Model Reference Adaptive Control Using Non-Euclidean Gradient Descent

Sang-Heon Lee, Robert Mahony, and Il-Soo Kim

Abstract In this paper, a non-linear approach to a design of model reference adaptive control is presented. The approach is demonstrated by a case study of a simple single-pole and no zero, linear, discrete-time plant. The essence of the idea is to generate a full non-linear model of the plant dynamics and the parameter adaptation dynamics as a gradient descent algorithm with respect to a Riemannian metric. It is shown how a Riemannian metric can be chosen so that the modelled plant dynamics do in fact match the true plant dynamics. The performance of the proposed scheme is compared to a traditional model reference adaptive control scheme using the classical sensitivity derivatives (Euclidean gradients) for the descent algorithm.

Keywords: adaptive control; discrete-time system; riemannian geometry

I. Introduction

Model reference adaptive control (MRAC) is one of the main approaches to the adaptive control of servo systems. In MRAC, the desired performance of a closed-loop system is specified in terms of a reference model and the controller parameters are adjusted to minimise a given error function.

Historically, the MIT rule [1] was the parameter adaptation mechanism used for the first published application of MRAC. In this paper, the control parameters are updated according to a continuous-time ordinary differential equation (ODE) generated by setting the time derivatives of the parameter equal to the negative gradient of a performance index. The performance index used was the integral squares of the response error, the difference between the actual closed-loop system output and the reference model output. This gradient is commonly known as the sensitivity derivative of the system. To balance between system stability and the adaptation speed, an adaptation gain is introduced into the control parameter ODE. This adaptation gain plays a crucial role in system stability, ensuring that the dynamics induced in the controller by the adaptation rule do not interfere with the system dynamics. Unfortunately, it is usually not possible a priori to choose a suitable value for the adaptation gain. Consequently, the MIT rule for the adaptation of control parameters suffers from a fundamental stability problem.

Among many subsequent approaches to adaptive control, the schemes that retain the closest resemblance to the MIT rule are those based on the Lyapunov design [2][3]. These designs have the advantage that they take into consideration of the combined system and parameter adaptation dynamics and design a controller to guarantee stability of the system [4][5]. (A recent overview of design procedure is given in [6-Chapter 5]) However, it has proved difficult to generate valid Lyapunov functions for the full system dynamics, and the classical Lyapunov design relies on the assumption that the adaptation dynamics do not evolve quickly compared to the system dynamics. Once again the gain of the adaptation dynamics plays a crucial role in determining system stability. To provide an estimate of when MRAC systems designed using Lyapunov techniques are practical, several authors [7][8] have used the averaging theory to produce rigorous stability results. Classical adaptive control designs were based around linear design techniques. However, work in the late eighties [9][10][11] showed that highly non-linear and even chaotic behaviour could result from relatively simple MRAC schemes. This perspective has led some people to view MRAC as a fully non-linear design problem. Authors [12[13][14][15] have made some advances in explicit non-linear adaptive control design methodology. However, much of this design methodology is still based around Lyapunov concepts. A fundamental limitation of Lyapunov theory is the difficulty of finding a suitable Lyapunov function for a given system.

In this paper, we present a non-linear approach to the design of model reference adaptive control schemes for linear systems. In our approach, we begin with a full non-linear system model, combining the non-linear parameter adaptation dynamics and the linear plant dynamics as a gradient descent algorithm with respect to a general Riemannian metric. At each step, the Riemannian metric is chosen so that the modelled plant dynamics do in fact match the true plant dynamics. Once the Riemannian metric is fully specified, the adaptation dynamics are uniquely defined. In this way, the adaptation dynamics induced in the adjustable parameters incorporate the knowledge of the true plant dynamics.

An advantage of the proposed design procedure is that the adaptation gain no longer plays a role in the adaptive mechanism. This is significant because in classical MRAC schemes, the non-linearities induced from the coupling of parameter adaptation and plant dynamics is negligible only when the adaptation gain is chosen smaller than would be desired. As a consequence, classical adaptation schemes generally result in very slow adaptation of the closed-loop systems. Conversely, the proposed scheme achieves fast convergence of the adaptive parameters by subsuming the adaptation gain into the Riemannian metric and incorporating knowledge of the plant dynamics in the adaptation rule.

Manuscript received: Jan. 20, 2000., Accepted: Mar. 10, 2000.
Sang-Heon Lee: Research Centre for Advanced manufacturing research, University of South Australia, Mawson Lakes, SA, 5066 Australia (email: sang-heon.lee@unisa.edu.au)
Robert Mahony: Dept. Systems Engineering, Research school of Information Science and Engineering, Australian National University, Canberra ACT 0200, Australia. (email:Robert.Mahony@syseng.anu.edu.au)
Il-Soo Kim: Dept. of Mechanical. Engineering, Mokpo National University, Muan-Gun, ChonNam Korea (email: ilsookim@chungkye.mokpo.ac.kr)

The design procedure is demonstrated by a case study of a simple single-pole, strictly proper, discrete-time plant. Since our aim in this paper is to study a new adaptation law in a simple situation, we make a number of strong assumptions on plant structure. We assume that the system is a deterministic model with no noise. The reference signal is taken to be a step function and the plant and the reference model are both strictly stable. In addition we assume that the high frequency gains of plant and reference model have the same sign.

This paper consists of seven sections including the introduction. Section 2 describes the explicit formulation of the MRAC scheme considered. In Section 3, a classical Euclidean gradient adaptation scheme using the MIT rule is reviewed. Section 4 shows how to form a non-linear adaptive system from the combination of parameter adaptation dynamics and the plant dynamics in the form of a non-Euclidean gradient descent algorithm. In Section 5, the problem of finding a positive definite matrix which defines the Riemannian metric required for the adaptive law is presented as a semi-definite programming problem. In Section 6, the performance of the proposed MRAC scheme is compared to a classical MRAC scheme. Section 7 reviews the contribution of the paper and outlines the advantages and limitations of the proposed scheme.

II. Problem formulation

In this section, a MRAC system for a simple linear plant is presented. Controller design is done on the assumption that the plant to be controlled has a single pole and no zero, and is a linear, stable, discrete-time system. The classical approach to MRAC in the discrete-time domain is shown in Fig. 1. The performance specifications are given in terms of a reference model, $\bar{G}$, along with the reference input signal, $r(k)$. Based on an estimate of the plant parameters, the certainty equivalence principle is used to design a feedback controller, C. The parameters are updated at each time instance k, according to the mismatch error, $e(k)$, between the actual closed-loop system output, $y(k)$, and the reference model output, $\bar{y}(k)$

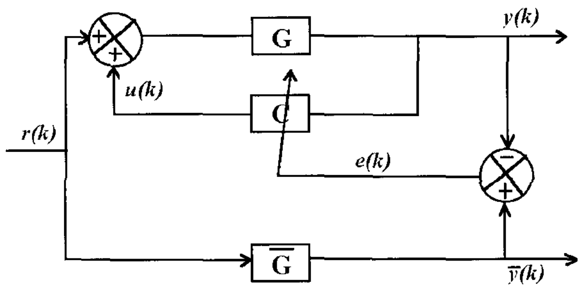


Fig. 1. System block diagram.

In this paper, the system is assumed to be deterministic with no noise. The reference signal, $r(k)$, is taken to be a step function. As a result, the adaptation algorithm is not persistently excited and the consequence of this choice is discussed in later analysis. In addition, we assume that the plant and the reference model are both strictly stable first order plants with relative degree one. It is assumed that the sign of the high frequency gain of the plant is known and is the same as that of the reference model.

The discrete-time plant, G, to be controlled and the reference model, $\bar{G}$ representing the control objective are given by

$$G(z)=\frac{bz^{-1}}{1+az^{-1}}, \qquad \bar{G}(z)=\frac{\bar{b}z^{-1}}{1+\bar{a}z^{-1}}, \tag{1}$$

where $a, b, \bar{a}, \bar{b} \in \Re$ are the unknown plant parameters and the reference model parameters respectively.

Equivalently, the difference equations for the closed-loop system output and the reference model output are

$$y(k)=b[r(k-1)+u(k-1)]-ay(k-1), \tag{2}$$

and

$$\bar{y}(k)=\bar{b}r(k-1)-\bar{a}\bar{y}(k-1). \tag{3}$$

The controller design is based on a simple pole/zero placement technique using the certainty equivalence principle; the closed-loop transfer function of the true plant is

$$Y(z)=\frac{G(z)}{[I-G(z)C(z)]}R(z), \tag{4}$$

where $C(z)$ is the Z-transform of the controller $c(k)$. Using the certainty equivalence to replace the plant $G(z)$ in Eq.4 with an estimate, $\hat{G}(z)$ and equating with the reference model behaviour yields

$$\bar{G}(z)=\frac{\hat{G}(z)}{|I-\hat{G}(z)C(z)|}. \tag{5}$$

Substituting Eq. 1 into Eq. 5 and solving for $C(z)$ yields

$$C(z)=\frac{(\bar{b}-\hat{b})+(\hat{a}\bar{b}-\bar{a}\hat{b})z^{-1}}{\bar{b}\hat{b}z^{-1}}, \tag{6}$$

where, $\hat{a}, \hat{b} \in \Re$ are the estimates of the unknown plant parameters.

A consequence of the simple design method used in here is that the controller $C(z)$ is non-causal. Since such a control strategy is impossible to apply in practice, it is necessary to modify Eq. 6 to yield a causal controller. The option taken is to include a stable, low-pass filter of relative order 1 in the feedback loop

$$M(z)=\frac{z^{-1}(1-\gamma)}{(1-\gamma z^{-1})}, \tag{7}$$

where $0 \leq \gamma < 1$. Thus, the difference equation of the final control action is

$$u(k-1)=\gamma u(k-2)+\frac{(1-\gamma)}{\bar{b}\hat{b}_{k-1}}\left[\left(\bar{b}-\hat{b}_{k-1}\right)y(k-1)+\left(\hat{a}_{k-1}\bar{b}-\bar{a}\hat{b}_{k-1}\right)y(k-2)\right]. \tag{8}$$

Remark 2.1 The low-pass filter, Eq. 7, can be interpreted in two ways. Firstly, since the filter has relative order 1, the overall relative order of the controller is zero and the control law can be implemented as a causal operator. Secondly, and perhaps more importantly, the designed controller *C(z)* of Eq. 6 is a PD (Proportional Derivative) controller. This is evident by rewriting Eq. 6 in the form

$$C(z) = \frac{1}{\bar{b}\hat{b}}(\bar{b}-\hat{b})(z-1) + \frac{(\bar{b}-\hat{b})+(\hat{a}\bar{b}-\bar{a}\hat{b})}{\bar{b}\hat{b}}.$$

Such controllers are highly susceptible to high frequency noise due to the derivative operation. In practice, the derivative operator is usually combined with a low-pass filter to ensure good behaviour. This is exactly the form of the new control action *M(z)C(z)*. Even though the original control action generated by *C(z)* is modified by the low-pass filter *M(z)*, the control action from *M(z)C(z)* still results in the same steady-state behaviour as *C(z)*.

III. Classical Euclidean gradient adaptation scheme

In this section, a brief review of the MIT rule of the adaptive control is given in the context of the model considered. We refer to this adaptation rule as a *Euclidean gradient adaptation scheme*.

The key principle of MRAC design is to use an error, the output mismatch error in this paper,

$$e(k) := \bar{y}(k) - y(k), \tag{9}$$

to measure the performance of the adaptive algorithm. Consider taking a cost function,

$$\Phi(\hat{\theta}_{k-1}) := \frac{1}{2}e(k)^2, \tag{10}$$

where $\hat{\theta}_{k-1} = (\hat{a}_{k-1}, \hat{b}_{k-1}) \in \Re^2$ is the vector of parameter estimates at time *k-1*. This cost function is used in the MIT rule of the original MRAC scheme [1], [6, Chapter 5]

The mismatch error *e(k)* is obtained from Eq.'s 2 and 3. Explicitly, writing *e(k)* to display its dependence on the parameters, $\hat{a}_{k-1}, \hat{b}_{k-1}$, yields

$$e(k) = \bar{b}r(k-1) - \bar{a}\bar{y}(k-1) - b$$
$$\left[r(k-1) + \gamma u(k-2) + \frac{(1-\gamma)}{\bar{b}\hat{b}_{k-1}}\left[\begin{matrix}(\bar{b}-\hat{b}_{k-1})y(k-1) \\ +(\hat{a}_{k-1}\bar{b}-\bar{a}\hat{b}_{k-1})y(k-2)\end{matrix}\right]\right] + ay(k-1). \tag{11}$$

Note that *u(k-2)*, *y(k-1)* and *y(k-2)* are independent from the *(k-1)*th parameters, $\hat{a}_{k-1}, \hat{b}_{k-1}$.

The partial derivatives of *e(k)* with respect to $\hat{a}_{k-1}$ and $\hat{b}_{k-1}$ are

$$\frac{\partial e(k)}{\partial \hat{a}_{k-1}} = -\frac{b}{\hat{b}_{k-1}}(1-\gamma)y(k-2),$$
$$\frac{\partial e(k)}{\partial \hat{b}_{k-1}} = \frac{b}{\hat{b}_{k-1}}\frac{(1-\gamma)}{\hat{b}_{k-1}}[y(k-1) + \hat{a}_{k-1}y(k-2)].$$

The adaptation mechanism for the parameter estimates vector is given by the discrete-time gradient descent algorithm;

$$\hat{\theta}_k = \hat{\theta}_{k-1} - s_k\frac{\partial \Phi(\hat{\theta}_{k-1})}{\partial \hat{\theta}_{k-1}} = \hat{\theta}_{k-1} - s_k e(k)\frac{\partial e(k)}{\partial \hat{\theta}_{k-1}}, \tag{12}$$

where

$$\frac{\partial e(k)}{\partial \hat{\theta}_{k-1}} = \begin{pmatrix}\frac{\partial e(k)}{\partial \hat{a}_{k-1}} \\ \frac{\partial e(k)}{\partial \hat{b}_{k-1}}\end{pmatrix} = \begin{pmatrix} -(1-\gamma)y(k-2) \\ \frac{1}{\hat{b}_{k-1}}(1-\gamma)[y(k-1)+\hat{a}_{k-1}y(k-2)]\end{pmatrix}\frac{b}{\hat{b}_{k-1}}, \tag{13}$$

and s_k is the adaptation gain. This is just a discrete-time version of the MIT rule [6, Chapter 5].

Remark 3.1 Since the true plant parameters a and b are unknown, some approximations are required to compute the gradient in practice. In the adaptive scheme, the model $\hat{a}_{k-1}$ and $\hat{b}_{k-1}$ are used to replace the unknown plant parameters a and b respectively. Also, to have the correct sign of the adaptation gain s_k, the sign of the high frequency gain of plant b is used and the term "sign $(\frac{b}{\hat{b}_{k-1}})$" is substituted $\frac{b}{\hat{b}_{k-1}}$.

Since an under exciting input reference signal is considered and the goal in MRAC systems is simply to force the error, $e(k) = \bar{y}(k) - y(k)$ to converge to zero, it is not necessary that the adjustable parameter values, $\hat{a}, \hat{b}$, should converge to the true plant parameters a, b, respectively. Rather, it is expected there is a whole set of the possible parameters $(\hat{a}, \hat{b})$ for which the error, $e(k) \equiv 0$. By assuming asymptotic convergence of the adaptive parameters to this set and then analysing the closed-loop difference equations, one finds the explicit equation for this set to be

$$(a+1)\hat{b}_{k-1} - \hat{a}_{k-1}b - b = 0. \tag{14}$$

The line defined by Eq.14 is termed as *I/O behaviour line*. It should be mentioned that since *e(k)=0* in the steady-state with $(\hat{a}, \hat{b})$ parameters on the I/O behaviour line, the adaptation dynamics are also zero. Thus, for a single step change in the reference input, the system should settle back into steady-state behaviour with constant controller coefficient after a short transient period.

Intuitively, it is expected that the adaptation gain s_k in Eq 12 has a significant effect on the parameter convergence rate, i.e. parameters will converge to I/O behaviour line slowly for small s_k and quickly for large s_k. In practice, however, for large s_k the adaptation behaviour becomes unpredictable and for small s_k convergence becomes sluggish. This observation is not conclusive evidence of the inadequacy of the scheme under normal operating conditions, but does tend to reduce confidence in the method.

Practically, MRAC schemes are used when there is a time-scale separation between the plant dynamics and the adaptive dynamics. Thus, the non-linearities introduced by the coupling of adaptation and plant dynamics are negligible, and the gradient, Eq. 13, maintains the properties expected in gradient descent algorithms. This requirement tends to force a choice of adaptation gain, s_k, smaller than would be desired and results in closed-loop systems with very slow transient behaviour. To overcome this difficulty, we have considered to design the adaptation dynamics to incorporate knowledge of the true plant dynamics.

IV. Non-Euclidean gradient adaptation scheme

In this section, a non-linear approach to the design of the adaptation algorithm in MRAC scheme is presented. In the proposed approach, the state of the plant is combined with the parameter adaptation dynamics to form a state for the full non-linear system. Taking the gradient of the cost function Φ, Eq. 10, with respect to a Riemannian metric, a gradient descent algorithm on the full non-linear state is induced. At each step, the Riemannian metric is chosen so that the modeled plant dynamics do in fact match the true plant dynamics. Once the Riemannian metric is fully specified, the adaptation dynamics are uniquely defined. In this way, the adaptation dynamics induced in the adjustable parameters incorporate the knowledge of the true plant dynamics. General background on Riemannian geometry can be found in [16, Appendix C.10]

The full state of the system given in Fig. 1 is defined to be

$$\xi_k = \begin{pmatrix} y(k) \\ \hat{a}_k \\ \hat{b}_k \end{pmatrix}. \tag{15}$$

Note that the state of the adjustable parameter estimates vector $\hat{\theta}_k = (\hat{a}_k, \hat{b}_k)$ in the classical MRAC scheme is $\Re^2$ while the state of parameter vector ξ_k in the proposed scheme is $\Re^3$. The *y(k)* state added in the proposed scheme is just the state of the linear system dynamics.

The cost function Φ is the same as used in Section 3, Eq. 10. However, now it is considered as a cost on the full state space $\Phi : \Re^3 \Rightarrow \Re$

$$\Phi(\xi_{k-1}) := \frac{1}{2} e(k)^2, \tag{16}$$

where *e(k)* is given by Eq. 9 and ξ_{k-1} is the full state of the system at time instance *k-1*.

To define the non-Euclidean gradient vector of $\Phi(\xi_{k-1})$, a Riemannian metric is introduced in $\Re^3$. A Riemannian metric is a bilinear, positive definite map for each $\xi_{k-1} \in \Re^3$.

$$\langle\langle *, * \rangle\rangle_{\xi_{k-1}} : \Re^3 x \Re^3 \Rightarrow \Re ,$$

which varies smoothly with ξ_{k-1}. For $\eta, \nu \in \Re^3$, tangent vectors of $\Re^3$ at ξ_{k-1} are then

$$\langle\langle \eta, \nu \rangle\rangle_{\xi_{k-1}} := \eta^T Q_{\xi_{k-1}}^{-1} \nu, \; Q_{\xi_{k-1}}^{-1} \in \Re^{3\times 3}, \tag{17}$$

where $Q_{\xi_{k-1}} = Q_{\xi_{k-1}}^T > 0$ (positive definite) and $Q_{\xi_{k-1}}^{-1}$ denotes the inverse of $Q_{\xi_{k-1}}$.

Let $D\Phi(\xi_{k-1})$ denote the vector of partial differentials of $\Phi(\xi_{k-1})$ with respect to the full state, i.e.

$$D\Phi(\xi_{k-1}) = \begin{pmatrix} \frac{\partial \Phi(\xi_{k-1})}{\partial y(k-1)} & \frac{\partial \Phi(\xi_{k-1})}{\partial \hat{a}_{k-1}} & \frac{\partial \Phi(\xi_{k-1})}{\partial \hat{b}_{k-1}} \end{pmatrix}^T$$

The gradient of $\Phi(\xi_{k-1})$, denoted grad $\Phi(\xi_{k-1})$, with respect to the choice of a Riemannian metric is uniquely generated as the solution of [17, Page 83]

$$\langle\langle X, \mathrm{grad}\Phi(\xi_{k-1}) \rangle\rangle_{\xi_{k-1}} := X^T D\Phi(\xi_{k-1}). \tag{18}$$

Solving Eq. 18 yields

$$\mathrm{grad}\Phi(\xi_{k-1}) = Q_{\xi_{k-1}} D\Phi(\xi_{k-1}). \tag{19}$$

Note that $Q_{\xi_{k-1}}$ is dependent on the states ξ_{k-1}.

Remark 4.1: As long as $Q_{\xi_{k-1}}$ is positive definite, then instantaneously, the cost $\Phi(\xi_{k-1})$ is decreasing. This becomes clearer if a continuous-time adaptation law is studied. Consider the continuous-time gradient descent adaptation law

$$\xi'(t) = -\mathrm{grad}\Phi(\xi(t)),$$

then, the directional derivative of the cost Φ in the direction of flow of $\xi(t)$ is

$$\Phi' = \frac{d}{dt}\Phi(\xi(t)) = -D\Phi(\xi(t))^T \mathrm{grad}\Phi = -\langle \mathrm{grad}\Phi, \mathrm{grad}\Phi \rangle = -\|\mathrm{grad}\Phi\|^2 < 0 . \tag{20}$$

Thus, the control parameters, $\hat{a}(t), \hat{b}(t)$, must evolve such that the cost $\Phi(\xi(t))$ is decreased. Of course, it is necessary to ensure that *y(t)* evolves to match the true dynamics of the plant by utilising the freedom of choice in $Q_{\xi(t)}$. This is a difficult problem in itself and in this paper we consider discrete-time plants and present a method of determining a suitable positive definite matrix $Q_{\xi_{k-1}}$.

The explicit equations for the entries of $D\Phi(\xi_{k-1})$ are computed as partial derivatives of the difference equation for *e(k)*, Eq. 11, with respect to the *(k-1)*'th parameters *y(k-1)*, $\hat{a}_{k-1}, \hat{b}_{k-1}$, respectively. The second and the third entries of $D\Phi(\xi_{k-1})$ are obtained using Eq. 13. For the additional state, the partial differentiation of the cost function $\Phi(\xi_{k-1})$ with respect to *y(k-1)*, $\frac{\partial \Phi(\xi_{k-1})}{\partial y(k-1)}$ is required. One has,

$$\frac{\partial \Phi(\xi_{k-1})}{\partial y(k-1)} = e(k) \frac{\partial e(k)}{\partial y(k-1)}, \tag{21}$$

where

$$\frac{\partial e(k)}{\partial y(k-1)} = \left[\frac{a\hat{b}_{k-1}}{b} - (1 - \frac{\hat{b}_{k-1}}{b})(1-\gamma) \right] \frac{b}{\hat{b}_{k-1}}.$$

As described in Section 2, the model estimates $\hat{a}_{k-1}$ and $\hat{b}_{k-1}$ can be substituted for the unknown plant parameters *a* and *b* respectively. Also, the sign of the high frequency gain of plant *b* is assumed to be known, yielding

$$\frac{\partial \Phi(\xi_{k-1})}{\partial y(k-1)} = e(k) \left[sign(\frac{\hat{b}_{k-1}}{b})\hat{a}_{k-1} - (1 - \frac{\hat{b}_{k-1}}{b})(1-\gamma) \right] sign(\frac{b}{\hat{b}_{k-1}}).$$

The gradient descent algorithm induced by Eq. 19 is simply

$$\xi_k = \xi_{k-1} - s_k \mathrm{grad}\Phi(\xi_{k-1}) = \xi_{k-1} - s_k Q_{\xi_{k-1}} D\Phi(\xi_{k-1}), \tag{22}$$

where s_k is the adaptation gain.

Note that since the only constraint on $Q_{\xi_{k-1}}$ at this stage is that $Q_{\xi_{k-1}}$ is positive definite, then, without loss of generality, one can write

$$s_k Q_{\xi_{k-1}} \equiv Q_{\xi_{k-1}}.$$

The gradient descent algorithm becomes

$$\xi_k = \xi_{k-1} - Q_{\xi_{k-1}} D\Phi(\xi_{k-1}), \tag{23}$$

with unit step size.

Remark 4.2: One of the advantages of the proposed scheme is the fact that the adaptation gain size no longer plays a role in the scheme. This is significant because, as mentioned in Section 2, the adaptation gain S_k affects the convergence rate of the adjustable parameters and in turn, the stability of the system. Instead of guessing the size of S_k to keep the system stable, the adaptation speed is automatically considered in the calculation procedure of the positive definite matrix $Q_{\xi_{k-1}}$ by incorporating the true plant dynamics. This guarantees fast convergence.

Remark 4.3: A disadvantage of the proposed scheme (cf. Eq. 23) is that there is no a priori guarantee that the cost $\Phi(\xi_{k+1}) < \Phi(\xi_k)$. Recalling that for the continuous-time adaptation rule, Eq. 20, one has $\Phi' < 0$. For S_k sufficiently small in Eq. 22, an equivalent result should hold. However, the necessity of subsuming S_k into $Q_{\xi_{k-1}}$ leads to potential stability problems. Understanding this issue is an area of ongoing research.

Consider the dynamics induced in the state *y(k)* by Eq. 23. These dynamics can be written

$$y(k) = y(k-1) - E_1^T Q_{\xi_{k-1}} D\Phi(\xi_{k-1}), \tag{24}$$

where $E_1 \in \Re^3$ is the unit vector with a one in the first element and zeros in the other elements. Given that *y(k)* and *y(k-1)* are measured directly from the plant output, then Eq. 24 generates a linear constraint on $Q_{\xi_{k-1}}$

$$E_1^T Q_{\xi_{k-1}} D\Phi(\xi_{k-1}) = y(k-1) - y(k). \tag{25}$$

As long as $Q_{\xi_{k-1}}$ satisfies Eq. 25, the first entry of the induced gradient dynamics in Eq. 23 exactly replicates the true plant dynamics.

This leads to an optimisation problem that lies at the heart of the proposed scheme.

Problem 4.1: *At each time instance k, find a matrix* $Q_{\xi_{k-1}}$ *which depends smoothly on* ξ_{k-1}, *satisfying*

1. $Q_{\xi_{k-1}} > 0$ *and* $Q_{\xi_{k-1}} = Q^T_{\xi_{k-1}}$ *(Positive definite).*
2. $E_1^T Q_{\xi_{k-1}} D\Phi(\xi_{k-1}) = y(k-1) - y(k)$ *(Linear Constraint)*

and such that the closed-loop system shows desirable behaviour.

Remark 4.4: Requirements 1 and 2 of Problem 4.1 are the practical requirements that ensure the gradient descent algorithm replicates the true plant dynamics and displays gradient characteristics. These constraints, however, leave a great deal of leeway in choosing $Q_{\xi_{k-1}}$ to ensure the closed-loop system shows desirable behaviour.

V. Determining an optimal positive definite matrix

In this section, a specific approach to solve the optimisation problem, Problem 4.1, is presented. The approach relies on choosing $Q_{\xi_{k-1}}$ to minimise a one-step-ahead estimate of the cost Φ of Eq. 16.

A natural approach to finding $Q_{\xi_{k-1}}$ is to generate a one-step-ahead estimate of the output, $\hat{y}^e(k+1)$, based on a particular $Q_{\xi_{k-1}}$ and then minimise the cost $\left\|\hat{y}^e(k+1) - \bar{y}(k+1)\right\|^2$ subject to the requirements of Problem 4.1. Ensuring that $Q_{\xi_{k-1}}$ satisfies Eq. 25 should not be difficult as this is simply a linear constraint on symmetric matrix space. Dealing with the positive definite constraint forces one into the realm of semi-definite programming. Following the lead of recent development of semi-definite programming [18], we introduce the cost function

$$\Psi^e_{\xi_{k-1}}(Q^e_{\xi_{k-1}}) = \left\|\hat{y}^e(k+1) - \bar{y}(k+1)\right\|^2 - \varepsilon \ln(\det(Q^e_{\xi_{k-1}})), \tag{26}$$

where $\hat{y}^e(k+1)$ is one-step-ahead estimation of output *y(k+1)* and $Q^e_{\xi_{k-1}}$ is the estimate of $Q_{\xi_{k-1}}$. Here, det($Q^e_{\xi_{k-1}}$) is the determinant of the matrix $Q^e_{\xi_{k-1}}$ and $\bar{y}(k+1)$ is the output of the reference model which can be easily obtained from Eq. 3.

Remark 5.1: The first term in Eq. 26 is the desired quadratic cost term while the second term is a self-concordant barrier function [18][19] added to ensure the minimum of $\Psi^e_{\xi_{k-1}}(Q^e_{\xi_{k-1}})$ always lies in the set of positive definite matrices. By choosing ε sufficiently small, the influence of the barrier function on the quadratic cost function is negligible except in the neighbourhood of the boundary of positive definite matrices.

When $Q_{\xi_{k-1}}$ needs to be determined, the current and the previous value of output *y(k),y(k-1)*, reference model output $\bar{y}(k), \bar{y}(k-1)$, input *r(k),r(k-1)*, and error *e(k)* as well as the previous values of control action *u(k-1)* and the adjustable parameter vector $\hat{a}_{k-1}$ and $\hat{b}_{k-1}$ are all known. In addition, the derivative of the cost, $D\Phi(\xi_{k-1})$, is also available.

The one-step-ahead output estimation $\hat{y}^e(k+1)$ is obtained using the one-step-ahead estimation of control action to the system, $\hat{u}^e_k(k)$,

$$\hat{y}^e(k+1) = \hat{b}^e_k(\hat{u}^e(k) + r(k)) - \hat{a}^e_k y(k). \tag{27}$$

The control action $\hat{u}^e_k(k)$ is generated from the difference equation for the control, Eq. 8,

$$\hat{u}^e(k) = \gamma u(k-1) + \frac{(1-\gamma)}{\bar{b}\hat{b}^e_k}\left[(\bar{b} - \hat{b}^e_k)y(k) + (\hat{a}^e_k\bar{b} - \bar{a}\hat{b}^e_k)y(k-1)\right], \tag{28}$$

where $\hat{a}^e_k$ and $\hat{b}^e_k$ are the one-step-ahead estimates of model parameters based on the gradient descent algorithm generated from the Riemannian metric given by the matrix $Q^e_{\xi_{k-1}} > 0$. Equation 23 yields

$$\hat{a}^e_k = \hat{a}_{k-1} - E_2^T Q^e_{\xi_{k-1}} D\Phi(\xi_{k-1}), \tag{29}$$

and

$$\hat{b}^e_k = \hat{b}_{k-1} - E_3^T Q^e_{\xi_{k-1}} D\Phi(\xi_{k-1}), \tag{30}$$

where E_2 and E_3 are the unit vectors with a one in the second and third elements respectively.

To solve Problem 4.1, we proceed by deriving a gradient descent algorithm on the set of positive definite matrices satisfying the constraints of the Problem 4.1. To find the gradient of $\Psi^{\varepsilon}_{\xi_{k-1}}(\mathcal{Q}^{\varepsilon}_{\xi_{k-1}})$, consider the Euclidean metric on the positive definite matrices,

$$\langle u,v\rangle = \mathrm{tr}(u^T v),$$

where $u = u^T, v = v^T, u,v \in \Re^{n\times n}$. This is just the metric on the positive definite matrices inherit as an open set of symmetric matrix space.

Using this metric along with the definition of gradient, Eq. 18, yields

$$\left\langle\left\langle X, \mathrm{grad}\Psi^{\varepsilon}_{\xi_{k-1}}(\mathcal{Q}^{\varepsilon}_{\xi_{k-1}})\right\rangle\right\rangle = \mathrm{tr}(X^T \mathrm{grad}\Psi^{\varepsilon}_{\xi_{k-1}}(\mathcal{Q}^{\varepsilon}_{\xi_{k-1}})) = \mathrm{D}\Psi^{\varepsilon}_{\xi_{k-1}}|_{\mathcal{Q}^{\varepsilon}_{\xi_{k-1}}}[X], \quad (31)$$

where $\mathrm{D}\Psi^{\varepsilon}_{\xi_{k-1}}|_{\mathcal{Q}^{\varepsilon}_{\xi_{k-1}}}[X]$ is the directional derivatives of $\Psi^{\varepsilon}_{\xi_{k-1}}$ in direction X evaluated at the point $\mathcal{Q}^{\varepsilon}_{\xi_{k-1}}$. This is given by[1]

$$\mathrm{D}\Psi^{\varepsilon}_{\xi_{k-1}}|_{\mathcal{Q}^{\varepsilon}_{\xi_{k-1}}}[X] = 2(\hat{y}^e(k+1) - \overline{y}(k+1))$$
$$\mathrm{D}(\hat{y}^e(k+1) - \overline{y}(k+1))|_{\mathcal{Q}^{\varepsilon}_{\xi_{k-1}}}[X] - \varepsilon\mathrm{tr}(X^T(\mathcal{Q}^{\varepsilon}_{\xi_{k-1}})^{-1},$$

or as only $\hat{y}^e(k+1)$ contains $\mathcal{Q}^{\varepsilon}_{\xi_{k-1}}$

$$\mathrm{D}\Psi^{\varepsilon}_{\xi_{k-1}}|_{\mathcal{Q}^{\varepsilon}_{\xi_{k-1}}}[X] = 2\left[\hat{y}^e(k+1) - \overline{y}(k+1)\right]\mathrm{D}\hat{y}^e(k+1)|_{\mathcal{Q}^{\varepsilon}_{\xi_{k-1}}}[X] - \varepsilon\mathrm{tr}(X^T(\mathcal{Q}^{\varepsilon}_{\xi_{k-1}})^{-1}. \quad (32)$$

Rewriting $\hat{y}^e(k+1)$ in terms of $\hat{a}^e_k$ and $\hat{b}^e_k$, one has an explicit form for $D\hat{y}^e(k+1)|_{\mathcal{Q}^{\varepsilon}_{\xi_{k-1}}}[X]$ as follows

$$\mathrm{D}\hat{y}^e(k+1)|_{\mathcal{Q}^{\varepsilon}_{\xi_{k-1}}}[X] =$$
$$\mathrm{D}\left\{\hat{b}^e_k[\gamma u(k-1) + r(k)] + \frac{(1-\gamma)}{\overline{b}}\left[\begin{matrix}(\overline{b} - \hat{b}^e_k)y(k) \\ +(\hat{a}^e_k\overline{b} - \overline{a}\hat{b}^e_k)y(k-1)\end{matrix}\right] - \hat{a}^e_k y(k)\right\}\Bigg|_{\mathcal{Q}^{\varepsilon}_{\xi_{k-1}}}.$$

As shown in Eq.'s 29 and 30, $\hat{a}^e_k$ and $\hat{b}^e_k$ are the only terms dependent on $\mathcal{Q}^{\varepsilon}_{\xi_{k-1}}$. Thus,

$$\mathrm{D}\hat{y}^e(k+1)|_{\mathcal{Q}^{\varepsilon}_{\xi_{k-1}}}[X] = -E_3^T X\mathrm{D}\Phi(\xi_{k-1})[\gamma u(k-1) + r(k)] + E_2^T X\mathrm{D}\Phi(\xi_{k-1})y(k)$$
$$+\frac{(1-\gamma)}{\overline{b}}\left\{E_3^T X\mathrm{D}\Phi(\xi_{k-1})y(k) + \left[-\overline{b}E_2^T X\mathrm{D}\Phi(\xi_{k-1}) + \overline{a}E_3^T X\mathrm{D}\Phi(\xi_{k-1})\right]y(k-1)\right\},$$

or,

$$\mathrm{D}\hat{y}^e(k+1)|_{\mathcal{Q}^{\varepsilon}_{\xi_{k-1}}}[X]$$
$$= \frac{1}{2}\mathrm{tr}\left\{X^T\left[\mathrm{D}\Phi(\xi_{k-1})E_3^T + E_3\mathrm{D}\Phi(\xi_{k-1})^T\right]\right\}\left\{-[\gamma u(k-1) + r(k)] + \frac{(1-\gamma)}{\overline{b}}[y(k) + \overline{a}y(k-1)]\right\}$$
$$+\frac{1}{2}\mathrm{tr}\left\{X^T\left[\mathrm{D}\Phi(\xi_{k-1})E_2^T + E_2\mathrm{D}\Phi(\xi_{k-1})^T\right]\right\}\{y(k) - (1-\gamma)y(k-1)\},$$

where $\mathrm{D}\Phi(\xi_{k-1})^T$ is the transpose of $\mathrm{D}\Phi(\xi_{k-1})$.

By substituting the above equation into Eq. 32 and comparing with Eq. 31 the gradient of the cost function is obtained explicitly,

$$\mathrm{grad}\Psi^{\varepsilon}_{\xi_{k-1}}(\mathcal{Q}_{\xi_{k-1}}) = \left[\hat{y}^e(k+1) - \overline{y}(k+1)\right]$$
$$\left\{\begin{matrix}\left[\mathrm{D}\Phi(\xi_{k-1})E_3^T + E_3\mathrm{D}\Phi(\xi_{k-1})^T\right]\left[-[\gamma u(k-1) + r(k)] + \frac{(1-\gamma)}{\overline{b}}[y(k) + \overline{a}y(k-1)]\right] \\ +\left[\mathrm{D}\Phi(\xi_{k-1})E_2^T + E_2\mathrm{D}\Phi(\xi_{k-1})^T\right][y(k) - (1-\gamma)y(k-1)]\end{matrix}\right\} - \varepsilon(\mathcal{Q}_{\xi_{k-1}})^{-1}. \quad (33)$$

The gradient, grad $\Psi^{\varepsilon}_{\xi_{k-1}}(\mathcal{Q}^{\varepsilon}_{\xi_{k-1}})$, gives the optimum descent *direction of the cost* $\Psi^{\varepsilon}_{\xi_{k-1}}(\mathcal{Q}^{\varepsilon}_{\xi_{k-1}})$ on the set of symmetric matrices. Due to the barrier function, a steepest descent algorithm based on a descent direction, grad $\Psi^{\varepsilon}_{\xi_{k-1}}(\mathcal{Q}^{\varepsilon}_{\xi_{k-1}})$, and initialised with a positive definite matrix will never evolve outside the set of positive definite matrices. It is now necessary to ensure that the gradient descent direction also satisfies the linear constraint, Eq.25. This is achieved by projecting grad $\Psi^{\varepsilon}_{\xi_{k-1}}(\mathcal{Q}^{\varepsilon}_{\xi_{k-1}})$ orthogonally onto the tangent space of the linear space,

$$\mathbf{L}(\xi_{k-1}) = \left\{\mathcal{Q}^{\varepsilon}_{\xi_{k-1}} \mid E_1^T\mathcal{Q}^{\varepsilon}_{\xi_{k-1}}\mathrm{D}\Phi(\xi_{k-1}) = y(k-1) - y(k)\right\},$$

to generate a constrained descent direction satisfying Eq. 25. The tangent space of the linear space $\mathbf{L}(\xi_{k-1})$ is

$$\mathbf{L}_0(\xi_{k-1}) = \mathbf{T}_{\mathcal{Q}^{\varepsilon}_{\xi_{k-1}}}\mathbf{L}(\xi_{k-1}) = \left\{X \mid E_1^T X\mathrm{D}\Phi(\xi_{k-1}) = 0\right\}. \quad (34)$$

Taking the projection of $\Psi^{\varepsilon}_{\xi_{k-1}}(\mathcal{Q}^{\varepsilon}_{\xi_{k-1}})$ onto $\mathbf{T}_{\mathcal{Q}^{\varepsilon}_{\xi_{k-1}}}\mathbf{L}(\xi_{k-1})$, one generates a constrained descent direction satisfying Eq. 25 The projection operator is denoted

$$\mathbf{P}_{\mathbf{L}_0(\xi_{k-1})} : \Re^{3\times 3} \rightarrow \mathbf{T}_{\mathcal{Q}^{\varepsilon}_{\xi_{k-1}}}\mathbf{L}(\xi_{k-1}),$$

and an explicit form for $\mathbf{P}_{\mathbf{L}_0(\xi_{k-1})}$ is computed in Appendix A.

The final result required for an effective optimisation procedure to solve Problem 4.1 is an initial condition satisfying the constraints. An explicit method for calculating such a matrix is given in Appendix B.

To conclude this section, we give the optimisation procedure used to generate $\mathcal{Q}_{\xi_{k-1}}$.

Algorithm 5.1: Optimisation algorithm to determine $\mathcal{Q}_{\xi_{k-1}}$.

0: Input the known output y(k),y(k-1), reference model output $\overline{y}(k), \overline{y}(k-1)$, input r(k), r(k-1), error e(k), control u(k-1), and parameter estimates $\hat{\theta}_{k-1} = (\hat{a}_{k-1}, \hat{b}_{k-1})$ as well as the derivative of the cost $\mathrm{D}\Phi(\xi_{k-1})$ to the algorithm.

1: Generate the initial positive definite matrix $\mathcal{Q}^{\varepsilon}_{\xi_{k-1}}(0)$ satisfying the linear constraint Eq. 25 (See Appendix B.)

2: Let $\varepsilon = 1$ in Eq. 26 and set $j = 0$.

3: Compute $\mathrm{grad}\Psi^{\varepsilon}_{\xi_{k-1}}$ using Eq. 33 and the projection $\mathbf{P}_{\mathbf{L}_0(\xi_{k-1})}(\mathrm{grad}\Psi^{\varepsilon}_{\xi_{k-1}})$. (The projection is taken to ensure the linear constraint (Eq. 25) is satisfied. See Appendix A.)

4: Compute $\alpha_j := \min_{\alpha>0}\Psi^{\varepsilon}_{\xi_{k-1}}\left\{\mathcal{Q}^{\varepsilon}_{\xi_{k-1}}(j) - \alpha\mathbf{P}_{\mathbf{L}_0(\xi_{k-1})}(\mathrm{grad}\Psi^{\varepsilon}_{\xi_{k-1}})\right\}$. (Use the MATLAB optimisation toolbox.)

5: Set $\mathcal{Q}^{\varepsilon}_{\xi_{k-1}}(j+1) = \mathcal{Q}^{\varepsilon}_{\xi_{k-1}}(j) - \alpha\mathbf{P}_{\mathbf{L}_0(\xi_{k-1})}(\mathrm{grad}\Psi^{\varepsilon}_{\xi_{k-1}})$.

6: Compute the projection $\mathcal{Q}^{\varepsilon}_{\xi_{k-1}}(j+1) = \mathbf{P}_{\mathbf{L}(\xi_{k-1})}\mathcal{Q}^{\varepsilon}_{\xi_{k-1}}(j+1)$, the orthogonal projection onto $\mathbf{L}(\xi_{k-1})$ (cf. Appendix A) to compensate numerical error.

7: If j is divisible by 5, let $\varepsilon = 10^{-1}\varepsilon$.

8: If $\varepsilon = 10^{-8}$ goto 9. Else j=j+1, goto 3.

9: $\mathcal{Q}_{\xi_{k-1}} = \mathcal{Q}^{\varepsilon}_{\xi_{k-1}}(j+1)$, this matrix always satisfies the requirements of Problem 4.1.

Remark 5.2 Note that the computational cost of calculating $\mathcal{Q}_{\xi_{k-1}}$ is significant. Simulations indicate that this calculation can be achieved in 2-3 seconds for the plant considered[2]. It is

[1] Here, we use the fact that $\ln(\det(\mathcal{Q}^{\varepsilon}_{\xi_{k-1}}))|_{\mathcal{Q}^{\varepsilon}_{\xi_{k-1}}}[X] = \mathrm{tr}(X^T(\mathcal{Q}^{\varepsilon}_{\xi_{k-1}})^{-1})$ [18, Page 70].

[2] Calculations were done using the MATLAB optimisation toolbox on a Sun UltraSPARC I machine, clock speed 167 MHz

expected that by improving the computational efficiency of the optimisation algorithm, this computational cost can be significantly reduced. As a consequence, the authors expect that the proposed method should be applicable to most process control problems. In contrast, the computational complexity is likely to rule out applications in telecommunications.

VI. Simulation results and discussion

In this section, the results of simulations of both the classical and proposed model reference adaptive control systems are presented. The simulation indicates several desirable features of the proposed algorithm, though there are still unanswered questions in the application of this method, such as developing a full understanding of its stability properties. Simulations of numerous examples have shown, however, that the proposed scheme displays good convergence properties.

For this simulation, the plant and reference model transfer functions were chosen to be

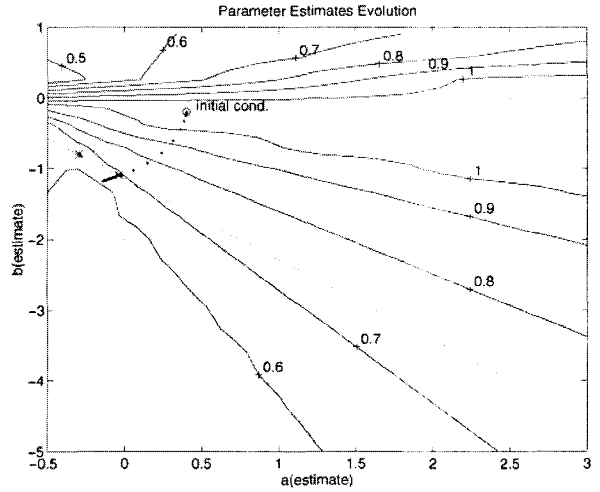
$$G(z)=\frac{-0.8z^{-1}}{1-0.3z^{-1}},\bar{G}(z)=\frac{-0.8z^{-1}}{1+0.6z^{-1}} \text{ and } \hat{G}(z)=\frac{-0.2z^{-1}}{1+0.4z^{-1}}$$

As mentioned before, the choice of adaptation gain for the classical MRAC system is important. The trade-off between the system stability and the parameter convergence speed was examined for various gain values. After several trials, a fixed adaptation gain $s_k = 0.2$ for all time instance k has been chosen for this example. In both schemes, the cost function $\Phi = \frac{1}{2}e(k)^2$ is used. The value of γ in the low-pass filter, Eq. 7, was chosen to be 0.9.

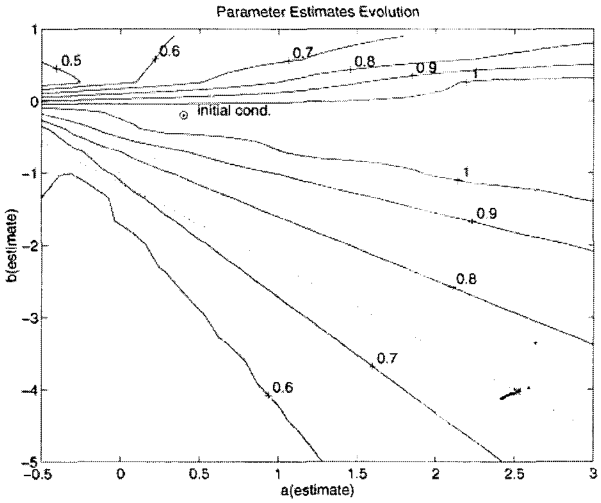
A step input was used as the reference command input. Because of this, the parameter estimates need not approach their true plant values, but should converge to the I/O behaviour line (cf. Eq. 14).

The simulation results are presented in Figs 2 to 4. In the Figs of parameter estimates evolution, Fig. 2, the dotted line is the I/O behaviour line and the contour lines are closed-loop system stability measure lines. These contour lines enclose regions in parameter space where the largest absolute value of a pole of the closed-loop system is less than or equal to a marked contour value.

In both the classical and proposed schemes, the convergence to the I/O behaviour line seems acceptable. However, the relative difference in performance is clearly shown in Fig. 3. Here, the log plot of error versus time is given and the extremely rapid asymptotic convergence of the proposed scheme is displayed. In the both schemes, the adaptation scheme is only initiated at time k=4 to avoid initialisation difficulties. The initial condition is chosen in an unstable region of parameter space and consequently, a short transient is observed in both schemes. Observe in Fig. 3(b), that immediately the parameter adjustment is initiated, the error is stabilised and quickly decreased. Conversely, in the classical MRAC scheme the small adaptation gain restricts the rate of adaptation, leading to a larger transient before convergence. Moreover, the adaptation gain also limits the asymptotic rate of the convergence. Note that increasing the magnitude of the adaptation gain for the classical scheme leads to stability problems.

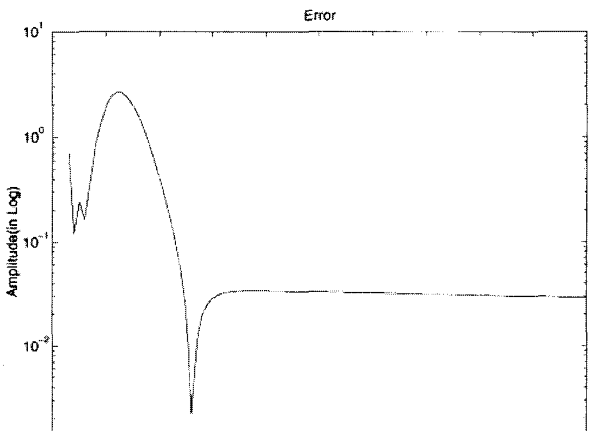


(a) Classical MRAC

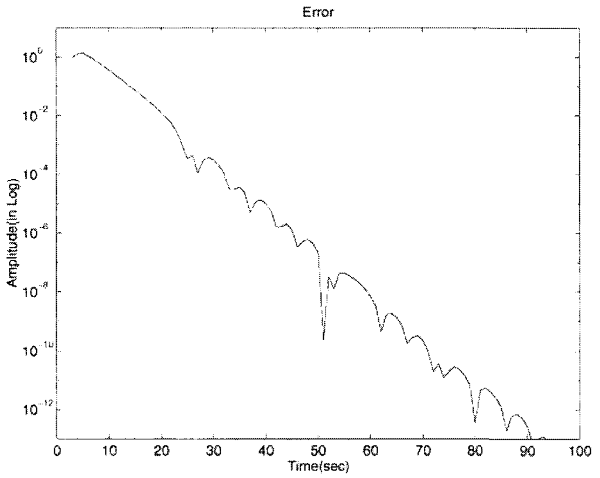


(b) Proposed MRAC

Fig. 2. Parameter estimates evolution.

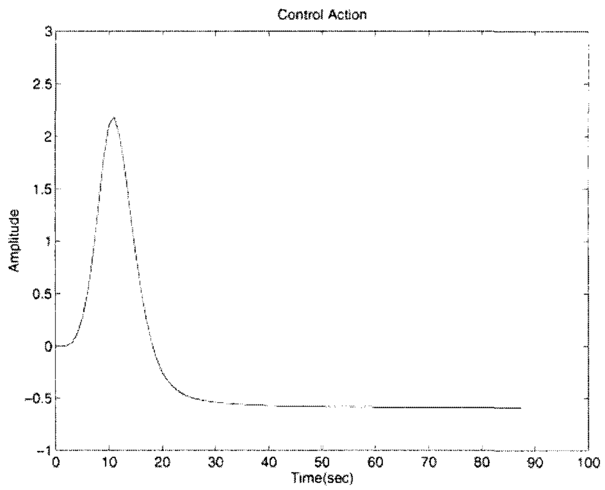


(a) Classical MRAC

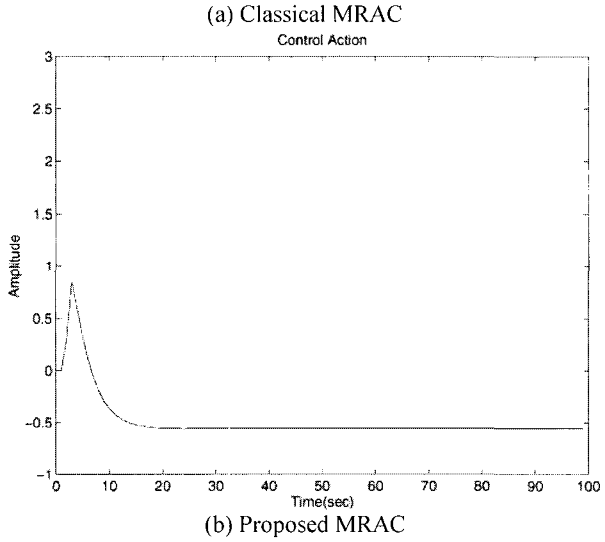


(b) Proposed MRAC

Fig. 3. Error plot.



(a) Classical MRAC



(b) Proposed MRAC

Fig. 4. Control action plot.

Remark 6.1 In fact, for the first two or three steps of the proposed parameter adjustment dynamics, the Riemannian metric Q_{ξ_k} chosen is nearly singular and has a large condition number. This leads to the large jumps in parameter values observed in Fig. 2(b). In most examples, such jumps are advantages since they quickly bring the parameter values into a stable region in which the asymptotic properties of the control algorithm are displayed. However, occasionally, the ill-conditioning in Q_{ξ_k} leads to poor adaptation and repeated ill-conditioning of $Q_{\xi_{k+1}}, Q_{\xi_{k+2}}$ ...etc. In these situations, the proposed scheme may display stability problems. This issue is an area of ongoing research.

Fig. 4 is the plot of control action versus time. It is clear that the proposed scheme offers significantly faster convergence to the I/O behaviour line without an increase in the control effort.

VII. Conclusion

In this paper, we have developed a new non-linear approach to the design of adaptive control schemes based around the use of non-Euclidean gradient descent algorithms. The proposed scheme has the advantages that the adaptation gain term no longer plays a dominant role in the system stability since it is subsumed into the Riemannian metric calculation. This results in significantly faster asymptotic convergence of parameters and more flexibility in the transient response of the closed-loop system. The key contributions of the paper are; the formulation of the gradient descent algorithm in such a way as to incorporate the true plant dynamics and the development of a criterion and method to determine suitable positive definite matrices for the adaptation mechanism. Simulations have shown that the parameter convergence of the proposed scheme is much faster than the classical MRAC scheme. Further work is required to investigate stability issues as well as the generalisation to continuous-time plants and more general system models.

Acknowledgement

The authors would like to express their gratitude to John Moore and Iven Mareels for stimulating discussion and advice

Appendix A: Projection operator

Let L be the affine subspace of symmetric matrices in $\Re^{n \times n}$ given by

$$L = \{M \mid \mathrm{tr}(A^T MB) = c\},$$

where A and B are any nonzero matrices, $M = M^T$ is a symmetric matrix and c is some scalar. The orthogonal projection of $\Re^{n \times n}$ onto L is denoted by $P_L : \Re^{n \times n} \Rightarrow L$. The Euclidean inner product $\langle X, Y \rangle$ on the set of symmetric matrices $X = X^T, Y = Y^T$ is given by $\langle X, Y \rangle = \mathbf{tr}(XY)$. To find the orthogonal projection associated with L, we convert $\mathrm{tr}(A^T MB) = c$ into the form

$$\mathrm{tr}(A^T MB) = \mathrm{tr}(\frac{1}{2}(AB^T + BA^T)M) = \left\langle \frac{1}{2}(AB^T + BA^T), M \right\rangle = c.$$

It is known that any matrix can be expressed as the sum of two matrices

$$\mathrm{M} = \mathrm{M}_{||} + \mathrm{M}_{\perp} \tag{36}$$

where

$$M_{\perp} = \mu_1 \frac{1}{2}(AB^T + BA^T), \mathrm{M}_{||} = [M - \mu_1 \frac{1}{2}(AB^T + BA^T)],$$

where $\mathbf{M}_{||}$ lies in the linear subspace L_0 defined by

$L_0 = \{M \mid \mathrm{tr}(A^T MB) = 0\}$ and $M_\perp$ isperpendicular to L_0. From the properties of inner product, one has

$$\langle M_{\|}, M_\perp \rangle = \mathrm{tr}([\frac{1}{2}(AB^T + BA^T)]^T [M - \mu_1 \frac{1}{2}(AB^T + BA^T)]) = 0,$$

or

$$\mu_1 = \frac{4\mathrm{tr}(A^T MB)}{\mathrm{tr}((AB^T + BA^T)(AB^T + BA^T))}. \tag{37}$$

Thus, the projection of $P_{L_0} : \Re^{n \times n} \Rightarrow L_0$ is given by

$$P_{L_0} = M - \mu_1 \frac{1}{2}(AB^T + BA^T). \tag{38}$$

Then, the projection of $P_L : \Re^{n \times n} \Rightarrow L$ is given by

$$P_L(M) = P_{L_0}(M) + \mu_2 \frac{1}{2}(AB^T + BA^T), \tag{39}$$

where μ_2 is obtained from

$$\left\langle \frac{1}{2}(AB^T + BA^T), \mu_2 \frac{1}{2}(AB^T + BA^T) \right\rangle = c,$$

or,

$$\mu_2 = \frac{4c}{\mathrm{tr}((AB^T + BA^T)(AB^T + BA^T))}. \tag{40}$$

The general form of the projection operator $P_L : \Re^{n \times n} \Rightarrow L$ is then

$$P_L(M) = M + 2(AB^T + BA^T) \frac{c - \mathrm{tr}(A^T MB)}{\mathrm{tr}((AB^T + BA^T)(AB^T + BA^T))}. \tag{41}$$

Appendix B: Finding a feasible initial positive definite matrix

In this appendix, a systematic method for a calculating an initial positive definite matrix, $Q_{\xi_{k-1}}(0)$, satisfying the constraints in Problem 4.1 is presented. This matrix is then used as the input for the Algorithm 5.

A positive definite matrix can always be factored into the form

$$Q_{\xi_{k-1}}(0) = P_{\xi_{k-1}} P_{\xi_{k-1}}^T, \tag{42}$$

where $Q_{\xi_{k-1}}(0) = (Q_{\xi_{k-1}}(0))^T > 0$ is an initial positive matrix required for Algorithm 5.1 and $P_{\xi_{k-1}}$ is a square root of $Q_{\xi_{k-1}}(0)$.

Substituting Eq. 42 into the linear constraint Eq. 25, Section 4, one has

$$E_1^T P_{\xi_{k-1}} P_{\xi_{k-1}}^T D\Phi\xi_{k-1} = y(k-1) - y(k).$$

Assume that $D\Phi\xi_{k-1} \neq 0$ (that is $e(k) \neq 0$), and define

$$g(k) = \frac{D\Phi\xi_{k-1}}{\|D\Phi\xi_{k-1}\|}, \text{ and } t(k) = \frac{y(k-1) - y(k)}{\|D\Phi\xi_{k-1}\|}.$$

Then, one has

$$E_1^T P_{\xi_{k-1}} P_{\xi_{k-1}}^T g(k) = t(k), \tag{43}$$

where *g(k)* and E_1 are both the unit length vectors, and all the scaling information is contained in *t(k)*.

The development proceeds by thinking of $P_{\xi_{k-1}}^T$ \$ as a transformation on $\Re^{3\times 3}$. Using the vector inner product $\langle u,v \rangle = u^T v$, where $u,v \in \Re^n$, one has from Eq. 43

$$\langle P_{\xi_{k-1}}^T E_1, P_{\xi_{k-1}}^T g(k) \rangle = t(k). \tag{44}$$

Consider the 2-dimensional subspace in $\Re^3$ given by $\mathrm{sp}\{E_1, g(k)\}$. An orthonormal basis for this subspace is provided by the vectors

$$W_1 = \frac{E_1 + g(k)}{\|E_1 + g(k)\|} \text{ and } W_2 = \frac{E_1 - g(k)}{\|E_1 - g(k)\|} \in \Re^3 \tag{45}$$

Expressing E_1 and *g(k)* as 2-dimensional vectors in this subspace, written in terms of the coordinates induced by the orthonormal vectors, W_1 and W_2, one has

$$\tilde{E}_1 = \begin{pmatrix} W_1^T \\ W_2^T \end{pmatrix} E_1 \text{ and } \tilde{g}(k) = \begin{pmatrix} W_1^T \\ W_2^T \end{pmatrix} g(k) \in \Re^2. \tag{46}$$

Here $\|\tilde{E}_1\| = 1$ and $\|\tilde{g}(k)\| = 1$ since E_1 and *g(k)* lie in the span of W_1 and W_2.

Note that the two coordinate vectors W_1 and W_2 are chosen such that the vectors $\tilde{E}_1$ and $\tilde{g}(k)$ always lie symmetrically about the W_2 axis in the right half plane of the 2-D subspace, spanned by W_1 and W_2 (see Fig. 5). To preserve the intuition provided by this construction, it is necessary to be careful about the sign of *t(k)* in Eq. 44.

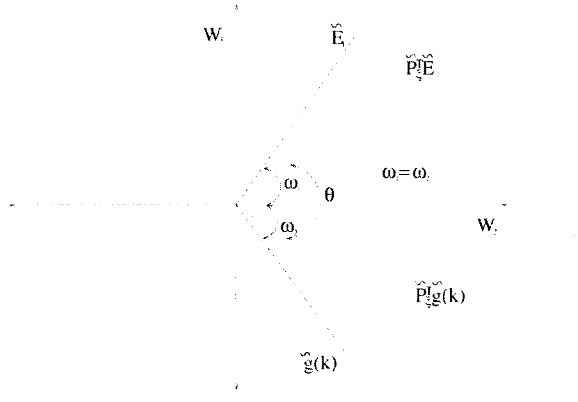


Fig. 5. Subspace Given by $\mathrm{sp}\{E_1, g(k)\}$

1. Case of $t(k) \geq 0$: In this case, the two vectors $\tilde{E}_1$ and $\tilde{g}(k)$ are in the right half plane. If the angle between two vectors $\tilde{E}_1$ and $\tilde{g}(k)$ is larger than a set value (in this case $\frac{\pi}{3}$) and less than π, a scaling matrix $\tilde{P}_{\xi_{k-1}}$ is chosen to premultiply $\tilde{E}_1$ and $\tilde{g}(k)$ which acts to reduce the angle. In the case that the angle is less than the set value, it is sufficient to choose $\tilde{P}_{\xi_{k-1}} = I_2 \in \Re^{2\times 2}$. This is the case that Eq. 43 can be satisfied by adjusting a scaling parameter δ_k. (cf. Eq. 57).

2. Case of $t(k) < 0$: In this case, the intuition of reducing the angle between $\tilde{E}_1$ and $\tilde{g}(k)$ is not valid. Take $E_1 = -E_1$ and $t(k) = -t(k)$. This leaves Eq. 44 valid, and returns the analysis to that discussed in case 1.

Remark B.1: The choice of $\tilde{P}_{\xi_{k,1}}$ to pre-multiply $\tilde{E}_1$ and $\tilde{g}(k)$ in case 1 can be thought of as tweaking the direction of gradient flow slightly by adjusting the metric to make sure that the sign of $\langle P^T_{\xi_{k,1}}E_1, P^T_{\xi_{k,1}}g(k)\rangle$ is the same as the sign of $t(k)$. Once this is the case, then ensuring that Eq. 44 is satisfied can be achieved by scaling $P_{\xi_{k,1}}$.

Remark B.2: To complete the construction indicated above, it is necessary that $\tilde{E}_1 \neq \pm\tilde{g}(k)$, or equivalently the angle between $\tilde{E}_1$ and $\tilde{g}(k)$ is neither 0 nor π. In practice, this seems never to occur. In the case that such a situation occur, the adaptation would be frozen for that time instance.

The matrix $\tilde{P}_{\xi_{k,1}} \in \Re^{2\times 2}$ applied case 1 above is simply

$$\tilde{P}_{\xi_{k,1}} = \begin{pmatrix} 1+\beta & 0 \\ 0 & 1 \end{pmatrix}, \text{ where } \beta > -1. \qquad (47)$$

To determine the unknown β, consider computing the angle between $\tilde{P}^T_{\xi_{k,1}}\tilde{E}_1$ and $\tilde{P}^T_{\xi_{k,1}}\tilde{g}(k)$ via the equation

$$\langle \tilde{P}^T_{\xi_{k,1}}\tilde{E}_1, \tilde{P}^T_{\xi_{k,1}}\tilde{g}(k)\rangle = \cos(\theta)\|\tilde{P}^T_{\xi_{k,1}}\tilde{E}_1\|\|\tilde{P}^T_{\xi_{k,1}}\tilde{g}(k)\|. \qquad (48)$$

Remark B.3: In this development, we do not explicitly fix θ, the desired angel between $\tilde{E}_1$ and $\tilde{g}(k)$. For the simulation, we have been choosing $\theta = \frac{\pi}{3}$.

Observe that the symmetry of the construction (cf. Fig. 5) yields

$$\|\tilde{P}^T_{\xi_{k,1}}\tilde{E}_1\| = \|\tilde{P}^T_{\xi_{k,1}}\tilde{g}(k)\|. \qquad (49)$$

Using this along with the symmetry of $P_{\xi_{k,1}}$, one has from Eq. 48

$$\tilde{E}_1^T\tilde{P}^2_{\xi_{k,1}}\tilde{g}(k) = \cos(\theta)\tilde{E}_1^T\tilde{P}^2_{\xi_{k,1}}\tilde{E}_1, \qquad (50)$$

or

$$\tilde{E}_1^T\tilde{P}^2_{\xi_{k,1}}[\tilde{g}(k) - \tilde{E}_1\cos(\theta)] = 0. \qquad (51)$$

To simplify the computation, let

$$\tilde{E}_1 = \begin{pmatrix} s_1 \\ s_2 \end{pmatrix} \text{ and } \tilde{g}(k) = \begin{pmatrix} s_1 \\ -s_2 \end{pmatrix}. \qquad (52)$$

Substituting Eq. 47 and Eq. 52 into Eq. 51 yields

$$\begin{pmatrix} s_1 & s_2 \end{pmatrix}\begin{pmatrix} (1+\beta)^2 & 0 \\ 0 & 1 \end{pmatrix}\left[\begin{pmatrix} s_1 \\ -s_2 \end{pmatrix} - \begin{pmatrix} s_1 \\ s_2 \end{pmatrix}\cos(\theta)\right] = 0,$$

or

$$s_1[1+2\beta+\beta^2][s_1 - s_1\cos(\theta)] + s_2[-s_2 - s_2\cos(\theta)] = 0.$$

Solving for β yields

$$\beta = -1 \pm \left\|\frac{s_2}{s_1}\right\|\sqrt{\left|\frac{1+\cos(\theta)}{1-\cos(\theta)}\right|}, \qquad (53)$$

where the argument inside the square root can not be negative because of triangular relationships.

Observe that s_1 and s_2 are the orthogonal projections of E_1 onto the axes W_1 and W_2. Thus, one has from Eq. 46 and Eq. 52,

$$s_1 = W_1^T E_1 = \langle W_1, E_1\rangle = \frac{[E_1^T + g^T(k)]}{\|E_1 + g(k)\|}E_1 = \frac{1+\langle E_1, g(k)\rangle}{\|E_1 + g(k)\|},$$

and

$$s_2 = W_2^T E_1 = \langle W_2, E_1\rangle = \frac{[E_1^T - g^T(k)]}{\|E_1 - g(k)\|}E_1 = \frac{1-\langle E_1, g(k)\rangle}{\|E_1 - g(k)\|}.$$

Also, it is easily verified that

$$\|E_1 + g(k)\|^2 = \|E_1\|^2 + \|g(k)\|^2 + 2E_1^T g(k) = 2[1+\langle E_1, g(k)\rangle],$$

and

$$\|E_1 - g(k)\|^2 = \|E_1\|^2 + \|g(k)\|^2 - 2E_1^T g(k) = 2[1-\langle E_1, g(k)\rangle].$$

Therefore, from the above equations, the explicit expression for s_1 and s_2 are

$$s_1 = \frac{1}{2}\|E_1 + g(k)\| \text{ and } s_2 = \frac{1}{2}\|E_1 - g(k)\|. \qquad (54)$$

Choosing the positive square root of Eq. 53 (so that $\beta > -1$) yields an explicit value for β

$$\beta = -1 + \frac{\|E_1 - g(k)\|}{\|E_1 + g(k)\|}\sqrt{\left|\frac{1+\cos(\theta)}{1-\cos(\theta)}\right|}. \qquad (55)$$

Thus, $\tilde{P}_{\xi_{k,1}} \in \Re^{2\times 2}$ of Eq. 47 is uniquely defined.

The matrix $P_{\xi_{k,1}} \in \Re^{3\times 3}$ is now defined by

$$P_{\xi_{k,1}} = \delta_k\left[\mathrm{I}_3 - \begin{pmatrix} W_1 & W_2 \end{pmatrix}\begin{pmatrix} W_1^T \\ W_2^T \end{pmatrix} + \begin{pmatrix} W_1 & W_2 \end{pmatrix}\tilde{P}_{\xi_{k,1}}\begin{pmatrix} W_1^T \\ W_2^T \end{pmatrix}\right], \qquad (56)$$

where $\mathrm{I}_3 \in \Re^{3\times 3}$ is the identity matrix and $\delta_k > 0 \in \Re$ is the scaling factor.

By construction, the matrix $P_{\xi_{k,1}}$ given by Eq. 56 should satisfy Eq. 43. Observe that when $P_{\xi_{k,1}}$ is substituted into Eq. 43, only the last term contributes to the inner product. Thus, Eq. 43 becomes

$$\delta_k^2 E_1^T\begin{pmatrix} W_1 & W_2 \end{pmatrix}\tilde{P}_{\xi_{k,1}}\begin{pmatrix} W_1^T \\ W_2^T \end{pmatrix}g(k) = t(k),$$

or,

$$\delta_k^2\langle P'_{\xi_{k,1}}E_1, P'_{\xi_{k,1}}g(k)\rangle = t(k),$$

where

$$P'_{\xi_{k,1}} = \begin{pmatrix} W_1 & W_2 \end{pmatrix}\tilde{P}_{\xi_{k,1}}\begin{pmatrix} W_1^T \\ W_2^T \end{pmatrix}.$$

Finally, solving for the scaling factor δ_k, one has

$$\delta_k = \sqrt{\frac{t(k)}{\langle P'_{\xi_{k,1}}E_1, P'_{\xi_{k,1}}g(k)\rangle}}, \qquad (57)$$

where the argument of the square root is strictly positive since

$\langle P'_{\xi_{k,1}} E_1, P'_{\xi_{k,1}} g(k)\rangle > 0$ by construction and $t(k) > 0$ by choice (cf. case 2).

Thus, an initial positive definite matrix $Q_{\xi_{k,1}}(0)$ for input into the optimisation procedure, Algorithm 5.1, is given by

$$Q_{\xi_{k,1}}(0) = P_{\xi_{k,1}} P^T_{\xi_{k,1}} = \delta_k^2 P'_{\xi_{k,1}},$$

where

$$P'_{\xi_{k,1}} = I_3 - \begin{pmatrix} W_1 & W_2 \end{pmatrix} \begin{pmatrix} W_1^T \\ W_2^T \end{pmatrix} + \begin{pmatrix} W_1 & W_2 \end{pmatrix} \tilde{P}_{\xi_{k,1}} \begin{pmatrix} W_1^T \\ W_2^T \end{pmatrix}. \quad (58)$$

References

[1] P. V. Osburn, H. P. Whitaker and A. Kezer, "New developments in the design of adaptive control systems", *IAS Paper*, pp.61-39, Institute of Aeronautical Sciences, USA, 1961

[2] P. C. Parks, "Lyapunov redesign of model reference adaptive control systems", *IEEE transaction on automatic control*, vol.11,pp. 362-367, 1966.

[3] B. Shackcloth and R. L. Butchart, "Synthesis of model reference adaptive systems by Lyapunov's second method", *Proceedings of IFAC symposium on the theory of self-adaptive control systems*, Teddington, England, 145-152, 1965.

[4] Y. D. Landau, *Adaptive control: The model reference approach*, Marcel Dekker, New York, 1979.

[5] K. S. Narendra and A. M. Annaswamy, *Stable adaptive systems*, Prentice Hall, Englewood Cliffs, NJ, 1989.

[6] K. J. Astrom and B. Wittenmark, *Adaptive control: 2nd edition*, Addison-Wesley, New York, 1995.

[7] R. L. Kosut, B. D. O. Anderson and I. M. Y. Mareels, "Stability theory for adaptive systems: Methods of averaging and persistency of excitation", *IEEE transaction on automatic control*, vol.32, pp. 26-34, 1987.

[8] B. D. O. Anderson, R. R. Bitmead, C. R. Johnson, P. Kokotovic , R. L. Kosut, I. M. Y. Mareels, L. Praly and B. Riedle, *Stability of adaptive systems: Passivity and averaging analysis*, Cambridge, M.I.T. Press, Massachusetts, 1986

[9] I. M. Y. Mareels and R. R. Bitmead, "Non-linear dynamics in adaptive control: Chaotic and periodic stabilization", *Automatica*, vol. 22, pp. 641-655, 1986.

[10] I. M. Y. Mareels and R. R. Bitmead, "Non-linear dynamics in adaptive control: Chaotic and periodic stabilization - II Analysis", *Automatica*, vol. 24, pp. 485-497, 1988.

[11] M. P. Golden and B. E. Ydstie, "Bifurcation in model reference adaptive control systems", *Systems and control letters*, vol.11, pp. 413-430, 1988.

[12] P. Kokotovic, I. Kanellakopoulos and M. Krstic, "On letting adaptive control be what it is: Nonlinear feedback", *IFAC adaptive systems in control and signal processing*, Grenoble, France, 1992.

[13] M.Krstic, I. Kanellakopoulos and P. Kokotovic, "Nonlinear design of adaptive controller for linear systems", *IEEE transactions on automatic control*, vol. 39, pp. 738-752, 1994.

[14] M.Krstic, I. Kanellakopoulos and P. Kokotovic, *Nonlinear and adaptive control design*, John Wiley & sons, New York, 1995.

[15] Z.-P. Jiang and L. Praly, "A self-tuning robust nonlinear controller", *IFAC Congress 96*, San Francisco, USA, 1996.

[16] U. Helmke and J.B. Moore, *Optimisation and dynamic systems*, Springer-Verlag, London, 1994

[17] M.P. do Carmo, *Riemannian geometry*, Birkhauser, Boston, 1992.

[18] L. Vandenberghe and S. Boyd, "Semidefinate programming", *SIAM review*, vol. 38, No. 1, pp. 49-95, 1996

[19] Y. Nesterov and A. Nemirovskii, "Interior-point polynomial algorithms in convex programming", *Society for industrial and applied mathematics*, Philadelphia, pensylvania, 1994.

Sang-Heon LEE

Obtained BEng. in aeronautical Eng. from Inha University, Korea, a MEngSci. in mechatronics from the University of New South Wales, Australia and a Ph.D. in systems engineering from the Australian National University in 1998. From Nov. 1998 to May 2000 he held a post as a post-doctoral fellow in the Dept. of Elec. Eng in Australian Defence Force Academy. Since July 2000, he has held the post of research leader in robotics and automatic control at the Centre for Advanced Manufacturing Research, University of South Australia, Australia. His research interests are in structural discrete-event systems, dual robot prototyping systems, machine vision, integrated on-line robot control systems, adaptive control, fuzzy logic control and neural networks.

Robert MAHONY

Obtained Bsc. in applied mathematics and geology from the Australian National University (ANU) in 1989. After working for a year as a geophysicist processing marine seismic data he returned to study at ANU and obtained a Ph.D. in systems engineering in 1994. Between 1994and 1997 he worked as a Research Fellow in the Cooperative Research Centre for Robust and Adaptive Systems based in the Research School of Information Sciences and Engineering, ANU, Australia. From 1997 to 1999 he held a post as a post-doctoral fellow in the CNRS laboratory for Heuristics Diagnostics and complex systems (Heudiasyc), Compiegne University of Technology, FRANCE. Between 1999 and 2001 he held a Logan Fellowship in the Department of Engineering and Computer Science at Monash University, Melbourne, Australia. Since July 2001 he has held the post of senior lecturer in mechatronics at the Department of Engineering, ANU, Canberra, Australia. His research interests are in non-linear control theory with applications in mechanical systems and motion systems, mathematical systems theory and geometric optimisation techniques with applications in linear algebra and digital signal processing.