J.W. Polderman

A state space approach to the problem
of adaptive pole assignment

A State Space Approach to the Problem of Adaptive Pole Assignment

J. W. Polderman

Centre for Mathematics and Computer Science
P.O. Box 4079, 1009 AB Amsterdam, The Netherlands

An algorithm for adaptive pole-placement for the class of single-input/single-output systems of order n is proposed. The asymptotic properties of the algorithm do not depend on persistently exciting signals. Excitation is used only initially to avoid pole-zero cancellation of the parameter estimates. The main result is that the asymptotic behaviour of the system equals the behaviour one would have obtained on the basis of the true system. Since this does not imply full identification of the desired control law, we propose the term weak self-tuning. The reason that such a result can be obtained without identification of the true system is the following: Suppose we have a wrong estimate of the system, and that based on that estimate we generate the controls, and that the incorrectness of the estimate is not revealed by the resulting closed-loop behaviour of the system, then the inputs are exactly equal to the ones we would have applied if we had known the system.

1. INTRODUCTION

This paper deals with the problem of adaptive pole assignment of single-input/single-output linear time-invariant systems of which only the order is assumed to be known. This problem has received considerable attention in the literature. Several approaches have been proposed. Pole assignment can be treated in the framework of model reference adaptive control. This has for instance been done in [16]. A drawback of this method is that due to the used convergence analysis only minimum phase systems can be handled. In [3,4,5,7,9] the problem is studied for general systems in input/output form with the only assumption that the order of the system is known. Algorithms based on parameter estimation and the certainty equivalence principle are proposed. In all of these papers stability results are derived under additional assumptions. The main reason that these assumptions have to be made seems to be that during the estimation procedure (unstable) pole-zero cancellation can occur. This can be avoided by assuming extra knowledge of the true system, which reduces the results essentially to local ones. Another way of avoiding that parameter estimates eventually have common factors is to use sufficiently exciting signals to assure convergence of the estimates to the true parameter value. It should be clear that additional injected signals can influence the performance of the system negatively, moreover it is always difficult to guarantee internal excitation by means of conditions on an external signal, since external excitation may be annihilated by unpredictable signals in the feedback loop.

In [10,12] algorithms are presented that overcome this difficulty. In [12] the parameter estimates are modified to keep them away from the boundary of the "dangerous region". In [10] the identification is done simultaneously in the parameter space and the controller space. By using an extra (non-linear) feedback driven by the discrepancy between the controller estimate and the expected behaviour on the basis of the parameter estimate, these two schemes are brought into agreement with each other. The extra feedback can be interpreted as an asymptotic vanishing excitation signal. A drawback of both

In this paper an adaptive pole assignment algorithm is proposed which does not call for external

exciting inputs and where no a priori information of the system is needed either. The underlying idea is that complete knowledge of the system parameters is not needed in order to be able to generate the proper sequence of inputs, and hence one should not make any attempt to learn the system completely. This observation is based on an analysis of the asymptotic structure of the problem. By this we mean the following. Generally speaking, in adaptive control one has a (partially) unknown plant and a control objective. The sequence of inputs that reaches the objective will in general depend on the unknown system, hence in one way or an other one should try to gain knowledge of the system on the basis of, initially, arbitrary inputs and the resulting outputs. Then, based on the information one can calculate the to be applied inputs more accurately. What one then hopes is that by applying this procedure the information about the system will eventually be sufficient to generate the desired control inputs. However, this procedure leads inevitably to lack of internal excitation, since information of the system will be obtained on the basis of closed-loop behaviour only. We believe that this is a fundamental problem in adaptive control. One way of dealing with this problem is to inject a sufficiently rich signal to assure internal excitation, and hence complete knowledge of the system. But before deciding to do that, it might be a good idea to investigate whether or not lack of internal excitation leads to sub-desired behaviour. One of the key observations of this work is that in the case of pole assignment the information one obtains from the closed-loop behaviour of the system without external signals, is just sufficient to generate the proper sequence of inputs. We believe that this property can be seen as a kind of a posteriori justification for a lot of existing algorithms. Based on this observation an algorithm will be derived which will asymptotically result in exactly the same closed-loop behaviour as one would have obtained on the basis of complete knowledge of the system. The problem of pole-zero cancellation in the estimation procedure is avoided by monitoring how far away we are from cancellation. If we are too close, we apply a special (finite) sequence of inputs, to bias the estimates from cancellation. The crucial novelty is then that we can prove that this has to be done only a finite number of times.

We will use both the input/output description as well as the input/state/output representation of the systems at hand. The state space description is more convenient for parts of the analysis and to state the results in a clear fashion, whereas the input/output description seems to be the right tool for the estimation part of the algorithm.

Our main result is that the asymptotic closed-loop behaviour of the adaptive controlled system equals the behaviour we would have obtained knowing the true system parameters. The proof of this result is independent of the desired pole locations. Hence even in the somewhat unrealistic situation where one wants to place the closed-loop poles in the unstable region our algorithm is applicable. This may look purely academic, but it shows that the adaptation of the controller parameters does not depend on stability properties of the system. The reason that we are able to derive such a result is that we consider the unknown parameters in the state space description as linear maps of which we want to know the action on certain subspaces. The variables on which these maps act can then be normalized without losing any information.

The paper is organized as follows. In section 2 we will formulate the problem statement. First we will give the non-adaptive problem and then its adaptive counterpart. A general theorem will be given which relates the adaptive requirement to the resulting closed-loop behaviour of the system. In section 3 we make an attempt to reveal some of the basic principles on which an algorithm should be based. These principles are widely used but seldom commented upon. In section 4 a detailed study is made of how much information of the system parameters we need for generating the appropriate sequence of controls, and how much information we can obtain from the closed-loop behaviour. In sections 5 and 6 algorithms will be presented for the observed state case and the non-observed state case respectively. To illustrate the algorithms, simulation results will be discussed in section 7. Finally, in section 8 we will draw some conclusions.

2. Preliminaries

Consider the following time-invariant finite dimensional linear system:

$$x(k+1) = Ax(k) + bu(k), \quad x(0) \tag{2.1.a}$$

$$y(k) = cx(k) \tag{2.1.b}$$

$(A,b,c) \in E$, where

$$E := \{(A,b,c) \in \mathbb{R}^{n \times n} \times \mathbb{R}^{n \times 1} \times \mathbb{R}^{1 \times n} \mid (A,b,c) \text{ minimal }\} \tag{2.2}$$

Also define:

$$\hat{E} := \{(A,b) \in \mathbb{R}^{n \times n} \times \mathbb{R}^{n \times 1} \mid (A,b) \text{ controllable }\} \tag{2.3}$$

Let $\Lambda := \{\lambda_1,..,\lambda_n\} \subset \mathbb{C}$ be such that $\lambda \in \Lambda \Rightarrow \bar{\lambda} \in \Lambda$. Let the control objective be the assignment of the closed-loop poles to the configuration Λ. Define $\sigma \in \mathbb{R}[X]$ by: $\sigma(X) = \prod_{i=1}^{n}(X - \lambda_i)$.

Define $f : E \to \mathbb{R}^{1 \times n}$ by:

$$f(A,b) := -[0...01] \, [b \vdots Ab \vdots .... \vdots A^{n-1}b]^{-1} \sigma(A) \tag{2.4}$$

Then the characteristic polynomial of $A + bf(A,b)$ is exactly σ, and moreover, since the system is single-input, $f(A,b)$ is the only feedback law with that property. (see [17]).

Suppose now that the true value (A_0,b_0,c_0) of the system parameters is unknown. Then the control objective has to be replaced by a weaker one. As a modified version of the original control objective we choose the following:

> Generate a sequence of inputs such that asymptotically the applied inputs equal the inputs that would have been calculated on the basis of the true system parameters. Moreover, the resulting controlled system should be stable, provided that Λ is contained in the unit disk.

The following theorem relates the above described requirements to the resulting closed-loop behaviour of the system.

2.1 Let $(A,b) \in \mathbb{R}^{n \times n} \times \mathbb{R}^{n \times 1}$, not necessarily controllable, and let $f \in \mathbb{R}^{1 \times n}$. Let the sequence $\{u(k)\}_{k \in \mathbb{N}}$ and $x(0) \in \mathbb{R}^n$ be given. Define $x(k)$ by:

$$x(k+1) = Ax(k) + bu(k) \tag{2.5}$$

Assume that for all k: $x(k) \neq 0$ and suppose:

$$\lim_{k \to \infty} \left| \frac{u(k) - fx(k)}{\|x(k)\|} \right| = 0 \tag{2.6}$$

Then there exists a sequence of matrices $\{\Delta_k\}_{k \in \mathbb{N}}$ such that for all k:

$$i) \quad x(k+1) = (A + bf + \Delta_k)x(k) \tag{2.7}$$

$$ii) \quad \lim_{k \to \infty} \Delta_k = 0 \tag{2.8}$$

Proof Define:

$$\Delta_k := \frac{b(u(k) - fx(k))x(k)^T}{x(k)^T x(k)} \tag{2.9}$$

then one can easily check that 2.7 holds. 2.8 follows from 2.6.

Comment Theorem 2.1 tells us that if the input of a linear system is asymptotically given by state feedback, then asymptotically the system will behave as if this feedback was used. This result holds whether or not f is stabilizing. An important feature in the assumption of the theorem is the

4

normalization. This decouples the result from the norm of the state trajectory, and emphasizes that everything depends only on the directions of the states.

If we replace (A,b) by (A_0,b_0) and f by $f(A_0,b_0)$, we see what kind of result we get if we are able to produce a sequence of inputs which satisfies (2.6). In section 5 and 6 we will present algorithms which produce such sequences.

3. ADMISSIBLE ALGORITHMS

In this section we will define a class of admissible adaptive pole assignment algorithms. There are two reasons why we think that at least an attempt in this direction should be made. The first reason is that we want to rule out algorithms from which it is intuitively clear that they will give bad behaviour, although mathematically things look good. Since we deal with deterministic systems we can, mathematically speaking, identify the system exactly within finite time. After having done so, we can then calculate the control law and apply it for ever. It should be clear that this kind of open-loop adaptive control will not be very robust against (slow) variations of the system parameters.

The second reason is that we want to make the principles that lead to the development of the algorithms proposed in sections 5 & 6 more explicit. It will turn out that our algorithm is just an example of an admissible algorithm that behaves well asymptotically. But it is quite possible that there are better algorithms within the class defined below.

DEFINITION 3.1 An admissible adaptive pole assignment algorithm is a procedure consisting of two parts: An estimation part and a control part. Furthermore there is a rule which connects these two parts. The procedure should obey the following rules.

(i) RECURSIVENESS At every time instant a new estimation of the system parameters (or a function of them) is made on the basis of a uniformly bounded number of past observations and past estimations.

(ii) NEUTRALITY The estimate at time $k+1$ equals the estimate at time k if and only if there is no discrepancy between the observed data and the data expected on the basis of the estimate at time k.

(iii) CERTAINTY EQUIVALENCE The estimation part and the control part are connected according to the certainty equivalence principle: The control action at time k, for almost all k is calculated on the basis of the latest estimate, as if this estimate represents the true system. With almost always we understand: always except for a finite number of times. At those time instants a alternative control action may be taken.

An admissible algorithm is called regular if it does not use the possibility of taking alternative control actions. It is called irregular otherwise.

COMMENT If the algorithm does not depend on time, then neutrality and certainty equivalence enter quite naturally. For, suppose we are lucky and the initial guess of the system parameters happens to be the true one, then the control action should equal the desired one. Since this should hold for every possible value of the system parameters certainty equivalence and neutrality follow. Another motivation for neutrality is the following: Suppose that the present estimate is not falsified by the newly observed data, why should one then change the estimate and moreover in which direction should this be done. The escape possibility of taking alternative control actions at a finite number of time instants is provided for the following reason. Since the estimation part produces estimates in the space of minimal triples, one should be able to avoid that estimates eventually become non-minimal. Our algorithms make use of this possibility.

Note that by definition it is not allowed to use external excitation signals.

In the next section we will investigate the question whether there is any chance that there exist admissible algorithms which give good asymptotic behaviour.

4. How much can we learn, and how much should we learn?

In this section we shall give a characterization of how much we can learn from the true system when the control sequence is generated by a regular admissible algorithm. The reason that we restrict our attention to regular algorithms (i.e. algorithms which generate *every* input on the basis of the latest estimate) is that for every admissible algorithm there is a time instant after which the algorithm behaves regular. Since we want to draw conclusions about possible limit points of an unspecified admissible algorithm, we should not include algorithms with very specific initial behaviour.

DEFINITION 4.1 Let T be a regular admissible algorithm. We call $(A,b,c,\hat{x}(0))$ indistinguishable from $(A_0,b_0,c_0,x(0))$ under the use of T, if and only if (A,b,c) is invariant under T, provided that the observed data is produced by (A_0,b_0,c_0), and the input sequence is generated by T. We call $(A,b,c,\hat{x}(0))$ indistinguishable from $(A_0,b_0,c_0,x(0))$ if and only if $(A,b,c,\hat{x}(0))$ is indistinguishable under the use of T, for every regular admissible T.

COMMENT The above definition is motivated by the following observation. Suppose we have a guess $(A,b,c,\hat{x}(0))$ of the system which happens to satisfy the requirements of the Definition 4.1. Suppose now that we apply a regular admissible algorithm to the system with initial value $(A,b,c,\hat{x}(0))$. Since $(A,b,c,\hat{x}(0))$ is invariant under the algorithm, the next and all the successive estimates will be (A,b,c). Hence by applying a regular algorithm we can never tell whether the true system parameters were (A,b,c) or (A_0,b_0,c_0). In that sense $(A,b,c,\hat{x}(0))$ cannot be distinguished from $(A_0,b_0,c_0,x(0))$. Note that the relation defined above is non symmetric and non transitive.

PROPOSITION 4.2 If the state trajectory is observed, then $(A,b,c,x(0))$ is indistinguishable from $(A_0,b_0,c_0,x(0))$ if and only if for all k:

$$(A_0+b_0f(A,b))x(k) = (A+bf(A,b))x(k), \tag{4.1}$$

where $x(k)$ is defined by:

$$x(k+1) = (A_0+b_0f(A,b))x(k) \tag{4.2}$$

PROOF Suppose $(A,b,c,x(0))$ is indistinguishable from $(A_0,b_0,c_0,x(0))$. Take (A,b) as the initial state of any admissible algorithm. According to the certainty equivalent principle we should then apply $u(0)=f(A,b)x(0)$. Now observe the new state $x(1)$. Since (A,b) is invariant under the algorithm, we conclude that:

$$x(1) = (A_0+b_0f(A,b))x(0) = (A+bf(A,b))x(0), \tag{4.3}$$

according to the neutrality principle. For general k the same conclusion holds.
Suppose now that for all k (4.1) holds, then again by the neutrality principle, (A,b) will be invariant under any admissible algorithm.

PROPOSITION 4.3 If the state trajectory is not observed, then $(A,b,c,z(0))$ cannot be distinguished from $(A_0,b_0,c_0,x(0))$ if and only if for all k:

$$y(k) = \tilde{y}(k), \tag{4.4}$$

where $y(k)$ and $\tilde{y}(k)$ are defined by:

$$x(k+1) = A_0x(k)+b_0f(A,b)z(k) \qquad ,x(0) \tag{4.5a}$$

$$y(k) = c_0x(k) \tag{4.5b}$$

$$z(k+1) = Az(k)+bf(A,b)z(k) \qquad ,z(0) \tag{4.6a}$$

$$\tilde{y}(k) = cz(k) \tag{4.6b}$$

PROOF The proof is completely analogous to that of Proposition 4.2.

COMMENT The essential conclusion of the above results is that if we have a guess of the true system which happens to be indistinguishable from the true system under the use of any admissible algorithm, we will not change this guess. The crucial question then arises what the consequences are for the resulting input sequence. The reassuring answer is that both in the observed state and in the non observed state case the applied input sequence is then exactly the desired one. The following theorems formalize this statement.

THEOREM 4.4 Suppose that for all k:

$$(A_0 + b_0 f(A,b))x(k) = (A + bf(A,b))x(k),$$ (4.7)

where $x(k)$ is defined as in (4.2); then for all k:

$$f(A,b)x(k) = f(A_0,b_0)x(k)$$ (4.8)

THEOREM 4.5 Suppose we have $(A_0,b_0,c_0,x(0))$ and $(A,b,c,z(0))$ for which (4.4), (4.5) and (4.6) hold, then for all k:

$$f(A,b)z(k) = f(A_0,b_0)x(k)$$ (4.9)

REMARK From Theorems 4.4 and 4.5 it follows that once we have a guess of the true system which is indistinguishable from the true system under the use of any admissible algorithm, the resulting input sequence is exactly the desired one. This reflects in a certain sense a narrow escape. For the minimum information about (A_0,b_0,c_0) is that we should be able to calculate the desired input at every time instant, i.e. $f(A_0,b_0)x(k)$. On the other hand it is obvious that the maximum information we can get from the true system by means of an admissible algorithm is described by 4.4, 4.5, and 4.6. Hence the maximum we can get is the minimum we need. This can be rephrased by saying that there is no conflict between closed-loop identification and pole assignment. It should be noted that this does not hold for general control criteria, see for instance [15].

For the proofs of Theorems 4.4 and 4.5 we will use the following two results, which we will prove in the appendix to this section.

THEOREM 4.6 Let $(A_0,b_0),(A,b) \in \hat{E}$ and $\mathcal{V}$ a linear subspace of $\mathbb{R}^{n \times n}$ such that:

i) For all $v \in \mathcal{V}$: $(A_0 + b_0 f(A,b))v \in \mathcal{V}$ (4.10)

ii) For all $v \in \mathcal{V}$: $(A_0 + b_0 f(A,b))v = (A + bf(A,b))v$ (4.11)

Then:

for all $v \in \mathcal{V}$: $f(A,b)v = f(A_0,b_0)v$. (4.12)

PROOF See the appendix to this section.

THEOREM 4.7 Let $\{(u(k),y(k))\}_{k \in \mathbb{N}}$ be a sequence in $\mathbb{R}^2$ and suppose there exist (A_1,b_1,c_1), (A_2,b_2,c_2), minimal triples of order n, and sequences $\{x(k)^{(1)},x(k)^{(2)}\}$ in $\mathbb{R}^n$, such that for all k:

$$x(k+1)^{(1)} = A_1 x(k)^{(1)} + b_1 u(k)$$ (4.13)
$$x(k+1)^{(2)} = A_2 x(k)^{(2)} + b_2 u(k)$$ (4.14)

$$y(k) = c_1 x(k)^{(1)} \tag{4.15}$$

$$y(k) = c_2 x(k)^{(2)} \tag{4.16}$$

Define $\mathfrak{X}_i = \text{span } \{x(k)\}_{k \in \mathbb{N}}$, and $d_i = \dim (\mathfrak{X}_i)$, $i = 1,2$.

(i) if $d_1 < n$, then there exists a non-singular matrix S, such that: $Sx(k)^{(1)} = x(k)^{(2)}$.

(ii) $d_1 = d_2$.

(iii) if there exists g_1 such that: $u(k) = g_1 x(k)^{(1)}$, then there exists a non-singular matrix S, such that: $Sx(k)^{(1)} = x(k)^{(2)}$.

PROOF See the appendix to this section.

PROOF OF THEOREM 4.4:

Suppose that (4.7) holds. Define $\mathfrak{X} := \text{span } \{x(k)\}_{k \in \mathbb{N}}$. Then : $(A_0 + b_0 f(A,b))\mathfrak{X} \subset \mathfrak{X}$. Moreover, for all $x \in \mathfrak{X}$: $(A_0 + b_0 f(A,b))x = (A + bf(A,b))x$. Hence by Theorem 4.6 we have for all $x \in \mathfrak{X}$: $f(A,b)x = f(A_0,b_0)x$. In particular: $f(A,b)x(k) = f(A_0,b_0)x(k)$.

PROOF OF THEOREM 4.5:

Suppose (4.4), (4.5) and (4.6) hold. Then, by putting $u(k) = f(A,b)z(k)$, the conditions of Theorem 4.7 are satisfied. Hence there exists a non-singular matrix S such that for all k: $Sx(k) = z(k)$. By replacing $z(k)$ by $S^{-1}x(k)$ in (4.5a) and (4.6a) we obtain two recursions for $x(k)$:

$$x(k+1) = (A_0 + b_0 f(A,b)S^{-1})x(k) \tag{4.17}$$

and:

$$x(k+1) = S(A + bf(A,b))S^{-1}x(k) \tag{4.18}$$

Define: $\tilde{A} := SAS^{-1}$, and $\tilde{b} := Sb$, then (4.17) and (4.18) can be written as:

$$x(k+1) = (A_0 + b_0 f(\tilde{A},\tilde{b}))x(k) \tag{4.19}$$

and:

$$x(k+1) = (\tilde{A} + \tilde{b}f(\tilde{A},\tilde{b}))x(k) \tag{4.20}$$

Hence, by Theorem 4.4 it follows that for all k: $f(\tilde{A},\tilde{b})x(k) = f(A_0,b_0)x(k)$, which is equivalent to: $f(A,b)z(k) = f(A_0,b_0)x(k)$.

APPENDIX.

PROOF OF THEOREM 4.6 Suppose that $\Lambda \subset \mathbb{R}$ and that $\lambda_i \neq \lambda_j$ for all $i \neq j$. Let $\mathcal{V}$ be one-dimensional. Then $\mathcal{V}$ is generated by an eigenvector v of $(A + bf(A,b))$ corresponding to let's say $\lambda := \lambda_i$. Hence by (4.11): $(A_0 + b_0 f(A,b))v = \lambda v$. Suppose (A_0,b_0) is in standard controllable form. Then $v = [1,\lambda,..,\lambda^{n-1}]^T$. The spectrum of $A_0 + b_0 f(A_0,b_0)$ is by definition of f equal to Λ. Hence λ is an eigenvalue of $(A_0 + b_0 f(A_0,b_0))$, and there exists $\tilde{v}$ such that $(A_0 + b_0 f(A_0,b_0))\tilde{v} = \lambda\tilde{v}$. From the standard controllable form it is easy to see that the only candidates for an eigenvector with eigenvalue λ are multiples of v, hence $\tilde{v} = \mu v$, for some $\mu \neq 0$. Hence $(A_0 + b_0 f(A_0,b_0))v = (A_0 + b_0 f(A,b))v$. Since $b_0 \neq 0$, we conclude that $f(A,b)v = f(A_0,b_0)v$.

If $\dim \mathcal{V} > 1$, then $\mathcal{V}$ has a basis of eigenvectors and the above reasoning gives the result. For general Λ the proof goes along the same lines, but then one has to study several different cases. We skip the details.

For the proof of Theorem 4.7 we will use the following:

LEMMA 4.8. Let $(A,b) \in \hat{E}$ and $x(0) \in \mathbb{R}^n$. Let $\{u(k)\}$ be a sequence of real numbers. Define:

$$x(k+1) = Ax(k)+bu(k) \quad k = 0,1,2,... \tag{4.21}$$

Define $\mathcal{X}:=$ span $\{x(k)\}_{k\in\mathbb{N}}$, and $d:=\dim(\mathcal{X})$. If $d<n$, then there exists a $g\in\mathbb{R}^{1\times n}$ such that for all k:

$$u(k) = gx(k) \tag{4.22}$$

PROOF Suppose (A,b) is in standard controllable form i.e.

$$A = \begin{bmatrix} 0 & 1 & : & : & 0 \\ : & 0 & : & : & : \\ : & : & : & : & 0 \\ 0 & 0 & : & : & 1 \\ a_1 & a_2 & : & : & a_n \end{bmatrix} \qquad b = \begin{bmatrix} 0 \\ : \\ 0 \\ 1 \end{bmatrix} \tag{4.23}$$

Define $a\in\mathbb{R}^{1\times n}$ by: $a:=(a_1,..,a_n)$ Define:

$$\tilde{A}:=A-ba \tag{4.24}$$

$$\tilde{u}:=ax(k)+u(k) \tag{4.25}$$

then:

$$x(k+1) = \tilde{A}x(k)+b\tilde{u}(k) \tag{4.26}$$

Suppose $x(0)=[x_1(0),..,x_n(0)]^T$, define $H\in\mathbb{R}^{n\times\mathbb{N}}$ by:

$$H := [x(0),x(1),x(2),x(3),....] \tag{4.27}$$

then:

$$H := \begin{bmatrix} x_1(0) & x_2(0) & x_3(0) & . & . & . & x_n(0) & \tilde{u}(0) & . & . & . \\ : & : & : & & & & \tilde{u}(0) & & : & \\ : & x_n(0) & \tilde{u}(0) & & & & : & & : & \\ x_n(0) & \tilde{u}(0) & \tilde{u}(1) & . & . & . & \tilde{u}(n-2) & \tilde{u}(n-1) & . & . & . \end{bmatrix} \tag{4.28}$$

Since $d<n$, rank$(H)<n$. Now H is a truncated Hankel matrix, hence its rank does not increase if we add the last row, shifted to the left, as the $n+1$-th row. This shifted row is:

$$[\tilde{u}(0),\tilde{u}(1),\tilde{u}(2),\tilde{u}(3),....] \tag{4.29}$$

Since the rank of the increased matrix is equal to the original one, the last row is a linear combination of the first n rows. In other words, there exist $\tilde{g}_1,\ldots,\tilde{g}_n\in\mathbb{R}$ such that:

$$r_{n+1} = \tilde{g}_1 r_1 +...+\tilde{g}_n r_n \tag{4.30}$$

where r_i denotes the i-th row. Define $\tilde{g}\in\mathbb{R}^{1\times n}$ by: $\tilde{g} := [\tilde{g}_1,\ldots,\tilde{g}_n]$. Then for all k:

$$\tilde{u}(k) = \tilde{g}x(k) \tag{4.31}$$

Define g by: $g:=\tilde{g}-a$. Finally:

$$u(k) = \tilde{u}_k-ax(k) \tag{4.32}$$

$$= \tilde{g}x(k)-ax(k) \tag{4.33}$$

$$= gx(k) \tag{4.34}$$

We will now prove Theorem 4.7:

PROOF OF THEOREM 4.7:

i). Suppose $d_1 < n$.

$$y(k+i) = c_1 A_1^i x(k)^{(1)} + \sum_{j=0}^{i-1} A_1^j b_1 u(k+i-j-1) \tag{4.35}$$

Define:

$$W := \begin{bmatrix} c_1 \\ c_1 A_1 \\ \vdots \\ c_1 A_1^{n-1} \end{bmatrix} \tag{4.36}$$

then:

$$Wx(k) = \begin{bmatrix} y(k) \\ y(k+1) - c_1 b_1 u(k) \\ \vdots \\ y(k+n-1) - A_1^{n-2} bu(k) - \dots - c_1 b_1 u(k+n-2) \end{bmatrix} \tag{4.37}$$

From which we conclude that:

$$\begin{bmatrix} W & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} x(k)^{(1)} \\ u(k) \\ \vdots \\ \vdots \\ \vdots \\ u(k+n-2) \end{bmatrix} = \begin{bmatrix} 1 & 0 & . & 0 \\ 0 & . & -c_1 b_1 & . & : \\ : & . & : & . & : \\ : & & 1 & -c_1 A_1^{n-2} b_1 & . & -c_1 b_1 \\ : & & & 1 & . & 0 \\ : & & : & . & : \\ 0 & & 0 & . & 1 \end{bmatrix} \begin{bmatrix} y(k) \\ : \\ : \\ y(k+n-1) \\ u(k) \\ : \\ u(k+n-2) \end{bmatrix} \tag{4.38}$$

From which we derive:

$$\begin{bmatrix} x_1^{(1)}(k) \\ : \\ x_n^{(1)}(k) \\ u(k) \\ : \\ u(k+n-2) \end{bmatrix} = T_1 \begin{bmatrix} y(k) \\ : \\ y(k+n-1) \\ u(k) \\ : \\ u(k+n-2) \end{bmatrix} \tag{4.39}$$

In the same way one derives that:

$$\begin{bmatrix} x_1^{(2)}(k) \\ : \\ x_n^{(2)}(k) \\ u(k) \\ : \\ u(k+n-2) \end{bmatrix} = T_2 \begin{bmatrix} y(k) \\ : \\ y(k+n-1) \\ u(k) \\ : \\ u(k+n-2) \end{bmatrix} \tag{4.40}$$

Hence:

$$\begin{bmatrix} x(k)^{(1)} \\ u(k) \\ \vdots \\ u(k+n-2) \end{bmatrix} = R \begin{bmatrix} x(k)^{(2)} \\ u(k) \\ \vdots \\ u(k+n-2) \end{bmatrix} \tag{4.41}$$

where $R = T_1 T_2^{-1}$. Now since $u(k+i) = b_2^{\#}(x(k+i+1)^{(2)} - A_2 x(k+i)^{(2)})$, there exist matrices $M_1^{(2)}, ..., M_n^{(2)} \in \mathbb{R}^{n \times n}$, such that for all k:

$$x(k)^{(1)} = M_1^{(2)} x(k)^{(2)} + ... + M_n^{(2)} x(k+n-1)^{(2)} \tag{4.42}$$

and similarly:

$$x(k)^{(2)} = M_1^{(1)} x(k)^{(1)} + ... + M_n^{(1)} x_{k+n-1}^{(1)} \tag{4.43}$$

Since by assumption $d < n$, we conclude from Lemma 4.9 that there exists g_1 such that $u(k) = g_1 x(k)^{(1)}$, hence $x(k+1)^{(1)} = (A_1 + b_1 g_1) x(k)^{(1)}$. Together with (4.43) this gives that there exists a matrix N_1 such that for all k:

$$x(k)^{(2)} = N_1 x(k)^{(1)} \tag{4.44}$$

Denote by $\mathcal{X}_2$ the linear span of $x(k)^{(2)}_{k \in \mathbb{N}}$, and by d_2 its dimension. From (4.44) it follows that: $d_2 \leq d_1$, hence by Lemma 4.9 there exists g_2 such that for all k: $u(k) = g_2 x(k)^{(2)}$. As above we conclude that there exists a matrix N_2 such that for all k:

$$x(k)^{(1)} = N_2 x(k)^{(2)} \tag{4.45}$$

Finally (4.44) together with (4.45) gives the statement.

ii) This follows immediately form part *i*).

iii) Suppose $u(k) = g_1 x(k)^{(1)}$, then just as in the proof of part (*i*) ((4.43)), we conclude that:

$$x(k)^{(2)} = N_1 x(k)^{(1)}$$

Since $d_1 = d_2 = n$, it follows that N_1 is non-singular.

5. AN ADMISSIBLE ALGORITHM FOR THE OBSERVED STATE CASE

We will now propose an admissible algorithm for the case that the state of the system (2.1) is observed. This is of course not a very realistic situation, but on a conceptual level it provides a good preparation for the non-observed state case. The algorithm is a modification of the one described in [13,14]. There an essential assumption was made on the controllability of the limit points of the sequence of parameter estimates. By introducing an alternative procedure when the parameter estimates are close to non-controllable, this assumption is relaxed. A drawback is that the analysis of the algorithm becomes a little bit more complicated. However, the superficial reader may take it for granted that parameter estimates and their limit points are controllable, without losing appreciation of what is going on.

ALGORITHM 5.2

We will introduce the algorithm inductively. Choose any sequence $\{\epsilon_k\}$ and any sequence $\{C_k\}$ such that:

$$\epsilon_k \downarrow 0 \quad \text{and} \quad C_k \uparrow \infty \tag{5.1}$$

INITIALIZATION $(\hat{A}_0, \hat{b}_0)$: arbitrarily, $h_0 = 0$, $j_0 = 0$, $x(0)$: given.

RECURSION

$$\delta_k = d^*((\hat{A}_k, \hat{b}_k), \hat{E}_k^c) \tag{5.2}$$

---

* d means Euclidean distance. As proven in [2], δ_k is the smallest singular value of the controllability matrix of $(\hat{A}_k, \hat{b}_k)$.

if $h_k = 0$

   then:

    { if $\delta_k \geq \epsilon_{j_k}$

      then:

$$u(k) = f(\hat{A}_k, \hat{b}_k)x(k) \tag{5.2.a.1}$$

$$h_{k+1} = h_k \tag{5.2.b.1}$$

$$j_{k+1} = j_k \} \tag{5.2.d.1}$$

     else (if $\delta_k < \epsilon_{j_k}$)

$$u(k) = C_{j_k} \|x(k)\| \tag{5.2.a.2}$$

$$h_{k+1} = n \tag{5.2.b.2}$$

$$\tau_{j_k} = k \tag{5.2.c.2}$$

$$j_{k+1} = j_k + 1 \}\} \tag{5.2.d.2}$$

else (if $h_k > 0$)

$$u(k) = 0 \tag{5.2.a.3}$$

$$h_{k+1} = h_k - 1 \tag{5.2.b.3}$$

$$j_{k+1} = j_k \} \tag{5.2.d.3}$$

$$\{ x(k+1) = A_0 x(k) + b_0 u(k) \tag{5.2.e}$$

$$\hat{x}(k+1) = \hat{A}_k x(k) + \hat{b}_k u(k) \tag{5.2.f}$$

$$\hat{A}_{k+1} = \hat{A}_k + (\|u(k)\|^2 + \|x(k)\|^2)^{-1}(x(k+1) - \hat{x}(k+1))x(k)^T \tag{5.2.g}$$

$$\hat{b}_{k+1} = \hat{b}_k + (\|u(k)\|^2 + \|x(k)\|^2)^{-1}(x(k+1) - \hat{x}(k+1))u(k) \} \tag{5.2.h}$$

REMARK The division in (5.2.g) and (5.2.h) can of course only be done if $x(k)$ or $u(k)$ is non-zero. Therefore if $x(k_0) = 0$ for some k_0, we do not change the estimates anymore and we take $u(k) = 0$ for all $k \geq k_0$. For the analysis of the algorithm we will assume that $x(k) \neq 0$ for all k. This assumption is also needed if we want to apply Theorem 2.1.

COMMENT Let us first explain how the $(k+1)-th$ estimate, $(\hat{A}_{k+1}, \hat{b}_{k+1})$ of (A_0, b_0) is calculated from $(\hat{A}_k, \hat{b}_k, u(k))$. Suppose $(\hat{A}_k, \hat{b}_k)$ has been calculated and that $u(k)$ has been applied to the true system, this gives:

$$x(k+1) = A_0 x(k) + b_0 u(k) \tag{5.3}$$

Define

$$\hat{G}_{k+1} := \{(A,b) \in \mathbb{R}^{n \times n} \times \mathbb{R}^{n \times 1} \mid Ax(k) + bu(k) = x(k+1) \} \tag{5.4}$$

then $\hat{G}_{k+1}$ is exactly the set of those parameters that are able to explain the transition from $x(k)$ to $x(k+1)$, given $u(k)$. Since a fortiori $(A_0, b_0) \in \hat{G}_{k+1}$, it is natural to choose $(\hat{A}_{k+1}, \hat{b}_{k+1})$ somewhere in $\hat{G}_{k+1}$. Since $\hat{G}_{k+1}$ is linear affine, we can take the orthogonal projection of $(\hat{A}_k, \hat{b}_k)$ on $\hat{G}_{k+1}$. One may check that the recursions for $(\hat{A}_k, \hat{b}_k)$ are indeed based on this geometrical consideration. As a first consequence we have that $(\hat{A}_{k+1}, \hat{b}_{k+1})$ is closer to (A_0, b_0) than $(\hat{A}_k, \hat{b}_k)$ and hence the sequence of estimates is bounded. The idea of orthogonal projection is not new, it was already used in [1] and [7], and seems to go back to [8].

The algorithm is obviously recursive. Also, since $(\hat{A}_{k+1}, \hat{b}_{k+1}) = (\hat{A}_k, \hat{b}_k)$ if and only if

$x(k+1) = \hat{x}(k+1))$, neutrality is guaranteed.

Certainty equivalence is less easy to see. Only in the first loop of the algorithm we take $u(k) = f(\hat{A}_k,\hat{b}_k)x(k)$, hence we have to show that the other loop is used only a finite number of times. The formal proof will be given in Lemma 5.3, but first we will explain intuitively how $u(k)$ is calculated. Of course one would prefer to take $u(k) = f(\hat{A}_k,\hat{b}_k)x(k)$, for all k, however it is always possible that $(\hat{A}_k,\hat{b}_k)$ is non-controllable which makes it impossible to calculate $f(\hat{A}_k,\hat{b}_k)$. If we assume extra knowledge of the system, for instance if we assume that (A_0,b_0) belongs to a known convex subset of $\hat{E}$, then $(\hat{A}_k,\hat{b}_k)$ will be controllable for all k, but we want to have a global result. Another possibility is to inject external signals to force $(\hat{A}_k,\hat{b}_k)$ to converge to (A_0,b_0). In finite time $(\hat{A}_k,\hat{b}_k)$ will then be in a convex neighbourhood of (A_0,b_0) contained in E. But we don't want to add external signals all the time. What we do is the following. δ_k measures how close $(\hat{A}_k,\hat{b}_k)$ is to non-controllable. If δ_k is large, we take $u(k) = f(\hat{A}_k,\hat{b}_k)x(k)$, if δ_k is small (measured by the sequence ϵ_j) we start an alternative procedure. First we take a large input, large compared with the norm of $x(k)$. Then we apply n times the zero-input, and the distance-checking procedure starts again. We denote by τ_j the time instant on which the alternative procedure starts for the $j-th$ time. τ_1 is the first time that an estimate $(\hat{A}_k,\hat{b}_k)$ is closer to the boundary of the set of controllable pairs than ϵ_1. τ_{j+1} is the first time after τ_j+n that an estimate is closer to the boundary of the set of controllable pairs than ϵ_{j+1}. At time τ_j, the input is taken to be $C_j\|x(\tau_j)\|$, after which we apply the zero input for n time steps and the distance checking procedure starts again. The sequence τ_j constructed in this fashion can be interpreted as a sequence of stopping times (terminology borrowed from the theory of stochastic processes).

As soon as $k=\infty$ (which can take quite a long time), it is easy to see that:

$$\tau_0 = \min \{ j \mid d((\hat{A}_j,\hat{b}_j),\hat{E}^c) \le \epsilon_0 \} \tag{5.5}$$

$$\tau_k = \min \{ j \ge \tau_{k-1}+n+1 \mid d((\hat{A}_j,\hat{b}_j),\hat{E}^c) \le \epsilon_k \} \tag{5.6}$$

The minimum is understood to be infinity if the set over which the minimization takes place is empty. Suppose now that the set of finite stopping times is infinite. Then, essentially what happens is the following. Due to the growing inputs at time τ_k, the dynamics of the system (i.e. the matrix A_0), will be dominated by the input. As a consequence b_0 will be identified asymptotically. Moreover, the states $x(\tau_k+1)$ will converge to the subspace spanned by b_0. Finally, since we apply zero-inputs and because of the controllability of (A_0,b_0), the states $x(\tau_k+1),....,x(\tau_k+n)$ will asymptotically span the whole state space. That means that asymptotically we will measure the action of A_0 on the whole state space and hence A_0 will be identified too. In other words $(\hat{A}_k,\hat{b}_k)$ will converge to (A_0,b_0). But since (A_0,b_0) is controllable, it has a positive distance to the boundary of $\hat{E}$. However, the assumption that the set of finite stopping times is infinite and the fact that ϵ_j tends to zero, imply that the limit points of $(\hat{A}_{\tau_k},\hat{b}_{\tau_k})$ are non-controllable. This is a contradiction and hence the number of finite stopping times is finite. This procedure of avoiding that estimates come too close to the boundary of the set of controllable pairs is, of course, not exclusively applicable to pole assignment. It can be used for every adaptive control problem where pole-zero cancellation can occur. It proves that a search through the (A,b)-space can be done as long as one is willing to accept temporary alternative inputs. Moreover, the alternative procedure is started and switched off automatically, which is completely in the spirit of adaptive control. The inputs $u(\tau_k),....,u(\tau_{\max})$ can be interpreted as an initial excitation signal, not for identification purposes but only to avoid pole-zero cancellation of the estimates. The sequences ϵ_j, C_j can be seen as design parameters. In [11] an other procedure for biasing the estimates from non-controllable is described for stochastic systems.

We know now that after some time instant we will always use $u(k)=f(\hat{A}_k,\hat{b}_k)x(k)$. The original motivation of this control policy combined with the projection on $\hat{G}_{k+1}$ lies in the theory described in section 4. For suppose that the sequence of estimates converges to (A,b) say. Then (A,b) is an invariant point of the algorithm and hence it should satisfy:

$$A + bf(A,b)|_{\mathfrak{X}} = A_0 + b_0 f(A,b)|_{\mathfrak{X}} \qquad (5.7)$$

where $\mathfrak{X}$ is the invariant subspace spanned by the asymptotic state-trajectory. From Theorem 4.6 we can then conclude that $f(A,b)|_{\mathfrak{X}} = f(A_0,b_0)|_{\mathfrak{X}}$, and in particular that asymptotically the applied input equals the desired input.

In the next three lemmata we will derive some basic properties of the algorithm including the finiteness of the set of finite stopping times.

LEMMA 5.1 $\|(\hat{A}_k,\hat{b}_k) - (A_0,b_0)\|$ is a decreasing sequence, hence it converges to some real constant $R \geqslant 0$.

PROOF This a direct consequence of the orthogonal projection feature, which assures that $\|(\hat{A}_k,\hat{b}_k) - (A_0,b_0)\| \geqslant \|(\hat{A}_{k+1},\hat{b}_{k+1}) - (A_0,b_0)\|$.

Although Lemma 5.1 is very simple, not to say trivial, it is an important feature of our algorithm. A direct consequence of 5.1 is that $(\hat{A}_k,\hat{b}_k)$ converges to a sphere with centre (A_0,b_0) and radius R. If $R = 0$ then $(\hat{A}_k,\hat{b}_k) \rightarrow (A_0,b_0)$ and we are done. In the sequel we shall therefore assume that $R > 0$.

LEMMA 5.2 $\lim\limits_{k \to \infty} \|[(\hat{A}_{k+1},\hat{b}_{k+1}) - (\hat{A}_k,\hat{b}_k)]\| = 0$

PROOF Suppose the claim is not true. Then there exists $\epsilon > 0$ and a sequence $\{s_k\}$, such that for all k:
$\|[(\hat{A}_{s_k+1},\hat{b}_{s_k+1}) - (\hat{A}_{s_k},\hat{b}_{s_k})]\| \geqslant \epsilon$
Now denote $\|(\hat{A}_k,\hat{b}_k) - (A_0,b_0)\|$ by r_k. Choose $\delta > 0$ and let k_0 be such that $R \leqslant r_{s_k} \leqslant R + \delta$ for all $k \geqslant k_0$. Using Pythagoras' theorem we see that for all $k \geqslant k_0$:

$$r_{s_k} - r_{1+s_k} \geqslant r_{s_k} - (r_{s_k}^2 - \epsilon^2)^{1/2} \qquad (5.8)$$

$$\geqslant R - ((R+\delta)^2 - \epsilon^2)^{1/2} = C \qquad (5.9)$$

for some positive constant C and δ sufficiently small.
Since r_k is non-increasing we have $r_{s_k} - r_{s_{k+1}} \geqslant C$, which yields:

$$r_{s_k} < r_{s_{k_0}} - C(k-k_0) \leqslant R + \delta - C(k-k_0). \qquad (5.10)$$

Hence there exists k such that $r_{s_k} < R$, which is a contradiction.

LEMMA 5.3 $\{\tau_k \mid k \in \mathbb{N}, \tau_k < \infty\}$ is finite.

PROOF Suppose the contrary. Assume that $(\hat{A}_{\tau_k},\hat{b}_{\tau_k})$ converges, say $\lim\limits_{k \to \infty} (\hat{A}_{\tau_k},\hat{b}_{\tau_k}) = (A,b)$. (Otherwise take a suitable subsequence). Then, for all k:

$$d((A,b),\hat{E}_c) < \epsilon_k \qquad (5.11)$$

Hence (A,b) is non-controllable. Since for all k: $(\hat{A}_{k+1},\hat{b}_{k+1}) \in \hat{G}_{k+1}$, we have:

$$x(k+1) = \hat{A}_{k+1}x(k) + \hat{b}_{k+1}u(k) \qquad (5.12.a)$$

$$= A_0 x(k) + b_0 u(k) \qquad (5.12.b)$$

In particular:

$$\frac{x(\tau_k+1)}{\|x(\tau_k+1)\|} = \frac{\hat{A}_{\tau_k+1}x(\tau_k) + \hat{b}_{\tau_k+1}C_k\|x(\tau_k)\|}{\|\hat{A}_{\tau_k+1}x(\tau_k) + \hat{b}_{\tau_k+1}C_k\|x(\tau_k)\|\|} \qquad (5.13)$$

$$= \frac{A_0 x(\tau_k) + b_0 C_k\|x(\tau_k)\|}{\|A_0 x(\tau_k) + b_0 C_k\|x(\tau_k)\|\|} \qquad (5.14)$$

14

Now:

$$\lim_{k\to\infty} \frac{x(\tau_k+1)}{\|x(\tau_k+1)\|} = \lim_{k\to\infty} \frac{A_0 x(\tau_k)+b_0 C_k \|x(\tau_k)\|}{\|A_0 x(\tau_k)+b_0 C_k \|x(\tau_k)\|\,\|}$$

(5.15)

$$= \lim_{k\to\infty} \frac{A_0 \dfrac{x(\tau_k)}{\|x(\tau_k)\|}+b_0 C_k}{\|A_0 \dfrac{x(\tau_k)}{\|x(\tau_k)\|}+b_0 C_k\|}$$

(5.16)

$$= \lim_{k\to\infty} \frac{b_0 C_k}{\|A_0 \dfrac{x(\tau_k)}{\|x(\tau_k)\|}+b_0 C_k\|} = \frac{b_0}{\|b_0\|} = \rho b_0$$

(5.17)

for some $\rho\neq0$. On the other hand, if we take the limit in (5.13), we obtain:

$$\rho b$$

(5.18)

Hence:

$$\lim_{k\to\infty} \hat{b}_{\tau_k} = b_0$$

(5.19)

From (5.2.a.3) and (5.19) we can now conclude that:

$$\lim_{k\to\infty} \frac{x(\tau_k+i)}{\|x(\tau_k+i)\|} = \frac{A_0^{i-1}b_0}{\|A_0^{i-1}b_0\|} \qquad i=1,...,n+1$$

(5.20)

Since for all k: $x(k+1) = \hat{A}_{k+1}x(k)+\hat{b}_{k+1}u_k$, we have:

$$x(\tau_k+i+1) = \hat{A}_{\tau_k+i+1}x(\tau_k+i) = A_0 x(\tau_k+i)$$

(5.21)

$$\hat{A}_{\tau_k+i+1}\frac{x(\tau_k+i)}{\|x(\tau_k+i)\|} = \frac{A_0 x(\tau_k+i)}{\|x(\tau_k+i)\|}$$

(5.22)

Taking limits at both sides gives:

$$AA_0^{i-1}b_0 = A_0 A_0^{i-1}b_0 \qquad i=1,...,n$$

(5.23)

Since (A_0,b_0) is controllable, we conclude that $A=A_0$ and hence:

$$\lim_{k\to\infty} (\hat{A}_{\tau_k},\hat{b}_{\tau_k}) = (A_0,b_0)$$

(5.24)

And thus

$$(A,b) = (A_0,b_0)$$

(5.25)

Since by assumption (A,b) is non-controllable, we have a contradiction and the statement follows.

COROLLARY 5.4 For all k sufficiently large, $(\hat{A}_k,\hat{b}_k)$ is controllable and moreover all the limit points of $\{\hat{A}_k,\hat{b}_k\}$ are controllable.

ANALYSIS OF THE ALGORITHM.
The properties of the algorithm will be derived in several steps. First we shall state our main result.

THEOREM 5.5 Consider the (controlled) system (2.1,5.2), there exists a sequence of matrices $\{\Delta_k\}_{k\in\mathbb{N}}$, such that:

$$i) \quad x(k+1) = (A_0 + b_0 f(\hat{A}_k, \hat{b}_k)) x(k) \qquad (5.26)$$

$$= (A_0 + b_0 f(A_0, b_0) + \Delta_k) x(k) \qquad (5.27)$$

$$ii) \quad \lim_{k \to \infty} \Delta_k = 0 \qquad (5.28)$$

COMMENT. Theorem 5.5 tells us that asymptotically the action of the closed-loop matrix is identical to that of the optimal closed-loop matrix. It should be noticed that we do not claim that the real closed-loop matrix converges to the optimal one, but only as far as the action on the real state-trajectory is concerned. This weaker form of convergence is not surprising, if we realise the fact that the estimation procedure only receives information about the action of the real closed-loop matrix on the state-trajectory. We propose the term 'weak self-tuning' for this kind of behaviour. Self-tuning would have implied that $\lim_{k \to \infty} A_0 + b_0 f(\hat{A}_k, \hat{b}_k) = A_0 + B_0 f(A_0, b_0)$, which we do not claim. Note that the above result is valid whether or not Λ is contained in the unit disk. This shows that the adaptation part of the algorithm does not depend on the stability properties of the closed-loop system. The reason that the result holds even for the instable case, is that the estimation part of the algorithm depends on the direction of $x(k)$ (i.e. $\frac{x(k)}{\|x(k)\|}$), rather than $x(k)$ itself. The normalization plays an important role in the proof of Theorem 5.5. Of course for stability of the closed-loop system it is needed that Λ is contained in the unit disk.

LEMMA 5.6 $\lim_{k \to \infty} [(A_0 + b_0 f(\hat{A}_k, \hat{b}_k)) - (\hat{A}_k + \hat{b}_k f(\hat{A}_k, \hat{b}_k))] \frac{x(k)}{\|x(k)\|} = 0$

PROOF Since the algorithm 5.2 is admissible, there exists k_0 such that $u_k = f(\hat{A}_k, \hat{b}_k) x(k)$ for all $k > k_0$. Hence by definition of $\hat{G}_{k+1}$, we have for all $k > k_0$:

$$[(A_0 + b_0 f(\hat{A}_k, \hat{b}_k)) - (\hat{A}_{k+1} + \hat{b}_{k+1} f(\hat{A}_k, \hat{b}_k))] \frac{x(k)}{\|x(k)\|} = 0 \qquad (5.29)$$

Using Lemma 5.2, Corollary 5.4, and taking limits at both sides of (5.29) gives the statement.

THEOREM 5.7

i) $\lim_{k \to \infty} \|(f(\hat{A}_k, \hat{b}_k) - f(A_0, b_0)) \frac{x(k)}{\|x(k)\|} \| = 0.$

ii) $\lim_{k \to \infty} [(A_0 + b_0 f(\hat{A}_k, \hat{b}_k)) - (A_0 + b_0 f(A_0, b_0))] \frac{x(k)}{\|x(k)\|} = 0.$

PROOF i) Suppose the claim is not true. Then there exist $\epsilon > 0$ and a subsequence $\{s_k\}$, such that for all k:

$$\|(f(\hat{A}_{s_k}, \hat{b}_{s_k}) - f(A_0, b_0)) \frac{x(s_k)}{\|x(s_k)\|} \| \geq \epsilon \qquad (5.30)$$

Assume that $\{s_k\}$ was already such that:

$$\lim_{k \to \infty} (\hat{A}_{s_k}, \hat{b}_{s_k}) = (A, b) \qquad (5.31)$$

$$\lim_{k \to \infty} \frac{x(s_k)}{\|x(s_k)\|} = x^* \qquad (5.32)$$

for some $(A, b) \in \hat{E}$ and $x^* \in \mathbb{R}^n$. Then for all l:

$$\lim_{k \to \infty} (\hat{A}_{l+s_k}, \hat{b}_{l+s_k}) = (A, b) \qquad (5.33)$$

16

Define $\bar{x}_k$ and $\bar{z}_k$ by:

$$\bar{x}_k = (A + bf(A,b))^k x^*, \qquad k = 0,1,2,... \tag{5.34}$$

$$\bar{z}_k = (A_0 + b_0 f(A,b))^k x^*, \qquad k = 0,1,2,... \tag{5.35}$$

Then, by Lemma 5.6 and (5.33), we have for all k:

$$\bar{x}_k = \bar{z}_k \tag{5.36}$$

Hence by Theorem 4.4 it follows that in particular:

$$f(A,b)x^* = f(A_0,b_0)x^* \tag{5.37}$$

This contradicts (5.30), and the statement follows.
ii) This follows from Lemma 5.6 and (i).

PROOF of Theorem 5.5:

This is now a direct application of Theorem 5.7 and Theorem 2.1.

REMARK In [13,14] Theorem 5.5 was proven slightly different. There the notion of excitation space was introduced. This is the space spanned by the limit points of $\{\frac{x(k)}{\|x(k)\|}\}_{k \in \mathbb{N}}$. Denote this space by $\mathfrak{X}$. From Theorem 5.7 we can then derive that:

$$\lim_{k \to \infty} [f(\hat{A}_k, \hat{b}_k) - f(A_0,b_0)]x = 0 \tag{5.38}$$

for all $x \in \mathfrak{X}$, which implies that only on the asymptotic active part of the state space the action of $f(\hat{A}_k, \hat{b}_k)$ is as desired. This illustrates the term weak self-tuning. Also it can be proven that $\mathfrak{X}$ is invariant under $A_0 + b_0 f(A_0,b_0)$.

6. AN ADMISSIBLE ALGORITHM FOR THE UNOBSERVED STATE CASE

We will now propose an adaptive pole assignment algorithm for the class of single-input/single-output discrete time systems of (known) order n. The algorithm is based on ideas developed in the previous section. There it was assumed that the state of the system was observed. This assumption is now relaxed, and hence the algorithm should also contain an observer part. Indeed, the algorithm consists of an estimation part including an adaptive observer and a control part. However, one can also view the estimation part as an adaptive partial realization procedure, since no attempt is made to identify the system parameters completely, and the true state trajectory (whatever that may be) is not reconstructed either. What we really end up with is not a complete realization of the unknown system, but an input/state/output description that is suitable for one input/output sequence, namely the asymptotic one. This section runs very much parallel to section 5 and the reader is referred to that section for some of the details and discussions.
We will first give three different descriptions of the system. Then we will introduce the algorithm. Then we will prove some basic properties. Next we will formulate the main theorem of this paper: the characterization of the asymptotic closed-loop behaviour of the controlled system. Finally, we will give the analysis of the algorithm, ultimately leading to the proof of the main theorem.

THE SYSTEM:
The true system is supposed to be linear, time-invariant, single-input, single-output and of known order n. Hence it has an input/output description of the form:

$$y(k+1) = a_0^0 y(k) + ... + a_{n-1}^0 y(k-n+1) + b_0^0 u(k) + ... + b_{n-1}^0 u(k-n+1) \tag{6.1}$$

Since we want to work in i/s/o form, we realize (6.1) as follows: Define

$(A_0, b_0, c) \in \mathbb{R}^{n \times n} \times \mathbb{R}^{n \times 1} \times \mathbb{R}^{1 \times n}$, by:

$$A_0 := \begin{bmatrix} a_0^0 & 1 & . & . & . & 0 \\ : & 0 & & & & : \\ : & : & & & & 0 \\ : & : & & & & 1 \\ a_{n-1}^0 & 0 & . & . & . & 0 \end{bmatrix} \qquad b_0 := \begin{bmatrix} b_0^0 \\ : \\ : \\ : \\ b_{n-1}^0 \end{bmatrix} \qquad c := [1 \ 0 \ ... 0] \tag{6.2}$$

And define for every k, $x(k) \in \mathbb{R}^n$, by:

$$x_1(k) = y(k) \tag{6.3.a}$$

$$x_i(k) = \sum_{j=0}^{n-i} a_{i+j-1}^0 y(k-1-j) + \sum_{j=0}^{n-i} b_{i+j-1}^0 u(k-1-j) \qquad i = 2,...,n \tag{6.3.b}$$

Then for all k:

$$x(k+1) = A_0 x(k) + b_0 u(k) \tag{6.4.a}$$

$$y(k) = cx(k) \tag{6.4.b}$$

Although there are of course many other realizations of 6.1, we will refer to 6.4 as the true realization, and to the sequence $\{x(k)\}_{k \in \mathbb{N}}$ as the true state trajectory.

We will also need the following non-minimal realization of 6.1. Define $F_0 \in \mathbb{R}^{(2n-1) \times (2n-1)}$ and $g_0 \in \mathbb{R}^{(2n-1) \times 1}$ by:

$$F_0 := \begin{bmatrix} a_0^0 & . & a_{n-2}^0 & a_{n-1}^0 & b_1^0 & . & b_{n-2}^0 & b_{n-1}^0 \\ 1 & & 0 & 0 & 0 & & 0 & 0 \\ 0 & & : & : & : & & : & : \\ : & & 0 & : & : & & : & : \\ : & & 1 & : & : & & : & : \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ : & & 0 & : & 1 & & : & : \\ : & & : & : & 0 & & : & : \\ : & & : & : & & 0 & & : \\ 0 & . & 0 & 0 & 0 & . & 1 & 0 \end{bmatrix} \qquad g_0 := \begin{bmatrix} b_0^0 \\ 0 \\ : \\ : \\ 0 \\ 1 \\ 0 \\ : \\ : \\ 0 \end{bmatrix} \tag{6.5}$$

Define:

$$\phi(k) := [y(k), y(k-1),...,y(k-n+1), u(k-1),...,u(k-n+1)]^T \tag{6.6}$$

then for all k:

$$\phi(k+1) = F_0 \phi(k) + g_0 u(k) \tag{6.7}$$

Moreover, since (A_0, b_0) is controllable, so is (F_0, g_0). Finally define $M \in \mathbb{R}^{(2n-1) \times n}$ by:

$$M := \begin{bmatrix} 1 & 0 & 0 & . & . & 0 & 0 & 0 & . & . & 0 \\ 0 & a_1 & a_2 & & & a_{n-1} & b_1 & b_2 & & & b_{n-1} \\ : & a_2 & a_3 & & . & 0 & b_2 & b_3 & & . & 0 \\ : & : & : & . & & : & : & : & . & & : \\ : & : & a_{n-1} & & & : & : & b_{n-1} & & & : \\ 0 & a_{n-1} & 0 & . & . & 0 & b_{n-1} & 0 & . & . & 0 \end{bmatrix} \tag{6.8}$$

then for all k we have by (6.3) that:

$$x(k) = M\phi(k) \tag{6.9}$$

THE ALGORITHM 6.11

Choose any sequence $\{\epsilon_k\}$ and any sequence $\{C_k\}$ such that:

$$\epsilon_k \downarrow 0 \quad \text{and} \quad C_k \uparrow \infty \tag{6.10}$$

INITIALIZATION $(\hat{A}_0, \hat{b}_0)$: arbitrarily, $h_0 = 0$, $j_0 = 0$, $\phi(0)$: given.

RECURSION

$$\hat{A}_k := \begin{bmatrix} \hat{a}_0(k) & 1 & . & . & . & 0 \\ : & 0 & & & & : \\ : & : & & & & 0 \\ : & : & & & & 1 \\ \hat{a}_{n-1}(k) & 0 & . & . & . & 0 \end{bmatrix} \quad \hat{b}_k := \begin{bmatrix} \hat{b}_0(k) \\ : \\ : \\ : \\ \hat{b}_{n-1}(k) \end{bmatrix} \tag{6.11.A}$$

$$\delta_k = d^*((\hat{A}_k, \hat{b}_k), \hat{E}_k^c) \tag{6.11.B}$$

if $h_k = 0$

   then:

      if $\delta_k \geq \epsilon_{j_k}$

      then:

$$\{ \; u(k) = f(\hat{A}_k, \hat{b}_k)\hat{z}(k) \tag{6.11.a.1}$$

$$h_{k+1} = h_k \tag{6.11.b.1}$$

$$j_{k+1} = j_k \; \} \tag{6.11.d.1}$$

      else (if $\delta_k < \epsilon_{j_k}$)

$$\{ \; u(k) = C_{j_k} \|\phi(k)\| \tag{6.11.a.2}$$

$$h_{k+1} = 2n - 1 \tag{6.11.b.2}$$

$$\tau_{j_k} = k \tag{6.11.c.2}$$

$$j_{k+1} = j_k + 1 \; \}\} \tag{6.11.d.2}$$

else (if $h_k > 0$)

$$\{ \; u(k) = 0 \tag{6.11.a.3}$$

$$h_{k+1} = h_k - 1 \tag{6.11.b.3}$$

$$j_{k+1} = j_k \; \} \tag{6.11.d.3}$$

$$\{ \; y(k+1) = a_0^0 y(k) + \ldots + a_{n-1}^0 y(k-n+1) + b_0^0 u(k) + \ldots + b_{n-1}^0 u(k-n+1) \tag{6.11.e}$$

$$\hat{y}(k+1) = \hat{a}_0(k)y(k) + \ldots + \hat{a}_{n-1}(k)y(k-n+1) + \hat{b}_0(k)u(k) + \ldots + \hat{b}_{n-1}(k)u(k-n+1) \tag{6.11.f}$$

$*$ d means Euclidean distance. As proven in [2], δ_k is the smallest singular value of the controllability matrix of $(\hat{A}_k, \hat{b}_k)$.

$$\lambda(k) = (y^2(k)+...+y^2(k-n+1)+u^2(k)+...+u^2(k+n-1))^{-1}[y(k+1)-\hat{y}(k+1)] \qquad (6.11.g)$$

$$\hat{a}_i(k+1) = \hat{a}_i(k)+\lambda(k)y(k-i) \qquad i=0,...,n-1 \qquad (6.11.h)$$

$$\hat{b}_i(k+1) = \hat{b}_i(k)+\lambda(k)u(k-i) \qquad i=0,...,n-1 \qquad (6.11.i)$$

$$\hat{x}_i(k) = \sum_{j=0}^{n-i}\hat{a}_{i+j-1}(k+1)y(k-1-j)+\sum_{j=0}^{n-i}\hat{b}_{i+j-1}(k+1)u(k-1-j) \quad i=1,...,n \qquad (6.11.j)$$

$$\hat{z}_i(k+1) = \sum_{j=0}^{n-i}\hat{a}_{i+j-1}(k+1)y(k-j)+\sum_{j=0}^{n-i}\hat{b}_{i+j-1}(k+1)u(k-j) \qquad i=1,...,n \ \Big\} \qquad (6.11.k)$$

REMARK The division in (6.11.g) and can of course be done only if $\phi(k)$ or $u(k)$ is non-zero. Therefore if $\phi(k_0) = 0$ for some k_0, we do not change the estimates anymore and we take $u(k) = 0$ for all $k \geq k_0$. For the analysis of the algorithm we will assume that $\phi(k) \neq 0$ for all k. This assumption is also needed if we want to apply Theorem 2.1.

COMMENT The interpretation of the algorithm is more or less the same as for the observed state case, the main difference being that now a state trajectory has to be invented too. Define:

$$\hat{G}_{k+1} := \{(a_0,..,a_{n-1},b_0,..,b_{n-1}) \,|\, y(k+1) = \sum_{i=0}^{n-1}(a_iy(k-i)+b_iu(k-i))\} \qquad (6.12)$$

Then $\hat{G}_{k+1}$ is linear affine and $(a_0^0,...,a_{n-1}^0,b_0^0,...,b_{n-1}^0)\in\hat{G}_{k+1}$. Define $(\hat{a}_0(k+1),...,\hat{a}_{n-1}(k+1),\hat{b}_0(k+1),...,\hat{b}_{n-1}(k+1))$ as the orthogonal projection of $(\hat{a}_0(k),...,\hat{a}_{n-1}(k),\hat{b}_0(k),...,\hat{b}_{n-1}(k))$ on $\hat{G}_{k+1}$.

THE OBSERVER

The definition of $\hat{x}(k)$ and $\hat{z}(k+1)$ is motivated by the following analysis.
At time k we compute $(\hat{a}_0(k+1),...,\hat{a}_{n-1}(k+1),\hat{b}_0(k+1),...,\hat{b}_{n-1}(k+1))$ on the basis of the observed data $(u(k-n+1),y(k-n+1),...,u(k),y(k),y(k+1))$. Suppose we want to have an i/s/o description of this finite i/o sequence:

$$\bar{x}(j+1) = \hat{A}(k+1)\bar{x}(j)+\hat{b}(k+1)u(j) \qquad j=k-n+1,...,k \qquad (6.13.a)$$

$$y(j) = c\bar{x}(j) \qquad (6.13.b)$$

Let $\bar{x}$ be the unique solution of the equations:

$$c\hat{A}_{k+1}^{j-1}\bar{x} + c\sum_{l=0}^{j-2}\hat{A}_{k+1}^{l}\hat{b}_{k+1}u(k-n+j-1-l) = y(k-n+j) \qquad j=1,...,n \qquad (6.14)$$

Then, if $(\hat{A}_{k+1},\hat{b}_{k+1})$ was the true parameter, $\bar{x}$ would have been the estimation of $x(k-n+1)$ based on the dead-beat observer for (A_0,b_0). Since we use $(\hat{A}_{k+1},\hat{b}_{k+1})$ instead of (A_0,b_0), the observer part of the algorithm can be interpreted as a certainty equivalence dead-beat observer. Now one may check that:

$$\hat{x}(k) = \hat{A}_{k+1}^{n-1}\bar{x} + \sum_{l=0}^{n-2}\hat{A}_{k+1}^{l}\hat{b}_{k+1}u(k-n+j-1-l) \qquad (6.15)$$

$$\hat{z}(k+1) = \hat{A}_{k+1}\hat{x}(k)+\hat{b}_{k+1}u(k) \qquad (6.16)$$

where $\hat{x}(k)$ and $\hat{z}(k+1)$ are defined by (6.11.j) and (6.11.k).
The next three lemmata give some essential properties of the algorithm.

LEMMA 6.1 $\|(\hat{A}_k,\hat{b}_k)-(A_0,b_0)\|$ is a decreasing sequence, hence it converges to some real constant $R \geq 0$.

PROOF See the proof of Lemma 5.1.

LEMMA 6.2 $\lim_{k\to\infty} \|[(\hat{A}_{k+1},\hat{b}_{k+1})-(\hat{A}_k,\hat{b}_k)]\| = 0$

PROOF See the proof of Lemma 5.2.

LEMMA 6.3 $\{\tau_k \mid k \in \mathbb{N}, \tau_k < \infty\}$ is finite.

PROOF Suppose the contrary. Assume that $(\hat{A}_{\tau_k}, \hat{b}_{\tau_k})$ converges, say $\lim_{k \to \infty} (\hat{A}_{\tau_k}, \hat{b}_{\tau_k}) = (A, b)$. (Otherwise take a suitable subsequence). Then (A, b) is non-controllable. Define $(\hat{F}_k, \hat{g}_k) \in \mathbb{R}^{(2n-1) \times (2n-1)} \times \mathbb{R}^{(2n-1) \times 1}$ by replacing a_i^0 by $\hat{a}_i(k)$ and b_i^0 by $\hat{b}_i(k)$ in (6.2). Now the proof is completely analogous to the proof of Lemma 5.3.

COROLLARY 6.4 For all k sufficiently large, $(\hat{A}_k, \hat{b}_k)$ is controllable and moreover all the limit points of $\{\hat{A}_k, \hat{b}_k\}$ are controllable.
The analysis of (6.11) is as follows: first we will prove that the sequence $\{\hat{z}(k+1)\}$ provides asymptotically a realization of the controlled system. Then we will apply Theorem 4.7 to the limiting behaviour of the system to connect the true state trajectory with the constructed one.

ANALYSIS OF THE ALGORITHM

THEOREM 6.5 Consider the (controlled) system (2.1,6.11). Assume that there exists $\lambda \in \Lambda$ such that $\lambda \neq 0$. Then there exists a sequence of matrices $\{\Delta_k\}_{k \in \mathbb{N}}$, such that for k sufficiently large:

$$i) \quad x(k+1) = A_0 x(k) + b_0 f(\hat{A}_k, \hat{b}_k) \hat{z}(k) \tag{6.17}$$

$$= (A_0 + b_0 f(A_0, b_0) + \Delta_k) x(k)) \tag{6.18}$$

$$ii) \quad \lim_{k \to \infty} \Delta_k = 0 \tag{6.19}$$

REMARK Just as in Theorem 5.5 we do not claim that (A_0, b_0) is identified, nor is $f(A_0, b_0)$ identified. Even the state trajectory is not reconstructed. The constructed state trajectory $\hat{z}(k)$ will in general not equal $x(k)$, nor will $f(\hat{A}_k, \hat{b}_k)$ be close to $f(A_0, b_0)$. In the limit, both $f(\hat{A}_k, \hat{b}_k)$ and $\hat{z}(k)$ will be wrong, but the resulting input sequence $u(k) = f(\hat{A}_k, \hat{b}_k) \hat{z}(k)$ will be as desired, and that is what really matters. Again this could be seen as a weak form of self-tuning. As in section 5, the above result is valid whether or not Λ is contained in the unit disk. This shows that the adaptation part of the algorithm does not depend on the stability properties of the closed-loop system. The reason that the result holds even for the instable case, is that the estimation part of the algorithm depends on the direction of $\phi(k)$ (i.e. $\frac{\phi(k)}{\|\phi(k)\|}$), rather than $\phi(k)$ itself. The normalization plays an important role in the proof of Theorem 6.5. Of course, for stability of the closed-loop system it is needed that Λ is contained in the unit disk.
Note that:

$$\hat{x}(k) = \hat{M}_k \phi(k) \tag{6.20}$$

$$\hat{z}(k) = \hat{M}_{k-1} \phi(k) \tag{6.21}$$

where the matrices $\hat{M}_k \in \mathbb{R}^{n \times (2n-1)}$ are defined by replacing a_i^0 by $\hat{a}_i(k)$ and b_i^0 by $\hat{b}_i(k)$ in (6.8). Define:

$$d_k = \|\phi(k)\| = [\sum_{j=0}^{n-1} y^2(k-1) + \sum_{j=1}^{n-1} u^2(k-1)]^{\frac{1}{2}} \tag{6.22}$$

LEMMA 6.6

$$\lim_{k \to \infty} \|\frac{1}{d_k} [\hat{z}(k+1) - (\hat{A}_k + \hat{b}_k f(\hat{A}_k, \hat{b}_k)) \hat{z}(k)]\| = 0 \tag{6.23}$$

PROOF From Lemma 6.3 and (6.11.a.1) we deduce that for k sufficiently large:

$$\hat{z}(k+1) = \hat{A}_{k+1}\hat{x}(k) + \hat{b}_{k+1}f(\hat{A}_k,\hat{b}_k)\hat{z}(k) \tag{6.24}$$

Hence:

$$\hat{z}(k+1) = (\hat{A}_{k+1}\hat{M}_k + \hat{b}_{k+1}f(\hat{A}_k,\hat{b}_k)\hat{M}_{k-1})\phi(k) \tag{6.25}$$

Now, since $\|(\hat{A}_k,\hat{b}_k) - (\hat{A}_{k+1},\hat{b}_{k+1})\| \to 0$, the continuity of f on E, the controllability of the limit points of $(\hat{A}_k,\hat{b}_k)$, we conclude:

$$\lim_{k\to\infty} \|\frac{1}{d_k}[\hat{z}(k+1) - (\hat{A}_{k+1} + \hat{b}_{k+1}f(\hat{A}_k,\hat{b}_k))\hat{z}(k)]\| \tag{6.26}$$

$$= \lim_{k\to\infty} \|(\hat{A}_{k+1}\hat{M}_k + \hat{b}_{k+1}f(\hat{A}_k,\hat{b}_k)\hat{M}_{k-1})\frac{\phi(k)}{\|\phi(k)\|} - (\hat{A}_{k+1}\hat{M}_{k-1} + \hat{b}_{k+1}f(\hat{A}_k,\hat{b}_k)\hat{M}_{k-1}\frac{\phi(k)}{\|\phi(k)\|}\| \tag{6.27}$$

THEOREM 6.7

$$\lim_{k\to\infty} \|\frac{f(A_0,b_0)x(k) - f(\hat{A}_k,\hat{b}_k)\hat{z}(k)}{d_k}\| = 0 \tag{6.28}$$

PROOF Suppose the claim is not true. Then there exist $\epsilon > 0$ and a sequence $\{t_k\}$ such that for all k:

$$\|\frac{f(A_0,b_0)x(k) - f(\hat{A}_k,\hat{b}_k)\hat{z}(k)}{d_k}\| \geq \epsilon \tag{6.29}$$

Let x^* be a limit point of $\frac{x(t_k)}{d_{t_k}}$. Say $\lim_{k\to\infty} \frac{x(t_k^{(1)})}{d_{t_k^{(1)}}} = x^*$, for some subsequence $\{t_k^{(1)}\}$ of $\{t_k\}$. Let $\{t_k^{(2)}\}$ be a subsequence of $\{t_k^{(1)}\}$, such that:

$$\lim_{k\to\infty} (\hat{A}_{t_k^{(2)}},\hat{b}_{t_k^{(2)}}) = (A,b) \tag{6.30}$$

and:

$$\lim_{k\to\infty} \frac{z(t_k^{(2)})}{d_{t_k^{(2)}}} = z^* \tag{6.31}$$

Then for all l:

$$\lim_{k\to\infty} (\hat{A}_{l+t_k^{(2)}},\hat{b}_{l+t_k^{(2)}}) = (A,b) \tag{6.32}$$

Define sequences $\{x(k)^*\}$ and $\{z(k)^*\}$ as follows:

$$x(0)^* := x^* \qquad z(0)^* := z^* \tag{6.33}$$

$$z(k)^* = (A + bf(A,b))^k z^* \tag{6.34}$$

$$x(k+1)^* = A_0 x(k)^* + b_0 f(A,b)z(k)^* \tag{6.35}$$

Then, by Lemma 6.6, the fact that $c\hat{z}(k) = cx(k)$, and (6.32), for all k:

$$cx(k)^* = cz(k)^* \tag{6.36}$$

hence by Theorem 4.7 there exists a non-singular matrix S such that for all k:

$$Sz(k)^* = x(k)^* \tag{6.37}$$

Hence, by Theorem 4.5:

$$f(A,b)z^* = f(A_0,b_0)x^*$$
(6.38)

which contradicts (6.29).

REMARK Theorem 6.7 tells us that asymptotically the applied input equals the desired input if we normalize with the norm of $\phi(k)$. For the derivation of the same result, but this time normalized with the norm of $x(k)$ (in order to be able to apply Theorem 2.1), we will study, as an intermediate step, the non-minimal realization (6.7) of (6.1).

We have for all k:

$$\phi(k+1) = F_0\phi(k)+g_0u(k)$$
(6.39)

Define $h_0 \in \mathbb{R}^{1\times(2n-1)}$ by:

$$h_0 := f(A_0,b_0)M$$
(6.40)

Then the desired input is:

$$u(k) = f(A_0,b_0)x(k) = f(A_0,b_0)M\phi(k) = h_0\phi(k)$$
(6.41)

Hence the desired closed-loop representation of (6.39) is given by:

$$\phi(k+1) = (F_0+g_0h_0)\phi(k)$$
(6.42)

However, the applied control is:

$$u(k) = f(\hat{A}_k,\hat{b}_k)\hat{z}(k) = f(\hat{A}_k,\hat{b}_k)\hat{M}_{k-1}\phi(k)$$
(6.43)

The following Theorem characterizes the asymptotic closed-loop behaviour of the realization (6.11,6.39).

THEOREM 6.8 There exists a sequence of matrices $\{\Delta'_k\}_{k\in\mathbb{N}}$ such that:

(i) $\phi(k+1) = (F_0+g_0f(\hat{A}_k,\hat{b}_k)\hat{M}_{k-1})\phi(k)$
(6.44)

$= (F_0+g_0h_0+\Delta'_k)\phi(k)$
(6.45)

(ii) $\lim\limits_{k\to\infty} \Delta'_k = 0$
(6.46)

PROOF The proof follows immediately from Theorem 6.7 and Theorem 2.1

LEMMA 6.9 Let $\{M_k\}_{k\in\mathbb{N}}$ be a sequence of matrices in $\mathbb{R}^{n\times n}$, such that $\lim\limits_{k\to\infty} M_k = M \neq 0$. Choose $x(0)\in\mathbb{R}^{n\times n}$ and define:

$$x(k+1) = M_k x(k)$$
(6.47)

Assume that for all k $x(k) \neq 0$. Denote by $\mathfrak{X}$ the linear span of the limit points of $\dfrac{x(k)}{\|x(k)\|}$. Then:

$i)$ $M\mathfrak{X} \subset \mathfrak{X}$
(6.48)

$ii)$ for all $x\in\mathfrak{X}, x\neq 0 : Mx \neq 0$
(6.49)

PROOF i) Suppose x^* is a limit point of $\{\dfrac{x(k)}{\|x(k)\|}\}$. Say $\lim\limits_{k\to\infty} \dfrac{x(s_k)}{\|x(s_k)\|} = x^*$, for some subsequence $\{s_k\}$. Then:

$$Mx^* = \lim\limits_{k\to\infty} \frac{1}{\|x(s_k)\|}M_{s_k}x(s_k) = \lim\limits_{k\to\infty} \frac{1}{\|x(s_k)\|}x(1+s_k)$$
(6.50)

$$= \lim_{k \to \infty} \frac{\|x(1+s_k)\|}{\|x(s_k)\|} \frac{x(1+s_k)}{\|x(1+s_k)\|} = \lim_{k \to \infty} \frac{\|M_{s_k}x(s_k)\|}{\|x(s_k)\|} \frac{x(1+s_k)}{\|x(1+s_k)\|} \tag{6.51}$$

$$= \|M_{s_k} \frac{x(s_k)}{\|x(s_k)\|}\| \frac{x(1+s_k)}{\|x(1+s_k)\|} = \|Mx^*\| \lim_{k \to \infty} \frac{x(1+s_k)}{\|x(1+s_k)\|} \tag{6.52}$$

Hence $Mx^* \in X$. By linearity the result follows.

ii) After a change of basis, M and M_k can be decomposed as:

$$M = \begin{bmatrix} M_{11} & 0 \\ 0 & 0 \end{bmatrix}, \quad M_k = \begin{bmatrix} M_{11}(k) & M_{12}(k) \\ M_{21}(k) & M_{22}(k) \end{bmatrix} \tag{6.53}$$

such that M_{11} is non-singular. Then:

$$\lim_{k \to \infty} \frac{x_2(k+1)}{\|x(k)\|} = \lim_{k \to \infty} \frac{(M_{21}(k)x_1(k) + M_{22}(k)x_2(k))}{\|x(k)\|} = 0 \tag{6.54}$$

$$\lim_{k \to \infty} \frac{x_1(k+1)}{\|x(k)\|} = \lim_{k \to \infty} \frac{(M_{11}(k)x_1(k) + M_{12}(k)x_2(k))}{\|x(k)\|} \neq 0 \tag{6.55}$$

This yields:

$$\frac{x(k)}{\|x(k)\|} \to \text{coker}(M) \tag{6.56}$$

By linearity the statement follows.

COROLLARY 6.10 Let x^* be a limit point of $\frac{x(k)}{d_k}$, then: $x^* \neq 0$.

PROOF One may check that $\Lambda \subset \sigma(G_0)$. Since by assumption at least one of the $\lambda's$ is non zero, we conclude by Lemma 6.9 that there exists $\mu > 0$ such that for i:

$$\frac{\phi(k+i)}{d_k} \geq \mu^i \tag{6.57}$$

Now suppose $x^* = 0$. Say $\frac{x(t_k)}{d_{t_k}} \to 0$, then:

$$\frac{\|x(t_k+1)\|}{d_{t_k}} = \frac{\|A_0 x(k) + b_0 f(\hat{A}_{t_k}, \hat{b}_{t_k})\hat{z}(k)\|}{d_{t_k}} \tag{6.58}$$

$$= \frac{\|(A_0 + b_0 f(A_0, b_0))x(t_k) + b_0(f(\hat{A}_{t_k}, \hat{b}_{t_k})\hat{z}(t_k) - f(A_0, b_0)x(t_k))\|}{d_{t_k}} \to 0 \tag{6.59}$$

This implies:

$$\frac{(y(t_k), y(t_k+1), .., y(t_k+n-1), u(t_k), u(t_k+1), .., u(t_k+n-1))}{d_{t_k}} \to 0 \tag{6.60}$$

$$\Rightarrow \frac{\phi(t_k+n-1)}{d_{t_k}} \to 0 \tag{6.61}$$

which contradicts (6.57).

COROLLARY 6.11

$$\lim_{k \to \infty} \|\frac{(u(k) - f(A_0, b_0)x(k))}{\|x(k)\|}\| = 0 \tag{6.62}$$

PROOF Since by the previous corollary $\dfrac{d_k}{\|x(k)\|} \leqslant \delta$, for some $\delta > 0$, we have:

$$\lim_{k \to \infty} \|\frac{(u(k) - f(A_0, b_0)x(k))}{\|x(k)\|}\| = \lim_{k \to \infty} \|\frac{(u(k) - f(A_0, b_0)x(k))}{d_k} \frac{d_k}{\|x(k)\|}\| = 0 \qquad (6.63)$$

PROOF OF THEOREM 6.5

The proof of Theorem 6.5 is now just an application of Theorem 2.1 and Corollary 6.11.

7. SIMULATIONS

The algorithms presented in sections 5 & 6 have been simulated extensively. As could be expected, convergence gets slower as the order of the system increases. The asymptotic behaviour of the controlled system was characterized in terms of the action of the asymptotic closed-loop matrix on the state-trajectory. The weak self-tuning property, however, can better be illustrated by a comparison between the applied input and the desired input. In many cases the assumption (2.6) in Theorem 2.1 implies that:

$$\lim_{k \to \infty} \frac{u(k)}{fx(k)} = 1 \qquad (7.1)$$

We will now give the graphs of the output of a second order unstable non-minimum phase system and of $\dfrac{u(k)^a}{u(k)^d}$ (where superscripts a and d stands for "applied" and "desired" respectively). The applied algorithm is the one introduced in section 6 (non-observed state case). The true system has the realization:

$$x(k+1) = \begin{bmatrix} -1 & 1 \\ 6 & 0 \end{bmatrix} x(k) + \begin{bmatrix} 1 \\ 2 \end{bmatrix} u(k) \qquad (7.2.a)$$

$$y(k) = [1 \ 0]x(k) \qquad (7.2.b)$$

The system was initially guessed as:

$$\hat{A}_0 = \begin{bmatrix} -3 & 1 \\ 7 & 0 \end{bmatrix} \quad \hat{b}_0 = \begin{bmatrix} 2 \\ 3 \end{bmatrix} \qquad (7.3)$$

The desired closed-loop characteristic polynomial was chosen to be:

$$X^2 - 1.7X + 0.72 \qquad (7.4)$$

If we look at figures 7.1 and 7.2, we see that initially the system behaves badly. Then after a certain learning period, the quotient of applied and desired input is close to 1 and the system begins to stabilize. One may check that $y(k)$ tends to zero exponentially fast, the exponent tending to the slowest desired pole: 0.9. At iteration 32 there is a peak in the graph of $\dfrac{u(k)^a}{u(k)^d}$. The explanation is that the state was suddenly too far away from the subspace it was converging to. Outside this subspace the control law was still far away from the desired one. Although after 150 iterations $\dfrac{u(k)^a}{u(k)^d}$ is very close to 1, the true system has not been identified at all. The simulations showed that the Euclidean distance between the true system parameters and $(\hat{A}_{150}, \hat{b}_{150})$ was almost 2.24 (initially 2.65). After 150 iterations the closed-loop poles were 0.89998 and 0.81490. The larger deviation of the second pole

from the desired one can be explained by the fact that the state-trajectory converges to the invariant subspace belonging to the other pole. Hence information about the invariant subspace belonging to the second pole gets poor very quickly. This illustrates the weak self-tuning feature: only the poles which are excited asymptotically are placed properly.
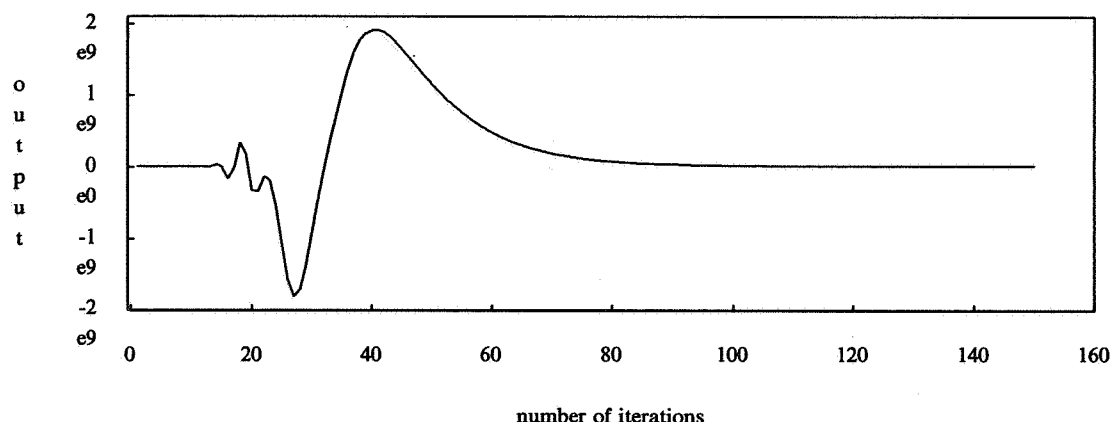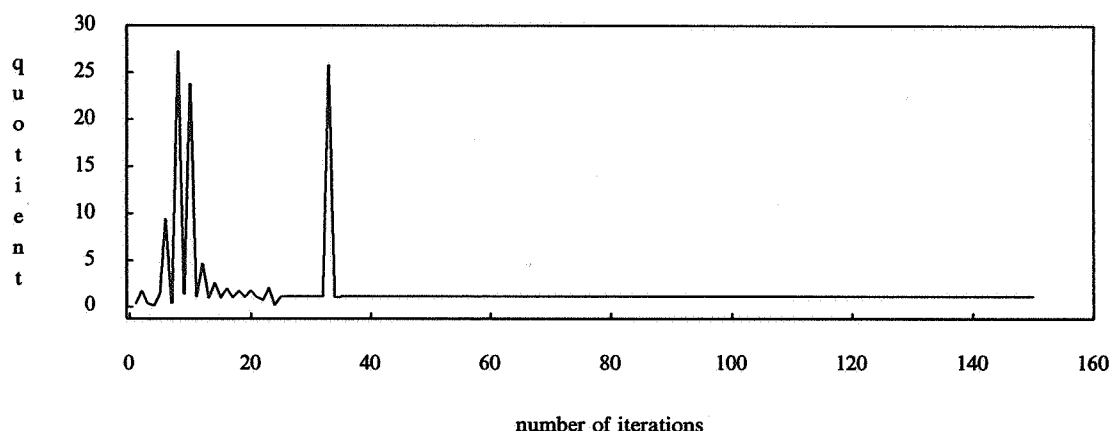

Figure 7.1. The output of the system.

Figure 7.2. The quotient of applied and desired input.

8. CONCLUSIONS AND REMARKS

Algorithms have been proposed for adaptive pole-assignment for single-input/single-output systems. Pole-zero cancellation of the estimates was avoided by introducing a finite number of special inputs. A weak form of self-tuning was derived without imposing conditions on the richness of the signals.

There are some interesting topics for further research. First of all the limit points of the estimates. We did not claim convergence of the sequence of estimates since we did not need it. It would, however, be interesting to know whether or not the estimates converge. If they do, it is obvious that the limit will be an invariant point of the algorithm, and will hence satisfy (5.7). For the first order case convergence of the estimates can be proven.

Another question is to which invariant subspace of $A_0 + b_0 f(A_0, b_0)$ the state trajectory converges. From simulations we know that in most cases the state trajectory converges to the invariant subspace belonging to the slowest desired pole, as in the non-adaptive case. It is not unlikely that this will

26

generically be the case, but a proof has not yet been found.

The algorithms as presented have an interesting potential possibility. The asymptotic results have been derived without imposing any conditions on excitation. In the initial period inputs are calculated on the basis of very little knowledge of the system, and hence they can be expected to be far from appropriate. This means that initially certainty equivalence does not make so much sense, and hence the input sequence could also contain an active learning part. This learning should then be tempered more and more as the tuning gets better and better. Since the results presented here do not depend on asymptotic excitation, this should in principle not change the asymptotic properties of the algorithm, but only the initial part.

Finally, the algorithm should be translated into the continuous time case.

We hope to report on these topics in the near future.

REFERENCES

1. A. BECKER, P.R. KUMAR, C.Z. WEI (1985). Adaptive control with the stochastic approximation algorithm: geometry and convergence. *IEEE Trans. Aut. Contr. AC-30*, 330-338.
2. R. EISING (1984). Between Controllable and Uncontrollable. *Systems & Control Letters 4*, 263-264.
3. H. ELLIOTT, R. CRISTI AND M. DAS (1985). Global Stability of Adaptive Pole Placement Algorithms. *IEEE Trans. Aut. Control. AC-30*, 348-356.
4. H. ELLIOTT AND W.A. WOLOVICH (1979). Parameter Adaptive Identification and Control. *IEEE Trans. Aut. Control. AC-24*, 592-599.
5. A. FEUER (1986). A parametrization for Model Reference Adaptive Pole Placement. *IEEE Trans. Aut. Control. AC-31*, 782-785.
6. G.C. GOODWIN AND K.S. SIN (1984). *Adaptive Filtering Prediction and Control.* Englewood Cliffs, NJ: Prentice Hall.
7. G.C. GOODWIN AND K.S. SIN (1981). Adaptive Control of Nonminimum Phase Systems. *IEEE Trans. Aut. Contr. AC-26*, 478-483.
8. S. KARZMARZ (1937). Angenäherte Auflösung von Systemen linearer Gleichungen. *Bull. Int. Acad. Polon. Sci. Mat. Nat. Ser A*, 355-.
9. G. KREISSELMEIER (1980). Adaptive Control via Adaptive Observation and Asymptotic Feedback Matrix Synthesis. *IEEE Trans. Aut. Contr. AC-25*, 717-722.
10. G. KREISSELMEIER AND M.C. SMITH (1986). Stable Adaptive Regulation of Arbitrary nth-Order Plants. *IEEE Trans. Aut. Contr. AC-31*, 299-305.
11. J.B. MOORE (1986). Central Tendency Pole Assignment. *Proceedings of the 25th IEEE Conference on Decision and Control*, Athens, Greece. 100-105.
12. R. LOZANO-LEAL AND G.C. GOODWIN (1985). A Globally Convergent Adaptive Pole Placement Algorithm Without a Persistency of Excitation Requirement. *IEEE Trans. Aut. Contr. AC-30*, 795-798.
13. J.W. POLDERMAN (1986). Adaptive Pole Assignment by State Feedback (short version). *Proceedings of the 25th IEEE Conference on Decision and Control*, Athens, Greece. 106-108.
14. J.W. POLDERMAN (1986). Adaptive Pole Assignment by State Feedback. To appear in: *Proceedings of the Workshop on Modelling and Adaptive Control.* 1986, July 7-15, Sopron Hungary.
15. J.W. POLDERMAN (1986). On the necessity of identifying the true system in adaptive LQ control, *Systems & Control Letters 8*, 87-91.
16. S.S. SASTRY (1984). Model-Reference Adaptive Control-Stability,Parameter Convergence, and Robustness. *IMA Journal of Mathematical Control & Information 1*, 27-66.
17. W.M. WONHAM (1979). *Linear Multivariable Control: a Geometric Approach.* Springer, New York.