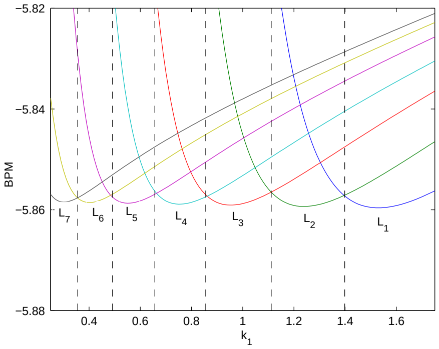


Figure 12. The nominal value of k_1 for each KF is at the minimum of the associated L_i (BPM) curves.

described in detail in [1], these numerical values are obtained by computing the *"Baram Proximity Measure (BPM)"* [11] so that the BPMs agree at the boundary of adjacent models. The outcome of this process (requiring the use of a genetic algorithm) is shown in Figure 12 and they are

$$k_1 \in \mathcal{K}_1 = \{1.53, 1.23, 0.95, 0.73, 0.56, 0.39, 0.31\} \quad \text{for KF \#1-KF \#7} \tag{4.1}$$

Eq. (4.1) means that in designing KF#1 we use $k_1 = 1.53$, in designing KF#2 we use $k_2 = 1.24$ and so on.

From the information above one can design each one of the seven KFs and, in addition, compute the constant, steady-state, residual covariance matrices S_j; $j = 1, 2, \ldots, 7$ required by the PPE algorithm [1] of eq. (4.2) below.

$$P_k(t+1) = \left[\frac{\beta_k \left(e^{-\frac{1}{2} r'_k(t+1) S_k^{-1} r_k(t+1)} \right)}{\sum_{j=1}^{N} \beta_j \left(e^{-\frac{1}{2} r'_j(t+1) S_j^{-1} r_j(t+1)} \right) \cdot P_j(t)} \right] P_k(t) \tag{4.2}$$

where:

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

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

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

As explained in detail in [1], under certain ergodicity and stationarity assumptions, one of the posterior probabilities will converge almost surely to unity and will "identify" the model closest to the true plant, i.e. the one with smallest BPM – see Fig. 12.

5. RMMAC Stochastic Simulations

In this section we present a few representative simulations[‡] that demonstrate the superior performance of the RMMAC system as compared to the best non-adaptive GNARC design. We examine the behavior of the RMMAC when the theoretical assumptions discussed in [1, 11], regarding ergodicity and stationarity are satisfied and guarantee the asymptotic identification of the correct model. We also present simulations when the theoretical assumptions are "mildly violated" and show that the RMMAC also results in very good disturbance-rejection. Furthermore, we demonstrate that the RMMAC recovers following a temporary forced instability configuration; see the discussion leading to Table II. Finally, we demonstrate that the RMMAC works very well when the uncertain spring stiffness changes slowly as a function of time in a sinusoidal manner.

5.1. Normal operating conditions

The simulations were implemented in discrete time using a zero-order hold with a sampling time of $T_s = 0.001$ secs[§]. In addition, the correct variances of the discrete-time white noise sequences, $\xi(\cdot)$ and $\theta(\cdot)$, were calculated and used to design the seven KFs and the posterior probability evaluator (PPE); these discrete-time numerical values were used in all Monte Carlo (MC) simulations in the sequel.

Among the numerous simulations done, only some representative results of RMMAC stochastic responses under normal conditions are shown here. Unless stated otherwise the results are averaged over 5 Monte Carlo (MC) runs. Furthermore, in all simulations we used a time-delay of 0.01 secs.

The dynamic behavior of the seven posterior probabilities when the true $k_1 = 1.65$ (stiff spring), well inside the subinterval (Model) #1, and the corresponding outputs for the RMMAC and the GNARC systems are shown in Figure 13. The correct model (Model #1) is identified in less than 40 secs. The improvement in disturbance-rejection by the RMMAC is evident as shown in Figure 13(b).

Figure 14 shows a different simulation when the true $k_1 = 0.45$ (soft spring), inside the subinterval #6 but close to subinterval #7 (see Table I). The correct model (Model #6) is identified in about 100 secs. Note that, in this case, the RMMAC takes longer to resolve the ambiguities between Models #6 and #7, because k_1 is very close to the boundaries of Models #6 and #7.

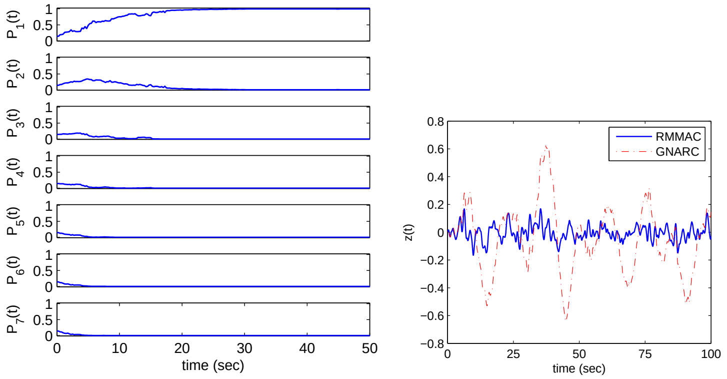
By comparing the performance of the RMMAC vs GNARC we see that we have much better disturbance-rejection for a hard spring (Fig. 13(b)) than for a soft spring (Fig. 14(b)). This confirms the performance predictions discussed in Section 3.

At this point, it is also of interest to compare the RMMAC with the case of correct identification to show that the probabilistic averaging "works".

In this example, the true spring value is $k_1 = 0.635$ in Model #3 and also close to Model #2 (see Table I). Thus one would expect a long period of time for the posterior probabilities to converge. Thus, we shall see the difference between the probabilistically-blended control signals

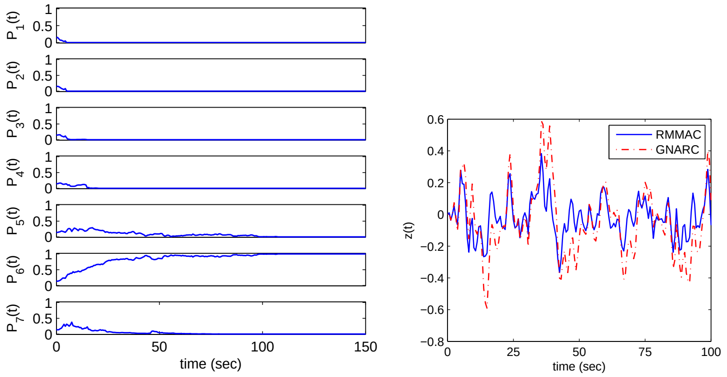
[‡] Out of several dozen conducted [2].

[§] The small sampling time was necessary because the compensators (see Figure 9) have a large bandwidth.



(a) Posterior probabilities of PPE (b) Output performance $z(t)$

Figure 13. Monte Carlo stochastic simulation results when $k_1 = 1.65$ in Model #1.

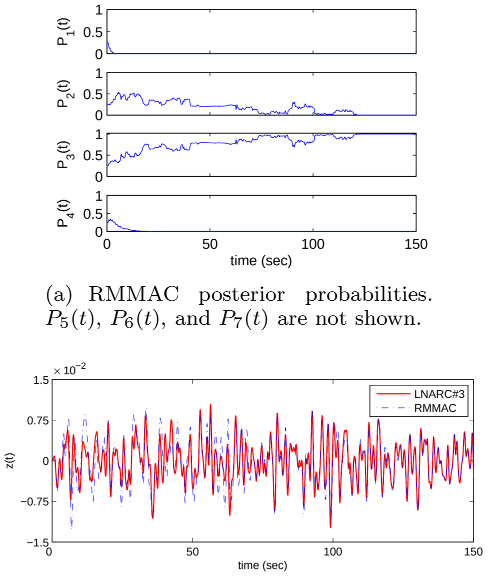


(a) Posterior probabilities of PPE (b) Output performance $z(t)$

Figure 14. Monte Carlo stochastic simulation results when true spring constant is $k_1 = 0.45$ in Model #6, but very close to Model #7.

(due to the LNARC #3 and LNARC #2) in the RMMAC and that of perfect identification from $t = 0$ only using the control signal of LNARC #3.

In Figure 15 we compare the performance of the RMMAC obtained with $k_1 = 0.635$ against the performance obtained when we use the perfect compensator LNARC #3 only, i.e. we fix $P_3(t) = 1$ and set $P_1(t) = P_2(t) = P_4(t) = 0 \; \forall t = 0, 1, 2, \ldots$. Obviously, in this case we have "perfect" identification from the start, i.e. t=0. Figure 15(a) shows the posterior probabilities. Note that, from Figure 15(b), *the probabilistic averaging, over a period of about 120 secs, results in insignificant performance deterioration* even if the convergence of the posterior probabilities happened only after about 120 secs. Of course, the performance response of the "perfect" compensator is *slightly better* than that of the RMMAC in the transient period as shown in Figure 15(b). However, after $t = 120$ secs the posterior probabilities converge, and, naturally, both the RMMAC and the "correct" LNARC #3 behave in an identical manner.



(a) RMMAC posterior probabilities. $P_5(t)$, $P_6(t)$, and $P_7(t)$ are not shown.

(b) Performance output $z(t)$

Figure 15. Performance comparison of using the LNARC #3 (perfect identification from t=0) with the RMMAC for $k_1 = 0.635$. Numerical averages for 5 MC simulations are presented; there is no sensor noise.

Mismatch Model/Compensator Instability We next evaluate the RMMAC response when it is forced to be unstable at time $t = 0$. Recall that Table II summarized the mismatch Model/LNARC stability cases of the designs.

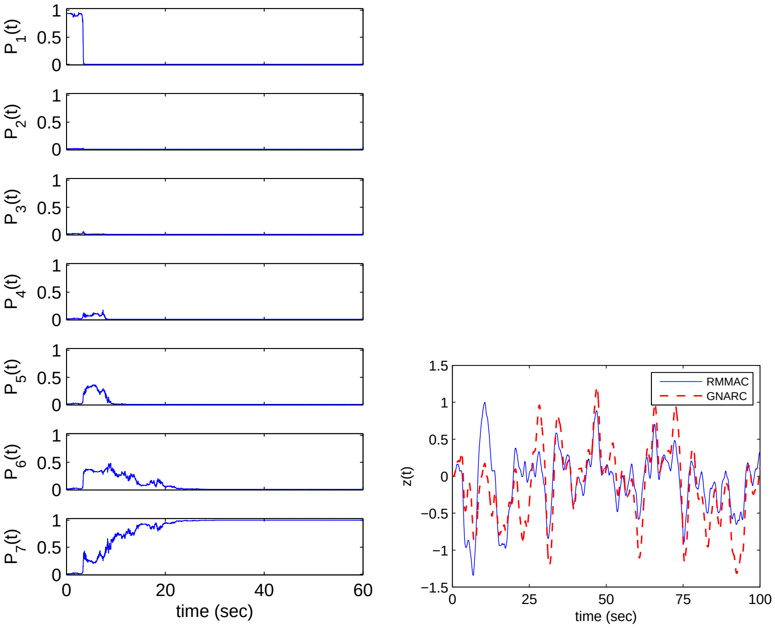
Figure 16 illustrates a typical result selected from several MC simulations. In Figure 16 the true value of k_1 is 0.3 in Model #7. Referring to Table II, the compensator LNARC #1, $K_1(s)$, with Model #7 corresponds to an unstable closed-loop pair. To impose this forced instability, the initial values of the probability vector are selected to be

$$P_1(0) = 0.94,\ P_2(0) = \ldots = P_7(0) = 0.01$$

so that initially, at $t = 0$, the RMMAC system is *forced* to be unstable. However, as depicted in Figure 16, the RMMAC recovers to a stable (and correct) configuration. Figure 16(a) shows that the "correct probability" $P_7(t) \rightarrow 1$ within about 20 secs, starting from its initial value $P_7(0) = 0.01$; the other six probabilities converge to zero within 10 secs as well. Figure 16(b) shows the output performance response in which the RMMAC recovers, after a brief oscillatory response, and returns to its predicted disturbance-rejection.

5.2. *RMMAC simulation results under "mild violations" of the theory*

It is important to evaluate any control system design when the theoretical assumptions are violated to a certain degree, because this will always happen in practice. In this subsection we examine the RMMAC performance when the disturbance is not generated by white noise and when the sensor noise is not white.



(a) Posterior probabilities of PPE

(b) Output performance $z(t)$

Figure 16. Forced mismatch Model/LNARC instability, at $t = 0$, with stochastic plant disturbance and white sensor noise; $k_1 = 0.3 \in M\#7$;
$$P_k(t)|_{t=0} = [.94\ .01\ .01\ .01\ .01\ .01\ .01]^T.$$

5.2.1. Step plant disturbance In this simulation, we used a deterministic periodic square-wave plant disturbance, $\xi(t) = \pm 2.0$, with a period of 60 secs, rather than pure plant white noise. The sensor noise was white as in Section 5.1. The KFs in the RMMAC were NOT aware of the square-wave disturbance; they continued to use eqs. (2.1), (2.2), and (2.3) to model the disturbance dynamics. As before, we used a time-delay of 0.01 secs.

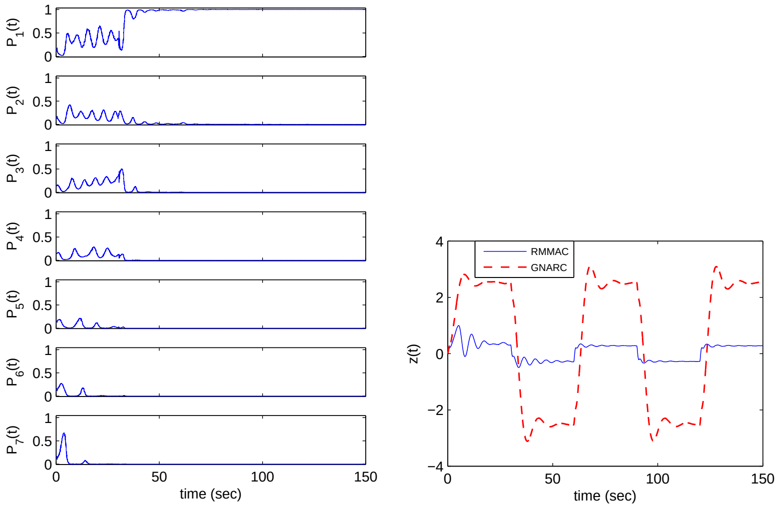
We evaluated the RMMAC performance over a wide variety of operating conditions, for different values of the uncertain spring constant [2]. In all simulations we noted that the RMMAC worked extremely well and we did not observe any instabilities.

Figure 17 shows a representative simulation using the value $k_1 = 1.75$, which is in Model #1. The sensor noise is also a deterministic sinusoidal measurement noise as $\theta(t) = 0.001\sin(10t)$. Figure 17(a) shows that the "correct" probability $P_1(t) \to 1$ within 50 secs. The improvement in disturbance-rejection is obvious from Figure 17(b). The control signals are noisier and slightly larger than those of the GNARC feedback system (not shown).

Sinusoidal sensor noise In this case, the disturbance $d(t)$ was a stochastic process as in the normal case. However, instead of using white measurement noise, we used in eq. (2.4) a deterministic sinusoidal measurement noise given by

$$\theta(t) = 0.001\sin(50t) \tag{5.1}$$

The KFs in the RMMAC were designed under the assumption that the sensor noise was



(a) Posterior probabilities transients

(b) Output performance $z(t)$

Figure 17. Simulation results of the RMMAC performance when the actual disturbance $d(t)$ is a ±2 periodic (filtered) square-wave with the period of $T = 60$ secs, and $k_1 = 1.75$ in Model #1.

white, as in eq. (2.11).

Once more, we performed several MC simulations using different values for the unknown spring constant. In all cases the RMMAC design worked very well and we did not observe any instabilities. Figure 18 shows a representative average of 10 MC runs using the value $k_1 = 1.75$ in Model #1. As illustrated in Figure 18(a) it takes 20 secs for the "correct" probability $P_1(t) \rightarrow 1$. During this 20 secs time-interval there is *significant blending* of the "local" control signals $u_1(t)$ and $u_2(t)$ by the RMMAC. Nevertheless, the RMMAC still delivers excellent disturbance-rejection (Figure 18(b)) with small controls (Figure 18(c)) which "show" the presence of the sinusoidal sensor noise. Thus, control probabilistic averaging did not produce any serious problems; on the contrary, the RMMAC still outperforms the GNARC system.

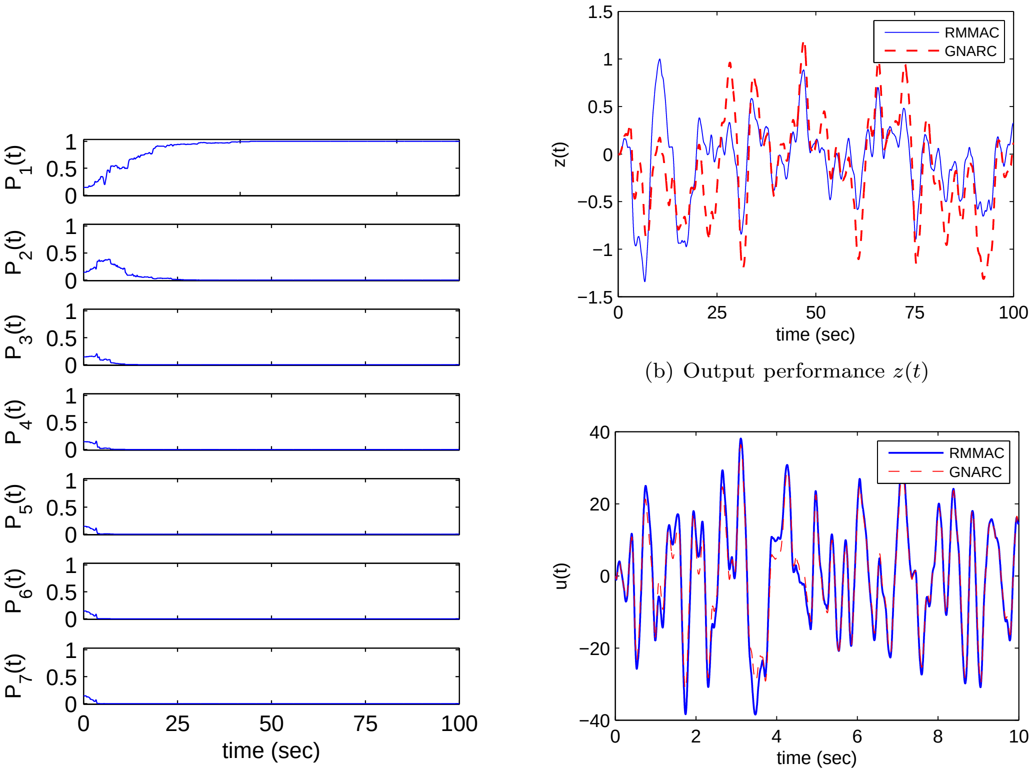
We conclude from the results of this subsection that, at least for this test case, the RMMAC system still works extremely well, even if we mildly violate some of the key theoretical assumptions. It continues to outperform the GNARC design, and seems somewhat "robust" to such mild violations of the theory.

5.2.2. Slowly varying parameter It is important to test any adaptive system when the uncertain parameter changes with time. As stressed in [1], the "slow" time-variation of uncertain parameters is one of the key reasons for using adaptive control.

We next evaluate the performance of the RMMAC for time-varying spring stiffness. In the following simulation the unknown spring constant k_1 is assumed to vary between its lower- and upper-bound in a sinusoidal manner as follows

$$k_1 = 1 - 0.75\ \cos(0.01t) \tag{5.2}$$

Figure 19 shows a representative simulation result with this slowly time-varying spring constant. Once more the RMMAC outperforms the GNARC design.



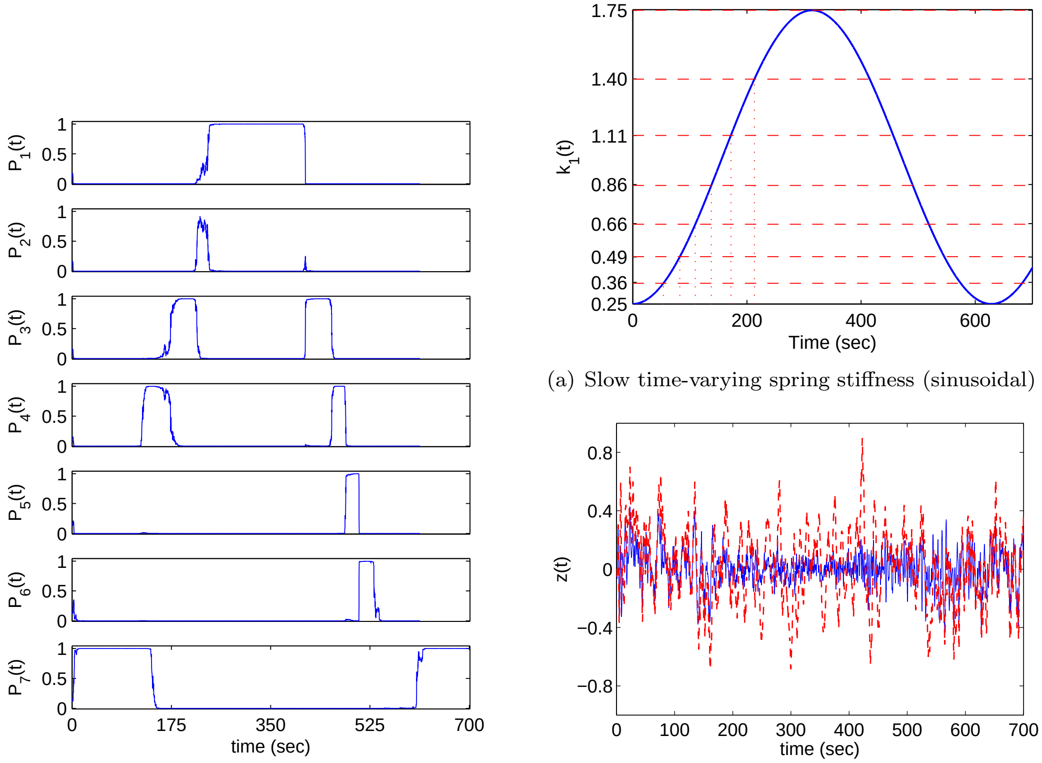
(a) Posterior probabilities of PPE

(b) Output performance $z(t)$

(c) Control signal $u(t)$

Figure 18. Monte Carlo simulation results of the RMMAC performance for sinusoidal sensor noise when $k_1 = 1.75$ in Model #1.

Figure 19(a) shows the slow time-varying spring used in the simulation. Figure 19(b) shows the behavior of the posterior probabilities that "track" the sinusoidal spring. Figure 19(c) compares the output of the GNARC with the RMMAC. In Figure 19(c) note that the RMMAC performance is much better during the times that the spring constant has large values, i.e. it is "stiff" in Model #1, as expected by the predictions of Section 3; see Figure 11. In many other simulations [2] (not shown) the RMMAC also showed better disturbance-rejection compared with the best GNARC under such mild violations, i.e. by introducing slowly varying springs.



(a) Slow time-varying spring stiffness (sinusoidal)

(b) Posterior probabilities of PPE

(c) Output performance $z(t)$

Figure 19. The RMMAC performance with slow sinusoidal $k_1(t)$ variation. Only one MC run is shown.

6. Severe Violations Of The Theoretical Assumptions

In this section we discuss the behaviour of the RMMAC system when we violate the theoretical assumptions in a very significant manner[¶].

First, we examine the RMMAC response when we *intentionally* force a mismatched model/LNARC instability (far more severe than that associated with Figure 12 in Section 5) over a long period of time, i.e. during several consecutive measurements. Second, we examine the RMMAC behaviour when the plant disturbances are, sometimes, far larger than those used to design the Kalman Filters (KFs).

In both of the above situations, the residuals are very much larger that their predicted ranges (by the residual covariance matrices) – see eq. (4.2). This leads to “confusion” in the identification process and causes the posterior probabilities to rapidly switch among

[¶] “Theories have limitations; stupidity does not!” We strongly believe that proponents of any adaptive methodology should report both the “good” and the “bad” behaviour of their designs and explain the causes of inferior performance or even instability.

the different models, thereby severely degrading the RMMAC performance [1,2]. We then demonstrate that use of the variant RMMAC/XI architecture [1,2] yields again a superior performance.

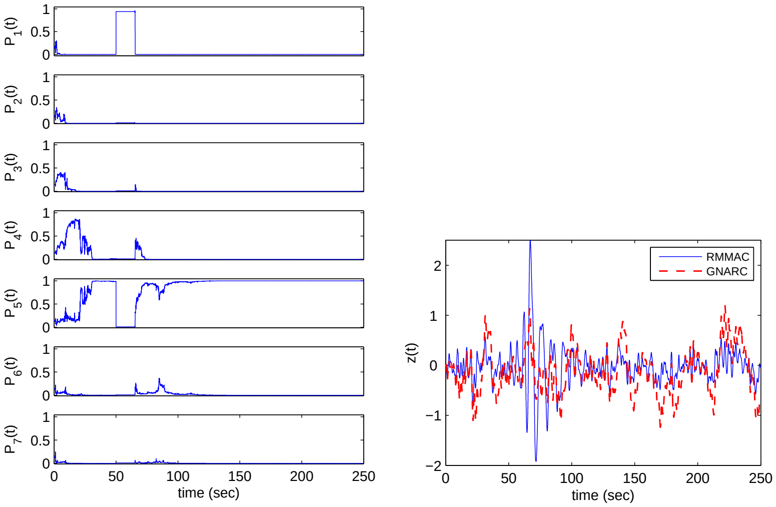
6.1. Severe enforced mismatched Model/LNARC instability

In this simulation we suppose that the true spring stiffness is k_1=0.6 . From Table I we know that k_1 is in Model #5. From Table II we note that if we use LNARC #1 in the feedback loop with Model #5 we always have instability. We assume that ξ and θ are white so the system operates under normal stochastic conditions.

To impose this forced instability, the values of the probability vector at times $t \in [50, 65]$ secs are kept fixed as

$$P_1(t) = 0.94,\; P_2(t) = \ldots = P_7(t) = 0.01, \qquad 50 \leq t \leq 65 \text{ secs}$$

so that at $t = 50$ the RMMAC system is forced to use the unstable compensator, LNARC#1. However, as shown in Figure 20, the RMMAC recovers to a stable configuration.



(a) Posterior probabilities transients (b) Output response

Figure 20. Forced mismatch instability for $50 \leq t \leq 65$ secs. The true value is $k_1 = 0.6$ in Model #5. Note the rapid recovery of the RMMAC after the forced instability, i.e. $P_5(t) \rightarrow 1$. Only one MC is shown.

Clearly, Figure 20 illustrates that during this period of enforced instability, and for some time thereafter, the RMMAC disturbance-rejection is much worse than that of the GNARC design. The enforced instability causes a highly oscillatory behaviour in the RMMAC system. However, after the transient oscillations die-out the RMMAC yields once more superior disturbance-rejection.

Since the sampling interval used is 0.001 secs, this means that the forced instability persists for 15,000 measurements! The probability of such an event being caused by the plant and/or sensor white noises, coming from the far-away tails of their Gaussian distributions over 15,000 consecutive measurements is truly infinitesimal. Therefore, we should not expect such instability to occur in a practical situation,

6.2. Highly Uncertain Plant Disturbance Environment

Uncertainty about the strength of the plant disturbances is a very common occurrence in engineering systems. For example, in aircraft applications most of the time the aircraft is subject to light to moderate turbulence; however, there may be short-term occurrences of very heavy turbulence. Thus, the performance of any disturbance-rejection feedback control system, adaptive or not, must be evaluated for both "normal" and "abnormal" disturbance environments. This is the reason that we investigate the RMMAC performance under such conditions.

The superior performance of an RMMAC design will be intimately related to the accuracy of the (MMAE) identification subsystem. In particular, the design of each of the seven KFs in Figure 2 is strongly dependent upon the modeling of the random disturbance $d(t)$ – see eqs. (2.1) to (2.3) – and in particular upon the intensity Ξ of the plant white noise. The value of Ξ determines the seven KF gain matrices, residual matrices and the (real-time) residuals. These, in turn, impact the PPE calculations and hence the convergence of the posterior probabilities.

We first evaluate the RMMAC performance when we change the intensity Ξ of the plant white noise in a periodic manner. Suppose that the plant disturbance intensity is switching between two different values $\Xi_{act} = 1$ and $\Xi_{act} = 100$ as shown in Figure 21(a). The RMMAC KFs design used $\Xi = 1$ as before. Suppose that the true spring is $k_1 = 0.3$ in Model #7. We wish to investigate if the posterior probabilities will oscillate. Table II shows that the compensators LNARC#1, #2, and #3 would create unstable closed-loop systems with this spring value. The simulation results are shown in Figure 21 for a single Monte-Carlo run.

The transients of the RMMAC posterior probabilities are shown in Figure 21(b). It can be seen that there is a model selection confusion (oscillation) among the posterior probabilities during the intervals of the *strong* disturbance input. Figure 21(c) compares the RMMAC output with that of the GNARC. Note that the posterior probabilities do not converge, when $\Xi_{act} = 100$. Thus, such an unknown (and abnormal) plant noise intensity has harshly violated the assumptions of the RMMAC. Also note (Figure 21(c)) that the RMMAC shows poorer performance in comparison to the GNARC during the periods of the "abnormal" disturbance.

It is interesting to examine closely the behavior of the posterior probabilities in Figure 20(b). During the "normal" disturbance intervals, denoted by NORM,

$$\text{NORM} \equiv \cup\, [0{,}50],[100{,}150],[200{,}250] \tag{6.1}$$

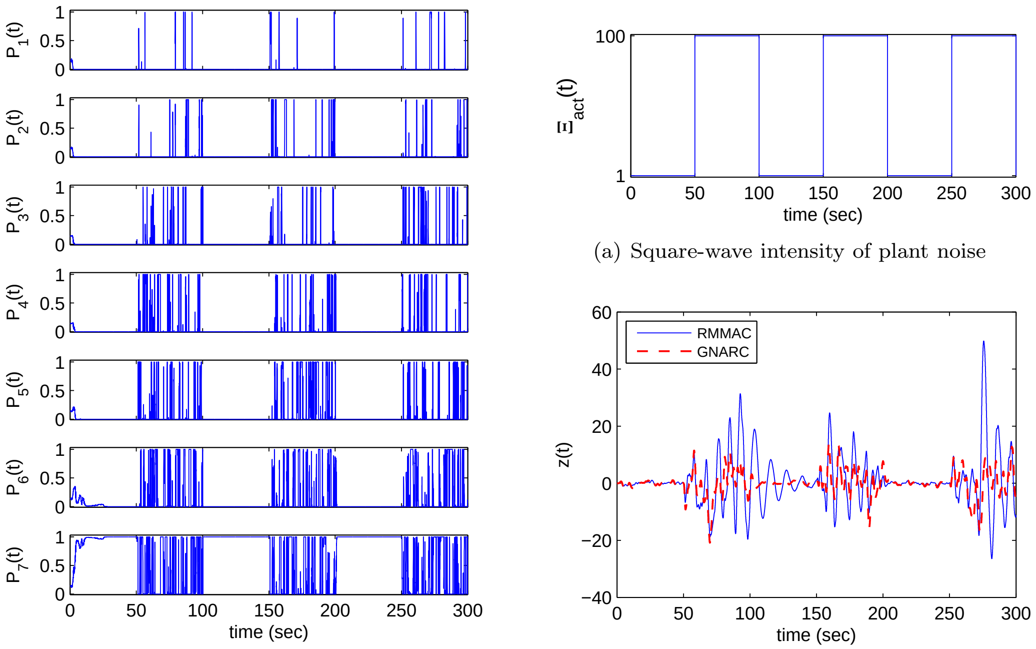
the correct posterior probability quickly converges to the correct Model #7, i.e.

$$P_7(t) \to 1 \quad \forall t \in \text{NORM} \tag{6.2}$$

During the "abnormal" disturbance intervals, denoted by ABNORM,

$$\text{ABNORM} \equiv \cup\, [50{,}100],[150{,}200],[250{,}300] \tag{6.3}$$

there is "confusion" in the behavior of the posterior probabilities. In point of fact, they temporarily place the "destabilizing" compensators, LNARC #1, #2, or #3, in the feedback



(a) Square-wave intensity of plant noise

(b) Posterior probabilities

(c) Performance output $z(t)$

Figure 21. Case 1B. The RMMAC performance under severe assumption violations. The plant disturbance intensity $\Xi = 1$ was used to design the KFs, but in the simulations it switches in $\Xi_{act} \in \{1, 100\}$. The value of the spring constant is $k_1 = 0.3$ in Model #7. Note that the RMMAC performance is worse than that of the GNARC, Figure 19(c).

loop. This, in turn, causes the oscillations in $z(t)$, Figure 20(c), when $t \in$ ABNORM which tend to die out when $t \in$ NORM, but sometimes not fast enough. The bottom line is that *the RMMAC disturbance-rejection is worse than that of the non-adaptive GNARC system.*

The explanation for the behavior shown in Figure 20 is quite simple. All seven KFs were designed in Section 4 under the assumption of "normal" disturbance, i.e. $\Xi = 1$. The Posterior Probability Evaluator (PPE) – see eq. (4.2) – determines the posterior probabilities by comparing the actual residuals, $r_k(t)$, with their precomputed covariance matrices, S_k. When the "abnormal" intensity, is used, the resulting residuals are much larger that those predicted by their associated residual covariances and the PPE algorithm of eq. (4.2) leads to the "confusion" seen in Figure 20(b). The KF gain matrices are simply not large enough and the actual measurements are not sufficiently weighted in the update cycle of the KF.

6.3. The RMMAC/XI Architecture

One possible solution to the problems discussed above is to adopt a *very conservative approach*, i.e. design the KFs for the "worst" possible performance, i.e. always use $\Xi = 100$. This was analyzed in [2] and we have found that it leads to *very slow convergence* of the posterior probabilities in the "normal" situation. Such slow convergence of the posterior probabilities would not be desirable, especially for the case of slowly time-varying parameters (see Section 5).

Such considerations led us to develop the so-called RMMAC/XI architecture [1] shown in Figure 22.

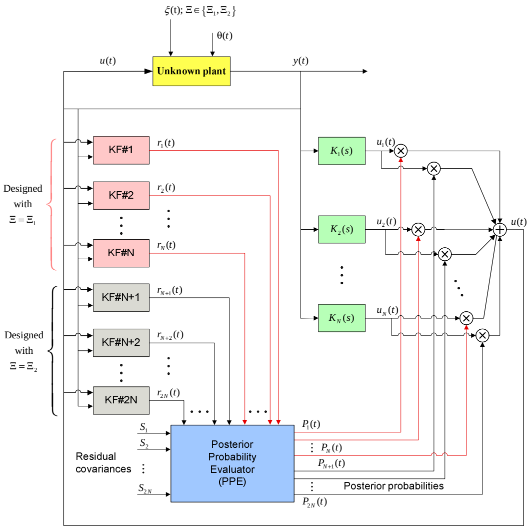
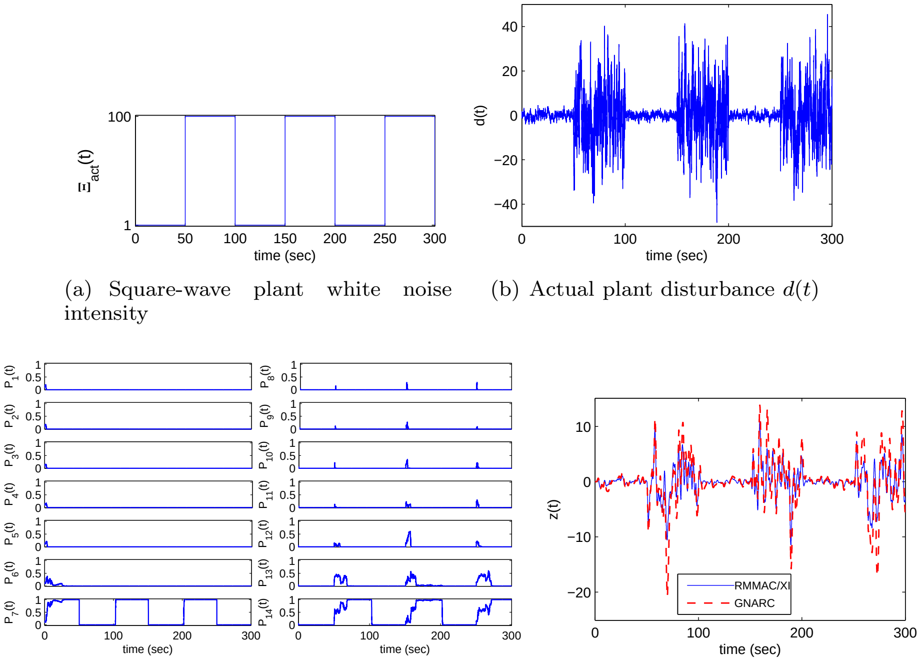


Figure 22. The general structure of the RMMAC/XI architecture. For the specific case study at hand, N=7, $\Xi_1 = 1$ and $\Xi_2 = 100$.

Basically, we design two sets of Kalman filters KF #1 to #7 for the first set using the "normal" white noise intensity $\Xi_1 = 1$ and seven more, KF #8 to #14 for the second set using the "abnormal" white noise intensity $\Xi_2 = 100$. KFs #1 to #7 were designed already in Section 4. The details of KFs #8 to #14 are not given here; they were redesigned using $\Xi_2 = 100$ and their nominal points were recomputed using the Baram Proximity Measure (BPM) in a manner completely analogous to that of Figure 12. Although the recomputed BPMs of the second set using $\Xi_2 = 100$ (KFs #8 to #14) were different than those of the first set BPMs using $\Xi_1 = 1$ (KFs #1 to #7) as shown in Figure 12, the calculations produced the similar values as in eq. (4.1) for the nominal k_i for KF #8 to KF #14. In essence, we have introduced two hypotheses in the "identification subsystem": the first associated with $\Xi_1 = 1$ and the second corresponding to $\Xi_2 = 100$. The bottom line is that in the RMMAC/XI architecture of Figure 21 we have 14 different (on-line) residuals, 14 different residual covariance matrices, and 14 different posterior probabilities. Thus, we have 14 different "models".

The reader should note that in the RMMAC/XI architecture of Figure 21 we use *exactly the same* 7 LNARCs which were designed in Section 3. The reason is that, as explained in more detail in [1], the design of the LNARCs *does not* depend on the value of Ξ; they are only influenced by the disturbance dynamics, eq. (2.3) , which were not changed. This is why the LNARCs do not have to be redesigned.

We shall next present a stochastic simulation comparing the response of the RMMAC design discussed in Section 6.2 with that of the RMMAC/XI design. We have used the same square-wave intensity matrix shown in Figure 20(a) and the same seed number. The true value of the



(a) Square-wave plant white noise intensity

(b) Actual plant disturbance $d(t)$

(c) Posterior probabilities

(d) The GNARC and the RMMAC/XI outputs

Figure 23. The RMMAC/XI performance when Ξ is a periodic "square-wave" as in Figure 21(a). The true spring constant is $k_1 = 0.3$ in Model #7 when $\Xi = 1$ and in Model #14 when $\Xi = 100$. Note that the "correct" posterior probabilities $P_7(t)$ and $P_{14}(t)$ quickly respond to the changes in Ξ. The RMMAC/XI yield better disturbance rejection to that of the GNARC.

spring uncertainty k_1=0.3 as in Section 6.2. We stress that in view of the additional hypothesis

$$\begin{aligned} k_1 &= 0.3 \in \text{ Model \#7 for } \Xi = 1 \\ k_1 &= 0.3 \in \text{ Model \#14 for } \Xi = 100 \end{aligned} \tag{6.4}$$

Figure 22 shows the results of the stochastic simulation. Figure 22(a) shows the square wave intensity; it is identical to that shown in Figure 20(a). Figure 22(b) shows the actual disturbance, $d(t)$, applied to the MSD system based upon the intensities of Figure 22(a) and eqs. (2.1) to (2.3); the periodic bursts of strong disturbance are evident. Figure 22(c) shows the time evolution of the 14 posterior probabilities. Figure 22(d) demonstrates that the RMMAC/XI system has now superior disturbance-rejection vis-à-vis the GNARC system (unlike Figure 20(c)).

The key observation is to notice the fast (and correct) convergence of the posterior probabilities $P_7(t)$ and $P_{14}(t)$ in Figure 22(c) to the correct models – see eq. (6.4) – following the intensity changes of Figure 22(a). There is no longer any "confusion" among the posterior probabilities, unlike those noted in Figure 20(b). This demonstrates how simple variants of the RMMAC can be derived by adding additional hypotheses and it represents an important "flexibility" of RMMAC-like architectures.

Finally, we illustrate in Figure 23 the comparisons of the disturbance-rejection of the RMMAC and RMMAC/XI architectures. Figure 23 is simply obtained from Figures 20(c) and 22(d). Obviously, for the first 50 seconds both yield the same response. Later, their responses are very different. The differences would be amplified if we have shortened the period of the Ξ square-wave.

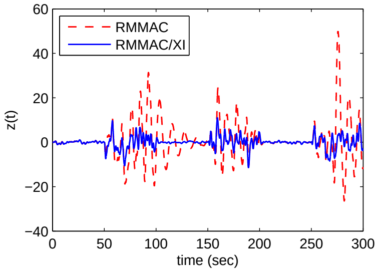


Figure 24. The RMMAC/XI performance compared to the standard RMMAC subject to the unknown plant noise intensities. The true spring is $k_1 = 0.3$.

6.4. Computation Times

We next summarize the computation time for a single complete stochastic simulation. The simulations were carried out in a Toshiba Satellite Pro laptop with an Intel Pentium M 1.5GHz processor, running under Windows XP Pro SP2. The sampling time was 0.001 secs (due to the large bandwidth of the LNARCs shown in Figure 8).

A single MC run of the RMMAC discussed in Section 5, involving real-time simulation of the plant, the 7 KFs, the PPE update every 0.001 sec, and the 7 LNARCs was carried out *approximately* in real-time. For the RMMAC/XI simulation of Section 6.3, involving real-time simulation of the plant, the 14 KFs, the PPE update every 0.001 sec, and the 7 LNARCs was carried out *approximately* 1.2 times real-time.

7. Concluding Remarks

The simulation results presented in this paper confirm our previous findings [1] that the RMMAC and RMMAC/XI significantly outperform the best non-adaptive (GNARC) design yielding superior disturbance-rejection design in the prespecified frequency region.

The RMMAC MSD test design in [1] was much easier compared to that considered in this paper. Here we excited the lightly-damped modes of the MSD system over a wider frequency range: $\omega \in [0, 0.1]$ rad/sec in [1] vs $\omega \in [0, 3.0]$ rad/sec in this paper. We also demanded superior disturbance-rejection over the same larger frequency region. This change in the disturbance dynamics and the posed performance specification had a very significant impact upon the benefits of adaptive control (comparing the GNARC and FNARC) and the number of models required for the RMMAC implementation; seven in this case compared with four in [1]. This reinforces our design philosophy that *explicit performance requirements must be well defined prior to the design of any adaptive system.*

Finally, we have again determined that the RMMAC is quite “robust” with respect to *mild violations* of the underlying theoretical assumptions [1,2,11]. On the other hand, *severe violations* of the theoretical assumptions can cause problems. Enforced long-term instability is one example, but it is quite unlikely in a stochastic environment. The RMMAC, however, can yield poor performance (but not long-term instability) if the magnitude of the disturbances occasionally is quite different from their “normal” level used to design the Kalman filters. In such cases, the RMMAC/XI architecture can greatly mitigate these performance problems.

acknowledgment

This work is based upon S. Fekri’s doctoral thesis [2] and was supported in part by FCT - ISR/IST plurianual funding through the POS Conhecimento Program in cooperation with FEDER and by project MAYA-Sub of the AdI.

REFERENCES

[1] S. Fekri, M. Athans, and A. Pascoal, “Issues, progress and new results in robust adaptive control,” *Int. J. of Adaptive Control and Signal Processing*, in press.

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

[3] J. Doyle, “Analysis of feedback systems with structured uncertainties,” *IEE Proc.*, vol. 129, pp. 242–250, Nov. 1982.

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

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

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

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

[8] G. Balas *et al.*, *μ-Analysis and Synthesis Toolbox of* MATLAB*, User’s Guide*, 4th ed. The MathWorks Inc., June 2001.

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

[10] H. Sorenson (ed.), *Kalman Filtering: Theory and Applications.* IEEE Press, 1985.

[11] Y. Baram, “Information, consistent estimation and dynamic system identification,” PhD. dissertation, MIT, Cambridge, MA, USA, 1976.

APPENDIX

GNARC and LNARC design specifications

A significant disadvantage of $D-K$ or $D, G-K$ iterations (complex or mixed-μ techniques) is that these design methods generally tend to produce compensators with the same order as the generalized plant. It is due to the fact that the frequency-dependent weights are included in the design framework in order to achieve the desired performance characteristics and to account for the structure in the uncertainty. Further, the dimensionality of the D and G scales increase the order of the generalized plant and, hence, of the $\mathcal{H}_\infty$ compensator [4].

One possible solution to this problem could be to constrain the order of compensator *a priori* in the design process by mainly fixing the D and G scales. In this case, our designs showed that we have unacceptable performance. So, we do not recommend *a priori* constraints on the order of compensator, and individually on the D scales.

Another approach to avoid compensators of high dimension is to use model order reduction techniques on the compensator realization that help one to find less complex low-order approximations to controller models. These can be significantly simplified to ease implementations and simulations efforts. Nonetheless, these techniques should consider the properties of the closed-loop system when reducing the order of the compensator in order to maintain both stability- and performance-robustness requirements for all possible values of uncertain parameters – this can be done by checking $\mu < 1$ again after doing the order reduction.

In defining the robust compensators using the mixed-μ toolbox of MATLAB [9], we have applied the order reduction to the compensators designed in this paper. Among the most important types of model reduction methods are based on the truncated balanced realization of the system state-space model which entails balancing the observability and controllability Grammians. The result is truncated to retain all Hankel singular values greater than a (possibly small) number. This can achieve a reduced-order robust compensator that preserves the major characteristics of the closed-loop system characteristics with minimal sacrifices to robust performance.

For the example investigated in this paper, it is also possible to reduce the controller order to 14 for LNARCs and 22 for GNARC, using truncated balanced realizations, and still satisfy robust stability and robust performance. It is stressed that the original order of GNARC/LNARCs compensators using D, G–K iteration are about 42 or more for this example. Table 7 shows other parameters that we used in the GNARC/LNARCs designs.

Table III. Parameters used in GNARC/LNARCs design using mixed-μ synthesis

MAXORD_D	40	% *Max order of D-scale*
MAXORD_G	10	% *Max order of G-scale*
MAX_ITER	25	% *Max number of D,G-K iteration*
DELTA_SET	[-1 0; 1 1]	% *Real spring stiffness & complex time-delay uncertainties*
FICT_BLK	[2 2]	% *Fictitious delta block for performance robustness*
BLK_DGK	[DELTA_SET ; FICT_BLK]	% *Delta block used in the D,G-K iteration*
OMEGA_DGK	logspace(-2,2,70)	% *Frequency range used in all designs*