Stabilization and Identification of Nonlinear Systems

Andrew D.B. Paice

B. Sc. Hons. (UWA)

May 1992

A thesis submitted for the degree of Doctor of Philosophy
of the Australian National University

Department of Systems Engineering Research School of Physical
Sciences and Engineering
The Australian National University

Declaration

These doctoral studies were conducted with Professor John B. Moore as supervisor, and Professor Brian D.O. Anderson and Dr. Robert R. Bitmead as advisors.

The contents of this thesis are the results of original research, and have not been submitted for a higher degree at any other university or institution.

Andrew Paice

Andrew D.B. Paice
May 11, 1992
Department of Systems Engineering,
The Australian National University,
Canberra, Australia.

Abstract

In this thesis methods for the stabilization and identification of nonlinear systems are presented. By considering successful theories for the stabilization of linear systems, approaches to the nonlinear problem are uncovered. This is one of the motivating forces of this thesis. We are interested in pushing back the restrictions of linearity as far as is possible so as to both expose the more general underlying nonlinear theory and to highlight the limitations of the linear theory when applied to nonlinear systems. One question which motivates our work is: " What class of systems may be stabilized by a specific adaptive controller."

In the first part of the thesis a factorization approach to the stabilization of nonlinear systems is taken. This is initially considered from an input-output point of view. Nonlinear definitions for coprimeness are presented, and conditions for the existence of such factorizations are examined. The role of the operators represented by matrices consisting of the factors of the plant and controller in the stability and well-posedness of the system is examined.

The Youla-Kucera parameterization for the class of all controllers for a given plant is extended to the nonlinear case. These stabilizing controllers are parameterized by a single stable operator, Q. By using the left factorizations of the plant and controller it is seen that the linear results may be readily generalized. The class of all plants stabilized by a particular controller may be parameterized by the stable operator S. This leads naturally to a characterization of the class of all bounded-input stable pairs. Given an initial stable plant-controller pair, a new system derived from the original plant and controller by the parameters S and Q will be stable if and only if the feedback system of S and Q is stable. Some results in Nonlinear Robust Control follow.

By formulating a generalization of linear fractional maps, an exact relationship between the nonlinear plant (controller) and its parameterizing operator S (Q), may be derived. This may be expressed either in terms of the right or left factorizations of the plant and controller, although it is noted that unless the left factorization approach is taken, it is not clear how the stability of the original system will relate to that of the feedback system consisting of S and Q.

A state space approach to the problem is then taken. It is seen that if a solution to the smooth stabilization problem can be found, a right factorization for the plant may be derived. A candidate controller is designed, based on the idea of constructing a state estimator for the plant. This controller also has a right factorization, and some of the earlier right factorization results may be applied. A left factorization is presented for a restricted class of plants. It is seen that in order to prove coprimeness for this scheme, the plant must be augmented by a unity feedthrough term. An example of constructing a right factorization is presented.

The need for a model to construct a stabilizing controller for a given system motivates the second part of the thesis. Here the possible role of Artificial Neural Networks in nonlinear system identification is investigated. By using these networks within a Recursive Prediction Error scheme, the convergence theory of Ljung may be applied. Some analysis of this scheme shows that this approach to the problem bears promise.

The power of Neural Networks in representing functions is then investigated. Specifically an architecture is proposed for which a bound on the number of nodes required to represent a general Lipschitz continuous function may be derived.

Acknowledgements

I would not have made it to the end of this degree, and to the end of this thesis, without the help and support of a number of individuals. Here I hope to give some recognition for their efforts on my behalf.

Firstly, I thank Prof. John Moore for the supervision and support he has provided during my time here. He has shown great tolerance and patience while guiding me into a number of interesting areas of research, often in spite of my apparent efforts to resist him.

Secondly, I thank my parents Fiona and David Paice for providing me with such a good start to life. This, my first real step into the world, has been made easier through their efforts.

For providing a stimulating environment in which to work, I extend my thanks to all the staff and students of Systems Engineering of the last few years. The seminars and coursework, although sometimes arduous, have proven to be greatly beneficial.

Lastly, I want to give thanks to all those I have called friend in the last few years. There have been too many people whose paths have intersected with mine to single any out, let it be enough to say that they have given me a taste of life at least as interesting as that which this thesis represents.

Preface

This thesis is divided into 7 chapters.

Chapter 1 introduces the topics considered, and motivates the approach that will be taken by presenting an overview of some relevant linear results.

In Chapter 2 the definitions which will be required for the following chapters are given, and the connections between coprimeness, matrices of the factors of the plant and controller, and the stability of the system are explored.

Chapters 3 and 4 explore the factorization approach to feedback stability. Firstly from an input-output framework, and secondly via state space techniques.

Chapter 5 presents a nonlinear Recursive Prediction Error algorithm in which Artificial Neural Networks are used as function estimators.

Chapter 6 investigates the representation properties of an ANN by deriving an upper bound on the number of nodes required to represent any Lipschitz continuous function.

Chapter 7 concludes the thesis, summarizing the main results and indicating areas for further research.

This thesis presents my own original research as well as some results obtained in collaboration with others in the three years I have been enrolled in the Department of Systems Engineering as a PhD student. A substantial part of this work was carried out by myself, approximately 80% of the total. More specifically I contributed all the technical details of Chapters 1 to 5, and approximately 30% of those of Chapter 6.

The research in Chapters 2 to 4 was carried out mostly in collaboration with Prof. J.B. Moore, although discussions with Prof. Roberto Horowitz lead to some of the insights relating to right factorizations of Chapter 2. The research which lead to Chapter 5 was started with Prof. Mark Damborg, Prof. J.B. Moore and Dr. Robert Williamson. The technical details and simulations presented herein were done by myself. The results of Chapter 6 are largely due to Dr. Robert Williamson, my contribution was through discussions, and working out some of the technical details.

Most of this research has been either published as journal or conference papers, or been accepted for publication. A list is given below.

Journal Publications:

[1]. Andrew D.B. Paice and John B. Moore, "On the Youla-Kucera parameterization for nonlinear systems." *Systems and Control Letters* **14**, *121-129, 1990.*

[2]. Andrew D.B. Paice and John B. Moore, "Robust stabilization of nonlinear plants via left coprime factorizations." *Systems and Control Letters* **15**, *125-135, 1990.*

[3]. Andrew D.B. Paice, John B. Moore and Roberto Horowitz, " Nonlinear feedback stability via coprime factorization analysis." *Journal of Mathematical Systems Estimation and Control. Accepted 1991.*

[4]. Robert C. Williamson and Andrew D.B. Paice, " The Number of Nodes Required in a Feed-Forward Neural Network for Functional Representation." *Submitted to Neural Networks 1990. Revised 1991.*

Conference Publications:

[1]. Mark J. Damborg, Robert C. Williamson, Andrew D.B. Paice and John B. Moore, "Adaptive Nonlinear Estimation with Artificial Neural Networks" *ISITA 1990, pp. 743-746.*

[2]. Andrew D.B. Paice and John B. Moore, "Robust stabilization of nonlinear plants via left coprime factorizations." *Proceedings of the 28th CDC 1990, Vol. 6, pp. 3379-3384.*

[3]. Andrew D.B. Paice, John B. Moore and Roberto Horowitz, " Nonlinear feedback stability via coprime factorization analysis." *American Control Conference. Accepted 1992.*

Contents

List of Figures

Chapter 1

Introduction

1.1 Motivation

In standard (linear) control theory two main approaches to the control of an uncertain system may be identified, the Robust and Adaptive Control approaches. The first involves assuming that there is some form of model to work with, but acknowledging that there will be some error between the actual and modelled plant. The objective is thus to design a controller which will stabilize the modelled plant, and hence the real plant if the modelling error is small enough. Results such as the Small Gain Theorem become useful in this case. The alternative is to use an Adaptive Control scheme, either designing the controller in terms of an identifier and from the identified plant producing (on-line) an appropriate stabilizing controller, or adjusting the parameters of the controller to minimise an error index via some other algorithm.

For linear systems, the class of all stabilizing controllers for a given plant may be characterized via the Youla-Kucera parameterization. A dual result gives the class of all plants stabilized by a given controller. By designing the controller so that the true plant is likely to be in this class of stabilized plants, robust stabilization results may be obtained. A time-varying generalization of this work allows adaptive controllers to be incorporated into the theory, unless the dynamics are inherently nonlinear.

Thus, by taking a factorization approach to stability it is possible to design robust controllers, or integrate an adaptive scheme into the system in a natural way. In developing a nonlinear version of the linear results on the factorization theory, we hope to take steps towards a useful approach to nonlinear control.

Note that in either case it is important to know in which class of functions the plant that is to be stabilized belongs. Additionally, in most adaptive schemes some form of plant identification algorithm is required. Hence it is natural to consider some form of nonlinear identification scheme. This motivates the second part of the thesis in which an approach to nonlinear system identification is considered.

In this first part of the thesis an overview of the linear results of the Youla-Kucera parameterization is presented, giving an introduction to the factorization approach to the stabilization of nonlinear systems. The ideas of coprimeness and how to construct the parameterization are reviewed briefly. A short introduction to system identification is also presented. More formal definitions, which generalize to the nonlinear case, are presented in later chapters.

Initially it is assumed that the plant which is to be stabilized is linear and exactly known, and that the designed controller may be exactly implemented. In this highly simplified environment some of the underlying issues of stabilization may be identified. By characterizing the stability of the system in terms of the Bezout identities a natural method of generating other controllers which stabilize the given plant is found. The dual result is readily obtained to give the class of plants stabilized by the given controller. In this way issues such as robustness of the controller to plant uncertainties may be naturally introduced. A state space approach to the problem is touched upon. Observe that once the state space representations are given, the factorization results may be directly applied. In deriving a candidate controller the need for a plant model is highlighted. In the case where there is *a priori* knowledge of the plant, factorization theory may be applied, however in most instances a system identification problem will have to be solved. By considering on-line identification, applications to adaptive control may be found.

1.1.1 Problem Statement

Consider a linear plant $G: \mathcal{U} \mapsto \mathcal{Y}$ which we wish to stabilize by a linear controller $K: \mathcal{Y} \mapsto \mathcal{U}$, as shown in Fig 1-1. For convenience we denote this feedback control system $\{G, K\}$. Simple manipulations show that:

$$\begin{pmatrix} e_1 \\ e_2 \end{pmatrix} = \begin{bmatrix} I & -K \\ -G & I \end{bmatrix}^{-1} \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} \qquad (1.1.1)$$

Hence the feedback loop will be well-posed if this matrix inverse exists, and stable if the matrix inverse is stable. Using the principle of superposition it may be shown that the

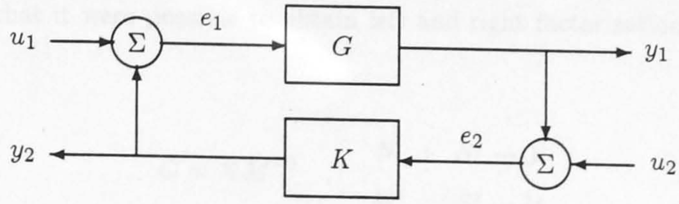

Figure 1-1: The feedback system $\{G, K\}$.

existence and stability of this operator is equivalent to finding a controller K such that the maps from each of the inputs to each of the outputs exist and are stable. Considering the effect of either u_1 or u_2 on e_1 and e_2, while the other input is set to zero, in Fig 1-1 gives:

$$\begin{aligned} e_1 &= (I - KG)^{-1}u_1 && (1.1.2)\\ e_1 &= (I - KG)^{-1}Ku_2 && (1.1.3)\\ e_2 &= (I - GK)^{-1}u_2 && (1.1.4)\\ e_2 &= (I - GK)^{-1}Gu_1 && (1.1.5)\end{aligned}$$

These equations are known as the closed-loop transfer mappings. Applying the principle of superposition we find that:

$$\begin{bmatrix} I & -K \\ -G & I \end{bmatrix}^{-1} \text{exists} \iff \left\{ \begin{array}{l} (I - KG)^{-1} \\ (I - KG)^{-1}K \\ (I - GK)^{-1} \\ (I - GK)^{-1}G \end{array} \right. \text{exist} \qquad (1.1.6)$$

$$\begin{bmatrix} I & -K \\ -G & I \end{bmatrix}^{-1} \text{is stable} \iff \left\{ \begin{array}{l} (I - KG)^{-1} \\ (I - KG)^{-1}K \\ (I - GK)^{-1} \\ (I - GK)^{-1}G \end{array} \right. \text{are stable} \qquad (1.1.7)$$

Hence the problem of stabilizing G is equivalent to the problem of finding K such that the right hand sides of (1.1.6) and (1.1.7) are satisfied.

1.1.2 Coprime Factors

Now suppose that it were possible to obtain left and right factorizations of G and K, as follows.

$$G = NM^{-1} \quad , \quad \begin{array}{lcl} N & : & \mathcal{S}^{\mathrm{r}} \rightarrow \mathcal{Y} \\ M & : & \mathcal{S}^{\mathrm{r}} \rightarrow \mathcal{U} \end{array} \tag{1.1.8}$$

$$G = \tilde{M}^{-1}\tilde{N} \quad , \quad \begin{array}{lcl} \tilde{N} & : & \mathcal{U} \rightarrow \mathcal{S}^{\mathrm{l}} \\ \tilde{M} & : & \mathcal{Y} \rightarrow \mathcal{S}^{\mathrm{l}} \end{array} \tag{1.1.9}$$

$$K = UV^{-1} \quad , \quad \begin{array}{lcl} U & : & \mathcal{S}^{\mathrm{l}} \rightarrow \mathcal{U} \\ V & : & \mathcal{S}^{\mathrm{l}} \rightarrow \mathcal{Y} \end{array} \tag{1.1.10}$$

$$K = \tilde{V}^{-1}\tilde{U} \quad , \quad \begin{array}{lcl} \tilde{U} & : & \mathcal{Y} \rightarrow \mathcal{S}^{\mathrm{r}} \\ \tilde{V} & : & \mathcal{Y} \rightarrow \mathcal{S}^{\mathrm{r}} \end{array} \tag{1.1.11}$$

Where $\mathcal{S}^{\mathrm{r}}$ and $\mathcal{S}^{\mathrm{l}}$ are appropriate factorization spaces. This is not an unreasonable assumption. For example if the system is well-posed, then an obvious left factorization for G is $G = [(I - GK)]\,[(I - GK)^{-1}G]$. As these are all linear matrices, it is possible to show that $(I - GK)^{-1}G = G(I - KG)^{-1}$, so G will have a right factorization $G = [G(I - KG)^{-1}]\,[(I - KG)]$.

The right hand sides of (1.1.6) and (1.1.7) now transform as follows:

$$(I - KG)^{-1} = M(\tilde{V}M - \tilde{U}N)^{-1}\tilde{V} \tag{1.1.12}$$

$$(I - KG)^{-1}K = M(\tilde{V}M - \tilde{U}N)^{-1}\tilde{U} \tag{1.1.13}$$

$$(I - GK)^{-1} = V(\tilde{M}V - \tilde{N}U)^{-1}\tilde{M} \tag{1.1.14}$$

$$(I - GK)^{-1}G = V(\tilde{M}V - \tilde{N}U)^{-1}\tilde{N} \tag{1.1.15}$$

Thus, if the factors of G and K are stable and the operators $(\tilde{V}M - \tilde{U}N)^{-1}$ and $(\tilde{M}V - \tilde{N}U)^{-1}$ exist and are stable, then the right hand sides of (1.1.6) and (1.1.7) will be satisfied. If the feedback system is stable then the closed loop transfer mappings will be stable. Thus the factorizations suggested previously for G will be stable factorizations. Furthermore it is possible to choose these factorizations such that:

$$\tilde{M}V - \tilde{N}U = I \tag{1.1.16}$$

$$\tilde{V}M - \tilde{U}N = I \tag{1.1.17}$$

These equations are known as the Bezout identities, and they identify a property of the factors known as *coprimeness* [57]. It is easily verified that given right and left coprime factorizations, that (1.1.16) and (1.1.17) are equivalent to the following matrix equation.

$$\begin{bmatrix} \tilde{V} & -\tilde{U} \\ -\tilde{N} & \tilde{M} \end{bmatrix} \begin{bmatrix} M & U \\ N & V \end{bmatrix} = I \qquad (1.1.18)$$

The results of this section are summarized in the following lemma.

Lemma 1.1 *Given a plant G with stable right and left factorizations (1.1.8), (1.1.9), suppose there exist stable maps U, V, $\tilde{U}$, $\tilde{V}$ with V, $\tilde{V}$ invertible, which satisfy the identity (1.1.18). Then $K = UV^{-1} = \tilde{V}^{-1}\tilde{U}$ is a stabilizing controller for G, and these are coprime factorizations for G and K.* □

1.1.3 The Youla-Kucera Parameterization

We have established that if there exist stable right and left factorizations for the plant G and controller K such that the Bezout identities (1.1.16), (1.1.17) hold, then the system $\{G, K\}$ is well-posed and stable. Note that no guarantee of the uniqueness of this controller is given. Additionally, in this idealized case, nothing can be said about the effect of approximations to the plant and controller being used in place of the original plant and controller. This leads us to consider whether the system of equations which have been developed could be used to determine if other controllers may stabilize the plant, and how an approximation to the plant or controller may be stabilized.

Given a stable factorization for the plant and controller as in (1.1.8)-(1.1.11) the stability of the system is determined by the Bezout identities. Consider the following identities.

$$\begin{aligned} \tilde{M}(V+NQ) - \tilde{N}(U+MQ) &= \tilde{M}V - \tilde{N}U + (\tilde{M}N - \tilde{N}M)Q \\ &= I \qquad (1.1.19) \\ (\tilde{V}+Q\tilde{N})M - (\tilde{U}+Q\tilde{M})N &= \tilde{V}M - \tilde{U}N + Q(\tilde{N}M - \tilde{M}N) \\ &= I \qquad (1.1.20) \end{aligned}$$

It is possible to show (see Appendix A.1) that these define right and left factorizations of a new controller, K_Q.

$$\begin{aligned} K_Q &= \tilde{V}_Q^{-1}\tilde{U}_Q \\ &= (\tilde{V}+Q\tilde{N})^{-1}(\tilde{U}+Q\tilde{M}) \qquad (1.1.21) \\ &= U_Q V_Q^{-1} \\ &= (U+MQ)(V+NQ)^{-1} \qquad (1.1.22) \end{aligned}$$

If the operator Q is stable, then (1.1.21) and (1.1.22) describe stable factorizations for K_Q which satisfy the Bezout identities (1.1.16), (1.1.17), and are thus coprime. Hence the system $\{G,\, K_Q\}$ is stable. This is summarized and extended in the following lemma.

Lemma 1.2 *Given a stable plant, controller pair with stable right and left coprime factorizations, (1.1.8)-(1.1.11), such that (1.1.18) is satisfied. Then the operator K_Q as defined in (1.1.21), (1.1.22) will stabilize the plant G iff Q is stable. Furthermore, given a controller $K^\star$ such that $\{G,\, K^\star\}$ is stable, there will exist a stable operator $Q^\star$ such that $K_{Q^\star} = K^\star$, this operator, $Q^\star$, will be given by:*

$$\begin{aligned} Q^\star &= M^{-1}(I-K^\star G)^{-1}(K^\star - K)V \qquad (1.1.23) \\ &= \tilde{V}(K^\star - K)(I-GK^\star)^{-1}\tilde{M}^{-1} \qquad (1.1.24) \end{aligned}$$

□

This is known as the *Youla-Kucera parameterization* of the class of all controllers of the plant G. It parameterizes every controller which will stabilize G in terms of a stable map Q. Hence, by finding a single stabilizing controller, all possible controllers may be generated. It is now possible to optimise the controller with respect to a given performance criteria by searching over the space of stable Q. Additionally we may now consider the effect of a different controller, $K^\star$, being used in place of the original, K, through the use of the parameter $Q^\star$ as given in (1.1.24). If the $Q^\star$ thus generated is stable, then $K^\star$ will stabilize G. Furthermore, this gives a measure of how robust the system is to errors in implementing the controller. Given that the difference between the desired controller K, and the implemented controller $K^\star$, is small enough, $Q^\star$ will be stable. Thus the system will be stable and robust with respect to small errors in controller implementation.

Additionally this gives a natural way of incorporating *a priori* knowledge of the plant into an adaptive scheme. A controller is designed to stabilize a plant with the known properties, and an adaptive controller is then incorporated into Q, which then adapts to stabilize unknown plant characteristics. See, for example, Tay [47], [48].

The dual results to those obtained for the controller may be readily obtained, giving the class of all plants stabilized by a given controller, and parameterized by the stable operator S.

$$G_S = (N + VS)(M + US)^{-1} = (\tilde{M} + S\tilde{U})^{-1}(\tilde{N} + S\tilde{V}) \tag{1.1.25}$$

This leads to results in robust control. The parameter S gives a measure of how robust the system is to uncertainties in the plant.

$$\begin{aligned} S &= V^{-1}(I - G_S K)^{-1}(G_S - G)M & (1.1.26)\\ &= \tilde{M}(G_S - G)(I - KG_S)^{-1}\tilde{V}^{-1} & (1.1.27) \end{aligned}$$

If the difference between the actual and nominal plant is small enough, then the system will be stable.

Further, these results may be used to prove that the problem of simultaneously stabilizing m plants is equivalent to the problem of stabilizing $m - 1$ plants with a stable controller.

The challenge is now to consider the stability of the system $\{G_S, K_Q\}$. The Youla-Kucera parameterization gives results on how to account for uncertainties in the plant or the controller, but not both. By considering the stability of $\{G_S, K_Q\}$ a more complete theory is obtained.

An alternative proof to Lemma 1.2 is instructive. It is possible to show that if the factorizations of G and K are coprime, the following results hold.

$$\begin{bmatrix} I & -K \\ -G & I \end{bmatrix}^{-1} \text{is stable} \Leftrightarrow \begin{bmatrix} \tilde{V} & -\tilde{U} \\ -\tilde{N} & \tilde{M} \end{bmatrix}^{-1} \text{is stable} \tag{1.1.28}$$

$$\begin{bmatrix} I & -K \\ -G & I \end{bmatrix}^{-1} \text{is stable} \Leftrightarrow \begin{bmatrix} M & -U \\ -N & V \end{bmatrix}^{-1} \text{is stable} \tag{1.1.29}$$

From (1.1.20) we know that $K_Q = \tilde{V}_Q^{-1}\tilde{U}$ is a left coprime factorization. Note that:

$$\begin{bmatrix} \tilde{V}_Q & -\tilde{U}_Q \\ -\tilde{N} & \tilde{M} \end{bmatrix}^{-1} = \begin{bmatrix} \tilde{V} + Q\tilde{N} & -\tilde{U} - Q\tilde{M} \\ -\tilde{N} & \tilde{M} \end{bmatrix} \tag{1.1.30}$$

$$= \left\{ \begin{bmatrix} I & -Q \\ 0 & I \end{bmatrix} \begin{bmatrix} \tilde{V} & -\tilde{U} \\ -\tilde{N} & \tilde{M} \end{bmatrix} \right\}^{-1} \tag{1.1.31}$$

$$= \begin{bmatrix} \tilde{V} & -\tilde{U} \\ -\tilde{N} & \tilde{M} \end{bmatrix}^{-1} \begin{bmatrix} I & Q \\ 0 & I \end{bmatrix} \tag{1.1.32}$$

Since $\begin{bmatrix} \tilde{V} & -\tilde{U} \\ -\tilde{N} & \tilde{M} \end{bmatrix}^{-1} = \begin{bmatrix} M & U \\ N & V \end{bmatrix}$, it is straightforward to see that $\{G, K_Q\}$ is stable iff Q is stable. Additionally this approach gives an approach to the problem of considering the stability of $\{G_S, K_Q\}$. See Appendices A.2 and A.3 for details.

$$\begin{bmatrix} \tilde{V}_Q & -\tilde{U}_Q \\ -\tilde{N}_S & \tilde{M}_S \end{bmatrix}^{-1} = \begin{bmatrix} \tilde{V} & -\tilde{U} \\ -\tilde{N} & \tilde{M} \end{bmatrix}^{-1} \begin{bmatrix} I & -Q \\ -S & I \end{bmatrix}^{-1} \tag{1.1.33}$$

$$\begin{bmatrix} M_S & -U_Q \\ -N_S & V_Q \end{bmatrix}^{-1} = \begin{bmatrix} I & -Q \\ -S & I \end{bmatrix}^{-1} \begin{bmatrix} M & -U \\ -N & V \end{bmatrix}^{-1} \tag{1.1.34}$$

These equations indicate that the stability of $\{G_S, K_Q\}$ is tied to the stability of the system $\{S, Q\}$. In [51] it is shown that $\{G_S, K_Q\}$ is stable iff $\{S, Q\}$ is stable. This characterizes the class of all stabilizing plant, controller pairs.

It is now possible to combine the adaptive and robustness results in a natural way. By searching over the class of Q we can adaptively stabilize the plant, while the map S gives a measure of robustness.

1.1.4 State Space Realizations

It has been established that once factorizations of the plant and controller have been found a theory which may incorporate the results of Robust and Adaptive control is developed. The problem is now to discover whether such factorizations may be found for a general linear plant.

For a good review of the solution to this problem, see [57]. The problem of finding a right coprime factorization *(rcf)* for G turns out to be related to the stabilizability of G, while finding a left coprime factorization *(lcf)* is related to the detectability of G.

Suppose $G\colon \mathcal{U} \mapsto \mathcal{Y}$ is a continuous time linear plant with a feed-through term, then G

will have a state space realization given by:

$$G(x_0): \begin{array}{rcll} \dot{x} & = & Ax + Bu & x(0) = x_0 \\ y & = & Cx + Du & \end{array} \tag{1.1.35}$$

Where A, B, C, D are matrices. Then suppose there exists a matrix F such that $A + BF$ is stable, *i.e.* the eigenvalues of $A + BF$ lie in the left half plane. Then $G(x_0)$ has a right coprime factorization given by:

$$G(x_0) = N(x_0)M(x_0)^{-1} \tag{1.1.36}$$

$$M(x_0) \quad : \quad \begin{array}{rcll} \dot{x}_m & = & (A + BF)x_m + Bs & x_m(0) = x_0 \\ u & = & Fx_m + s & \end{array} \tag{1.1.37}$$

$$N(x_0) \quad : \quad \begin{array}{rcll} \dot{x}_n & = & (A + BF)x_n + Bs & x_n(0) = x_0 \\ y & = & Cx_n + Ds & \end{array} \tag{1.1.38}$$

Dually, if there exists a matrix H such that the matrix $A + HC$ is stable, then there exists a left coprime factorization for $G(x_0)$ as follows

$$G(x_0) = \tilde{M}(x_0)^{-1}\tilde{N}(x_0) \tag{1.1.39}$$

$$\tilde{M}(x_0) \quad : \quad \begin{array}{rcll} \dot{x}_m & = & Ax_m + H(Cx_m - y) & x_m(0) = 0 \\ s & = & Cx_m - y & \end{array} \tag{1.1.40}$$

$$\tilde{N}(x_0) \quad : \quad \begin{array}{rcll} \dot{x}_n & = & Ax_n + Bu - H(Cx_n + Du) & x_n(0) = x_0 \\ y & = & Cx_n + Ds & \end{array} \tag{1.1.41}$$

Additionally, given the matrices F and H as above, it is possible to construct a controller for $G(x_0)$ as follows.

$$K(x_0) \quad : \quad \begin{array}{rcll} \dot{x}_k & = & (A + BF)x_k + H(y - (C + DF)x_k) & x_k(0) = x_0 \\ u & = & Fx_k & \end{array} \tag{1.1.42}$$

This controller has stable left and right coprime factorizations which satisfy the Double Bezout Identity (1.1.18).

1.1.5 Identification

It may be seen that the controller (1.1.42) stabilizes the plant by firstly acting as a state estimator for the $G(x_0)$, and then setting the input to the plant to be the stabilizing state feedback Fx. Note the importance of having an accurate model of the plant. If the plant is not known it is not possible to design a controller based on this approach. Hence, when considering the problem of controlling an unknown plant, some form of identification algorithm must first be considered. Then, when a model of the plant is known, it is possible to design a nominal controller.

An on-line version of this idea is used in Adaptive Control. This is known as the Certainty Equivalence Principle. It is assumed that at each time instant the estimate of the plant model within the adaptive controller is an accurate one, and the controller outputs are based on this. The plant identification parameters are then updated based on the difference between the predicted and the actual output of the plant. Here it is important to have a parameterization which may be adjusted to find a representation of the plant.

Thus the importance of system identification has been established. It is straightforward to see that the parameterization used in the identification algorithm is of crucial importance. In performing the initial identification it is important to use as general a class of models as possible. Consider the single input, single output (SISO) case first.

Any linear, time invariant, discrete time, SISO plant has a representation as follows:

$$\begin{aligned} y_k &= a_1 y_{k-1} + a_2 y_{k-2} + \ldots + b_0 u_k + b_1 u_{k-1} + b_2 u_{k-2} + \cdots && (1.1.43) \\ &= \sum_{i=1}^{\infty} a_i y_{k-i} + \sum_{i=0}^{\infty} b_i u_{k-i} && (1.1.44) \end{aligned}$$

This allows far more generality than is necessary, however. Most linear systems of interest are of finite order, and thus have the form:

$$y_k = \sum_{i=1}^{n} a_i y_{k-i} + \sum_{i=0}^{m} b_i u_{k-i} \qquad (1.1.45)$$

This allows a very simple form of plant identification. First, nominal values for m and n are chosen, then the parameter set, $a_1, \ldots, a_n, b_0, \ldots, b_m$, which leads to a representation of the plants input-output behaviour is determined.

Formally, define the parameter $\theta^T = (a_1, \ldots, a_n, b_0, \ldots, b_m)$. Given an input sequence of length l, $u = \{u_i\}_{i=0}^{l}$, define y_0 to be the sequence of length l resulting from substituting u into (1.1.35), and let y_θ be the sequence resulting from (1.1.45), where $u_i = 0$, $\forall i < 0$. Note that it may be the case that $l = \infty$. Then the identification problem becomes one of solving the following problem:

$$\min_{\theta} ||y_0 - y_\theta|| \tag{1.1.46}$$

There are a few difficulties with taking this approach to system identification. Firstly, the effects of the initial conditions may only be accounted for by estimating values for $\{u_i\}_{i=-m}^{-1}$. Additionally, once we have an estimate for θ, it is then necessary to calculate a realization of the form of (1.1.35). Since it is usually the case that there are errors in the measurement of the inputs and outputs of the plant it is also necessary to incorporate noise into the estimation scheme. Note also that it is difficult to include *a priori* knowledge about the plant in such a scheme, and generalization to multi-input, multi-output systems is not straightforward. Hence consider a parameterization of the form:

$$\begin{aligned} x_{k+1} &= A(\theta)x_k + B(\theta)u_k + v_k \\ y_k &= C(\theta)x_k + D(\theta)u_k + w_k \end{aligned} \tag{1.1.47}$$

Estimating the noise sequence v_k is difficult in this framework, as it is not easy to separate the effects of the noise w_k from that of the filtered noise Kv_k. Additionally, due to the non uniqueness of a realization for $G(x_0)$ of this form, convergence of the parameter θ is difficult to guarantee. These problems may be solved by using an innovations representation, where it is assumed that $v_k = Kw_k$, and K is then estimated by $K(\theta)$.

$$\begin{aligned} x_{k+1} &= A(\theta)x_k + B(\theta)u_k + K(\theta)w_k \\ y_k &= C(\theta)x_k + w_k \end{aligned} \tag{1.1.48}$$

Having found the form of the model estimate that will be used, we choose an algorithm that gives the estimate of θ. These algorithms may be divided into two broad classes, based on whether they use on-line or off-line techniques. Off-line algorithms seek to minimise a cost such as (1.1.46) given an input signal and the resulting output signal. An alternative is an on-line algorithm which, at each time step, updates the estimate of the state and the parameter θ. In this case we seek to minimise a prediction error index, with respect to the parameter θ, for example:

$$\min_{\theta_k} ||y_k - \hat{y}_{k|\theta}|| \tag{1.1.49}$$

Note that an on-line algorithm may be used as part of an adaptive controller, which may in turn be incorporated into an existing feedback system through the factorization framework. Alternatively it may be simplified to give a state estimator, rather than a combined state and parameter estimator, thus leading to a controller such as (1.1.42). Hence we have more interest in on-line estimators than off-line estimators. One example of an on-line estimation scheme which may be generalized to the nonlinear case is the Recursive Prediction Error scheme, which will be studied in Chapter 5.

Note that in the linear case we are constrained to using matrices for $A(\theta)$, $B(\theta)$, $C(\theta)$, $D(\theta)$, hence it is straightforward to obtain a parameterization, once the dimension of the state estimate $\hat{x}_k$ has been decided upon.

1.2 Thesis Outline.

In the first three chapters we develop a factorization approach to the problem of stabilizing the system $\{G, K\}$ of Fig 1-1 when the plant and controller may be nonlinear. The approach taken is similar to the development given in the previous section. By finding nonlinear generalizations of these results it is hoped that some of the Robust Stabilization and Adaptive Control results may be reproduced in a nonlinear setting.

In Chapter 2 the stage is set for later work. First the signal spaces which will be used are defined. Ideas for the stability of nonlinear operators are then presented, and definitions of the well-posedness and stability of the closed loop $\{G, K\}$ are given. Coprimeness definitions are developed from an input-output perspective rather than algebraically, and results concerning the relationship between the fractional descriptions of the plant and controller and the stability of the system are presented. This gives a body of results on which a more complete nonlinear factorization theory can be based.

The Youla-Kucera parameterization for nonlinear systems is developed in Chapter 3. The class of all controllers stabilizing a given plant is derived, and the class of all plants stabilized by a given controller is presented. This leads to a characterization of the class of all stabilizing plant-controller pairs. Some nonlinear robust stabilization results may now be easily proved.

In Chapter 4 a state space approach to the factorization of nonlinear systems is presented. Techniques are demonstrated which lead to right factorizations for a plant for which one can solve the smooth stabilization problem. The problem of solving left factorizations does not appear to be solvable within the presented framework, however a solution due to Moore and Irlicht [36] is presented to demonstrate that there are alternative approaches which are met with some success.

This completes our investigations into taking a factorization approach to the stabilization of nonlinear systems. The discussion of state space realizations and left factorizations leads us to consider the problem of the identification of nonlinear systems.

In Chapter 5 research into the use of a Recursive Prediction Error algorithm in conjunction with Artificial Neural Networks (ANNs) to perform nonlinear system identification is presented. It is found that the results of the theory on recursive stochastic algorithms due to Ljung [31], [33], may be applied to prove convergence of the proposed scheme for some cases. Simulation studies are also presented.

As ANNs are being used to represent the nonlinear operators of Chapter 5, we are motivated to consider the power of an ANN to represent a given function. It is already known that ANNs may act as universal approximators. In Cybenko [4], Funahashi [11] and Hornik, Stinchcombe and White [22], for instance it is shown that given a sufficient number of nodes, any function may be approximated to any given accuracy. However, the reverse problem of stating the number of nodes required to represent a function from a given functional class to a prescribed accuracy is yet to be solved. This problem is considered in Chapter 6. An ANN architecture is proposed for which the number of nodes required to represent any map from the class of Lipschitz continuous functions defined on a compact domain, is determined. It is then possible to calculate the bit complexity of this representation, which may then be compared with the ε-entropy for this class of functions. In this way an idea of the efficiency of the proposed architecture to represent general Lipschitz continuous functions is obtained.

Conclusions and suggestions for further work are given in Chapter 7.

Chapter 2

Preliminaries

2.1 Introduction

As seen in the introduction, the study of coprime factorizations of linear systems leads to a theory giving the class of all stabilizing controllers for a linear plant, the class of plants stabilized by a given controller, and thus to a theory which may be used to derive results in Robust and Adaptive Control. The key concept in this field is that of coprimeness, considered as resulting from the Bezout identity. As seen in Section 1.1, if a plant and controller satisfy a Bezout identity the class of all controllers and plants may be naturally generated. In this chapter we seek a definition of coprimeness which allows us to generalize these results.

The Bezout approach to coprimeness was initially carried over to the nonlinear field see Desoer [6, 7], Verma [55]. Verma [54] took a geometric approach and found Bezout-independent definitions which used the idea of the graph $\mathcal{G}_r(G)$, of the plant G. It was demonstrated that there is a set-theoretic definition of right coprimeness, which is equivalent to the Bezout definition for linear operators, based on the idea of preventing pole-zero cancellations between the factors. This generalizes readily to give a right coprimeness definition for arbitrary nonlinear operators. However the approach to take for defining left coprimeness is not as clear. A set-theoretic left coprimeness definition based on preventing pole-zero cancellations is not equivalent to the factorization satisfying a Bezout identity.

Hammer, in his series of papers [13]-[16] considered a left factorization approach to the feedback stabilization problem. In this work a definition of left coprimeness is developed based on an input-output approach. The plant is restricted to be injective (one to one), and the left factorization derived is first used within a Bezout identity. Tay [49] generalized these results to a slightly larger class of nonlinear plants, including noninjective plants which are constrained such that the pre-image of any output signal is either bounded, or has no elements which are bounded. Once again the left factorization is used within a Bezout identity. Thus it is not clear whether left coprimeness should be defined via a Bezout identity or a set-theoretic condition.

The interest in finding Bezout independent definitions comes from results such as (1.1.28), and the dual result for right factorizations. Given an arbitrary plant and controller each with left factorizations, we can test for stability of the system via (1.1.28) if the factorizations are coprime. It appears to be easier to check a set-theoretic definition than to construct solutions to the Bezout identity. Additionally the interpretation of coprimeness is more straight forward with a set based definition in that the links to preventing pole-zero cancellations are more evident than when coprimeness is considered in the context of a Bezout identity.

In this chapter we set the stage for the following chapters. First definitions for the stability and well-posedness of a feedback system with external inputs are given. We then move onto the definitions of left and right coprimeness developed by us in [40, 41], which represent a natural generalization of the idea of right-half-plane coprimeness for continuous time systems. These are based on the definitions presented by Tay and by Hammer. It is observed that the condition imposed on the plant in [49] is both necessary and sufficient for the existence of a *lcf* for a plant.

The connections between the well-posedness and stability of a system, and the factors of the plant and controller are considered. It is found that for *rcfs* the results are comparable to those available in linear systems theory. However, the connection is not clear for *lcfs*. The linear results obtained in this case rely implicitly on the use of the principle of superposition, which is disallowed in the nonlinear case. The main contribution here is to demonstrate ways of getting around this restriction. The idea is to find a way of restricting the operator so that the change in the output may be bounded when the change in the input is known. The method that we use is to use differential boundedness, which bounds the change in the output of an operator given that the input changes by less than a prescribed amount. It may be seen that if the plant satisfies a Lipschitz condition, where the change in the output is bounded by an amount proportional to the change in the input, or some similar condition, the results will still hold.

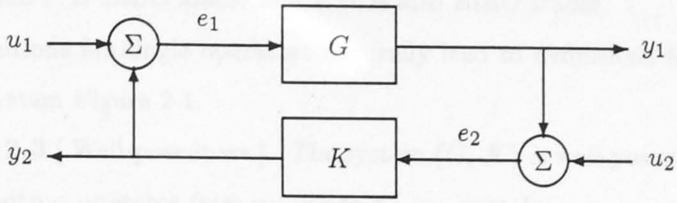

Figure 2-1: The feedback system $\{G, K\}$.

2.2 Signal Spaces and Stability

In the sequel the stability problem for the feedback system $\{G, K\}$, as in Figure 2-1, is developed from an input-output point of view. The mapping G will represent the plant, and K the controller.

A signal, x, is referred to as coming from some vector space $\mathcal{X}$ without saying whether it is a discrete or continuous time signal. We partition the space $\mathcal{X}$ into two subspaces, $\mathcal{X}_b$ and $\mathcal{X}_u$. The former consists of all signals in $\mathcal{X}$ which are bounded, or stable, while the latter consists of all signals in $\mathcal{X}$ which are unbounded. The signal $x \in \mathcal{X}$ is said to be bounded when $||x||$ is finite, for some norm $|| \cdot ||$.

As Hammer pioneered the work in this area in discrete time, we use his notation when presenting discrete time results. In particular we work with the signal sequences $S_0(R^n)$, the set of all sequences with elements in R^n, where R is the set of extended real numbers, such that all elements of the sequence before the 0^{th} place are zero. We also work with the set of signals $S_0(\varepsilon^n)$, the subset of $S_0(R^n)$ which has the elements of its sequences bounded by ε. When we do not want to specify the explicit bound we will use the notation $S_u(R^n)$ and $S_b(R^n)$ to denote the unbounded and bounded subspaces of $S_0(R^n)$, respectively.

Continuous time systems with real input spaces are also considered. Given a real vector space $\mathcal{X}$, the space $C(\mathcal{X})$ is the space of continuous functions with continuous first derivative, mapping from some open interval of $\mathbb{R}$ to $\mathcal{X}$. The subspaces of bounded and unbounded signals are denoted $C_b(\mathcal{X})$ and $C_u(\mathcal{X})$, respectively.

The definition of stability that is to be used is now presented. It has a very general form so that it may account for the various specific notions of stability which exist. Almost any specific stability definition may be constructed by an appropriate definition of the spaces $\mathcal{U}_b$ and $\mathcal{Y}_b$.

Definition 2.1 [BIBO Stability] *A map $F: \mathcal{U} \to \mathcal{Y}$ is said to be bounded-input, bounded-output stable (*BIBO stable*) when the image of $\mathcal{U}_b$ under F is contained in $\mathcal{Y}_b$.*

Definition 2.2 [Unimodularity] *An invertible operator $F: \mathcal{U} \to \mathcal{Y}$ is said to be* unimodular *when F is BIBO stable and F^{-1} is also BIBO stable.*

These definitions for single operators naturally lead to definitions for the stability of the feedback system Figure 2-1.

Definition 2.3 [Well-posedness] *The system $\{G, K\}$ is* well-posed *if the closed-loop system input-output operator from u_1, u_2 to e_1, e_2, namely*

$$\begin{bmatrix} I & -K \\ -G & I \end{bmatrix}^{-1} \textit{exists.} \tag{2.2.1}$$

Remark 2.1 Note that if (2.2.1) holds, the closed loop transfer mappings will exist. *i.e.* $(I-GK)^{-1}$ and $(I-KG)^{-1}$) will exist. This may be seen by considering one of u_1, u_2 identically zero. However the converse does not hold due to G and K being nonlinear. Thus the implication in (1.1.6) holds, but the reverse implication does not.

In the sequel only those systems which are well-posed will be considered.

Definition 2.4 [Internal Stability] *The system $\{G, K\}$, assumed well-posed, is said to be* internally stable *iff for all bounded-inputs u_1, u_2 the outputs y_1, y_2 and e_1, e_2 are bounded. This is equivalent to*

$$\begin{bmatrix} I & -K \\ -G & I \end{bmatrix}^{-1} \textit{is BIBO stable.} \tag{2.2.2}$$

Remark 2.2 Further to Remark 2.1, (2.2.2) is a sufficient condition for the stability of the closed loop transfer mappings, but the converse does not hold.

As we are dealing with nonlinear systems a notion of local stability will be required. Formally, local stability is defined in terms of the norm of the largest signal which will not destabilize the system.

Definition 2.5 [Bounded-Input Stability] *The system $\{G, K\}$, assumed well-posed, is said to be ε_1, ε_2* bounded-input stable *iff for all inputs $|u_1| < \varepsilon_1$, $|u_2| < \varepsilon_2$ the outputs y_1, y_2 and e_1, e_2 are bounded.*

Remark 2.3 Note that internal stability is a stronger condition than bounded-input stability. All internally stable systems, $\{G, K\}$, are ε_1, ε_2 bounded-input stable for all ε_1, ε_2. Additionally, all ε_1, ε_2 bounded-input stable systems are ε_1', ε_2' bounded-input stable for all $\varepsilon_1' \leq \varepsilon_1$, $\varepsilon_2' \leq \varepsilon_2$.

Remark 2.4 In the linear case all bounded-input systems are internally stable. If

$|u_1| < \varepsilon_1$, $|u_2| < \varepsilon_2$ implies that the outputs y_1, y_2 and e_1, e_2 are bounded, then by linearity it is possible to bound the output resulting from any other bounded signal. Hence the system is BIBO stable.

2.3 Coprimeness

If a factorization approach to the stabilization of the plant G is to be taken in analogy with the linear theory of Youla-Kucera parameterizations, the concepts of right and left coprimeness should be explored in a nonlinear systems context. Definitions of right and left coprimeness are now given, and are explored in the following sections.

The following definitions were first presented in [41], and have been developed from the point of view of preventing the nonlinear equivalent of unstable pole-zero cancellations, and thus, for linear systems, specialize to right half plane coprimeness. Motivation for taking this approach to coprimeness may be found in Hammer [15].

Definition 2.6 [Right Coprimeness] *Let M, N be a right factorization for $G : \mathcal{U} \to \mathcal{Y}$*

$$G = NM^{-1} \quad , \quad N : \mathcal{S}^{\mathrm{r}} \to \mathcal{Y}$$
$$M : \mathcal{S}^{\mathrm{r}} \to \mathcal{U} \tag{2.3.1}$$

where M and N are BIBO stable mappings from the factorization space $\mathcal{S}^{\mathrm{r}}$ to the input and output spaces. Then M, N is a right coprime factorization of G *(rcf) iff for all unbounded inputs $s \in \mathcal{S}^{\mathrm{r}}_u$, Ms or Ns is unbounded.*

Definition 2.7 [Left Coprimeness] *Let $\tilde{M}$, $\tilde{N}$ be a left factorization for $G : C(\mathcal{U}) \to C(\mathcal{Y})$*

$$G = \tilde{M}^{-1}\tilde{N} \quad , \quad \tilde{N} : \mathcal{U} \to \mathcal{S}^{\mathrm{l}}$$
$$\tilde{M} : \mathcal{Y} \to \mathcal{S}^{\mathrm{l}} \tag{2.3.2}$$

where $\tilde{M}$, $\tilde{N}$ are BIBO stable mappings from the input and output spaces to the factorization space $C(\mathcal{S}_{\mathrm{l}})$. Then $\tilde{M}$, $\tilde{N}$ is a left coprime factorization of G *(lcf) iff the set of all unbounded $u \,\epsilon\, C_u(\mathcal{U})$ such that Gu is bounded and $\tilde{N}u$ is unbounded is the empty set, $\emptyset$. In other words, for all bounded $s \in C_b(\mathcal{S}_{\mathrm{l}})$, $\tilde{M}^{-1}s$ is bounded or $\{u : \tilde{N}u = s\}$ is bounded, which is an explicit dual statement of the definition for right coprimeness.*

If the system $\{G,\ K\}$ is well-posed and stable, assume that in addition to having stable coprime descriptions for G as in (2.3.1) and (2.3.2) there are factorizations for $K\colon \mathcal{Y} \mapsto \mathcal{U}$. The problem of the existence of such factorizations will be considered in the next two subsections.

$$\begin{aligned} K = UV^{-1} \quad &, \quad U : \mathcal{S}^{\mathrm{l}} \to \mathcal{U} \\ &\quad\ V : \mathcal{S}^{\mathrm{l}} \to \mathcal{Y} \end{aligned} \tag{2.3.3}$$

$$\begin{aligned} K = \tilde{V}^{-1}\tilde{U} \quad &, \quad \tilde{U} : \mathcal{Y} \to \mathcal{S}^{\mathrm{r}} \\ &\quad\ \tilde{V} : \mathcal{U} \to \mathcal{S}^{\mathrm{r}} \end{aligned} \tag{2.3.4}$$

where V, U, $\tilde{V}$, $\tilde{U}$ are BIBO stable operators and $\mathcal{S}^{\mathrm{l}}$ and $\mathcal{S}^{\mathrm{r}}$ are the factorization spaces.

In this section the definitions of stability that we will be using have been presented, along with set-theoretic definitions of coprimeness. In this section the existence of these factorizations, and the relationship between the stability and well-posedness of the system and the factors, and matrices of these factors is explored.

2.3.1 Right Coprime Factorization Results

We first review the connection between right coprime factorizations and the Bezout identity.

Lemma 2.1 [42] *Given a stable right factorization of G, as in (2.3.1), suppose that there exists a BIBO stable mapping $L\colon \mathcal{U} \times \mathcal{Y} \mapsto \mathcal{S}^{\mathrm{r}}$ such that*

$$L \begin{bmatrix} M \\ N \end{bmatrix} = Z, \quad Z \textit{ unimodular} \tag{2.3.5}$$

Then $G = NM^{-1}$ is a right coprime factorization for G □

Proof. Consider $L\colon \mathcal{U} \times \mathcal{Y} \mapsto \mathcal{S}^{\mathrm{r}}$ a BIBO stable mapping which satisfies (2.3.5). Suppose that N, M is not a coprime factorization for G. Then there exists an unbounded $s \in C(\mathcal{S}_{\mathrm{r}})$ such that Ms and Ns are both bounded. As L is BIBO, $L\begin{pmatrix} Ms \\ Ns \end{pmatrix} = Zs$ is bounded, however as Z is unimodular, Zs is unbounded. This gives a contradiction, proving the result. ∎

Remark 2.5 In the case that $L = \begin{bmatrix} L_1 & L_2 \end{bmatrix}$, this lemma specializes to Lemma 2.1 of [40].

Remark 2.6 This result specializes directly to the linear case.

The link between well-posedness and stability of the system $\{G, K\}$, and the existence of *rcf* of G is now explored. These results were first presented by Hammer [12, 14] in discrete time.

Lemma 2.2 (Review) *Consider a nonlinear plant $G : S_0(R^m) \rightarrow S_0(R^n)$ such that the inverse image of an unbounded element of the range of G is either bounded, or contains no elements which are bounded. Furthermore, suppose that there exists a feedback controller $K : S_0(R^n) \rightarrow S_0(R^m)$, as in Fig. 2-1 such that the closed-loop is well posed, giving existence of $(I - KG)^{-1}$, and achieves stability of $G\,(I - KG)^{-1}$, but not necessarily other closed-loop transfer mappings. Then,*

(i). [12] existence of the controller K, with KG strictly causal, implies the existence of right bounded input bounded output (BIBO) stable factorizations,

$$G = N^* M^{*-1}, \;\; N^* : S^* \rightarrow S_0(R^n), \;\; M^* : S^* \rightarrow S_0(R^m) \tag{2.3.6}$$

where N^* and M^* are BIBO stable and S^* is the factorization space.

(ii). [14] existence of N^* and M^*, as in (i), implies the existence of a right coprime factorization,

$$G = N M^{-1}, \;\; N : S^r \rightarrow S_0(R^n), \;\; M : S^r \rightarrow S_0(R^m) \tag{2.3.7}$$

where N and M are BIBO stable and S^r is the factorization space.

(iii). [14] existence of a right coprime factorization of G over the factorization space, S^r, implies the existence of BIBO stable maps

$$\tilde{V} : S_0(R^m) \rightarrow S^r, \tilde{U} : S_0(R^n) \cap Im(G) \rightarrow S^r \tag{2.3.8}$$

such that the following Bezout identity holds,

$$\tilde{V} M - \tilde{U} N = I : S^r \rightarrow S^r \tag{2.3.9}$$

□

Remark 2.7 This provides a springboard to the use of *lcfs* in stabilizing G, as shall be seen in Section 2.3.2

Remark 2.8 This lemma may be readily dualized in terms of giving right factorizations and a Bezout Identity for the controller K. Combining the dual results gives the following corollary.

Corollary 2.1 *Consider a nonlinear plant $G : S_0(R^m) \to S_0(R^n)$ such that the inverse image of an unbounded element of the range of G is either bounded, or contains no elements which are bounded. Furthermore, suppose that there exists a feedback controller $K : S_0(R^n) \to S_0(R^m)$, as in Fig. 2-1 such that the closed-loop is well posed, giving existence of all the closed-loop transfer mappings, and stability of $G(I-KG)^{-1}$ and $K(I-GK)^{-1}$. Then there will exist right coprime factorizations of G and K as in (2.3.1), (2.3.3). Furthermore there will exist maps $\tilde{V}$, $\tilde{U}$, $\tilde{M}$, $\tilde{N}$ such that the Bezout identities (1.1.16), (1.1.17) hold.* □

Remark 2.9 Note that the maps $\tilde{V}$, $\tilde{U}$, $\tilde{M}$, $\tilde{N}$ referred to in this Corollary are not necessarily left factorizations of the plant and controller.

Thus the relationship from the well-posedness and stability of the system $\{G, K\}$ is established. Further results, more along the lines of the linear results (1.1.28), may also be obtained. The following theorem and lemma show that well-posedness and coprimeness are necessary and sufficient for the existence and stability of the operator inverse.

Theorem 2.1

Given $\{G, K\}$, and $G = NM^{-1}$ and $K = UV^{-1}$ rcfs *as in (2.3.1) and (2.3.3), then $\{G, K\}$ is well-posed iff*

$$\begin{bmatrix} M & -U \\ -N & V \end{bmatrix}^{-1} \text{ exists} \tag{2.3.10}$$

and is internally stable iff

$$\begin{bmatrix} M & -U \\ -N & V \end{bmatrix}^{-1} \text{ is BIBO stable} \tag{2.3.11}$$

□

Proof. First we note that

$$\begin{aligned}\begin{bmatrix} I & -K \\ -G & I \end{bmatrix}^{-1} &= \begin{bmatrix} I & -UV^{-1} \\ -NM^{-1} & I \end{bmatrix}^{-1} \\ &= \left\{\begin{bmatrix} M & -U \\ -N & V \end{bmatrix}\begin{bmatrix} M^{-1} & 0 \\ 0 & V^{-1} \end{bmatrix}\right\}^{-1} \\ &= \begin{bmatrix} M & 0 \\ 0 & V \end{bmatrix}\begin{bmatrix} M & -U \\ -N & V \end{bmatrix}^{-1} \end{aligned} \tag{2.3.12}$$

It is straightforward to see (2.3.10) holds iff $\{G, K\}$ is well-posed.

($\Leftarrow$) Suppose that (2.3.11) holds, then for all a, b bounded we define c, d as follows.

$$\begin{pmatrix} c \\ d \end{pmatrix} = \begin{bmatrix} M & -U \\ -N & V \end{bmatrix}^{-1}\begin{pmatrix} a \\ b \end{pmatrix} \tag{2.3.13}$$

c, d are bounded. Hence, by (2.3.12),

$$\begin{bmatrix} I & -K \\ -G & I \end{bmatrix}^{-1}\begin{pmatrix} a \\ b \end{pmatrix} = \begin{bmatrix} M & 0 \\ 0 & V \end{bmatrix}\begin{pmatrix} c \\ d \end{pmatrix} \tag{2.3.14}$$

Under (2.3.1), (2.3.3) M and V are BIBO stable. Hence Mc and Vd are BIBO thus showing that the system inverse operator exists and is BIBO.

($\Rightarrow$) Suppose that $\{G, K\}$ is well posed and stable and that $G = NM^{-1}$ and $K = UV^{-1}$ are stable *rcfs*. Let

$$\begin{pmatrix} e \\ f \end{pmatrix} = \begin{bmatrix} I & -K \\ -G & I \end{bmatrix}^{-1}\begin{pmatrix} a \\ b \end{pmatrix} \tag{2.3.15}$$

then for all a, b bounded, we have e, f bounded. Define c, d as in (2.3.13), note that as a, b and e, f are bounded, the following equations hold

$$\begin{pmatrix} a \\ b \end{pmatrix} = \begin{bmatrix} Mc - Ud \\ -Nc + Vd \end{bmatrix} \tag{2.3.16}$$

$$\begin{pmatrix} e \\ f \end{pmatrix} = \begin{pmatrix} Mc \\ Vd \end{pmatrix} \tag{2.3.17}$$

As e is bounded Mc is bounded, and since a and Mc are bounded, Ud is bounded. Similarly, as b and f are bounded, Vd and Nc are bounded. By coprimeness of NM^{-1}, since Nc and Mc are both bounded c is bounded. Similarly, by coprimeness of UV^{-1}, d is bounded. This completes the proof. ∎

Hence the stability and well-posedness of the system depends on the existence and stability of the operator $\begin{bmatrix} M & -U \\ -N & V \end{bmatrix}^{-1}$. In fact the relationship is somewhat stronger, coprimeness also results from the stability of this operator.

Lemma 2.3 *Suppose we have $G = NM^{-1}$ and $K = UV^{-1}$, such that the operators M, N, U, V are BIBO stable. Then these are rcfs for G and K if they satisfy (2.3.11)* □

Proof. Since the matrix inverse is stable we require that unbounded inputs yield unbounded inputs. Consider x an unbounded signal, and consider the action of the system as follows.

$$\begin{bmatrix} M & -U \\ -N & V \end{bmatrix}\begin{pmatrix} x \\ 0 \end{pmatrix} = \begin{bmatrix} Mx - U0 \\ -Nx - V0 \end{bmatrix} \tag{2.3.18}$$

As x is unbounded, the output is also unbounded. As V and U are BIBO stable operators, Mx or Nx must be unbounded, giving coprimeness of M, N. Considering the action of $\begin{bmatrix} M & -U \\ -N & V \end{bmatrix}\begin{pmatrix} 0 \\ y \end{pmatrix}$ for y unbounded gives coprimeness of U, V. ∎

Remark 2.10 These results are exactly the same as those obtained in the linear theory, as described by (1.1.29). This generalization to the nonlinear case is straightforward as the principle of superposition is not required in the proof. When left factorizations are considered the principle of superposition is essential to the proof, and so the linear results do not readily generalize.

2.3.2 Left Coprime Factorization Results

In linear systems theory the Bezout identity may be used to check left coprimeness. There does not appear to be a generalization for nonlinear systems.

Note that the definition of left coprimeness we use induces some restrictions on the plant G. In discrete time, consider plants $G : S_0(R^m) \to S_0(R^n)$ such that the inverse image of an unbounded element of the range of G is either bounded, or contains no

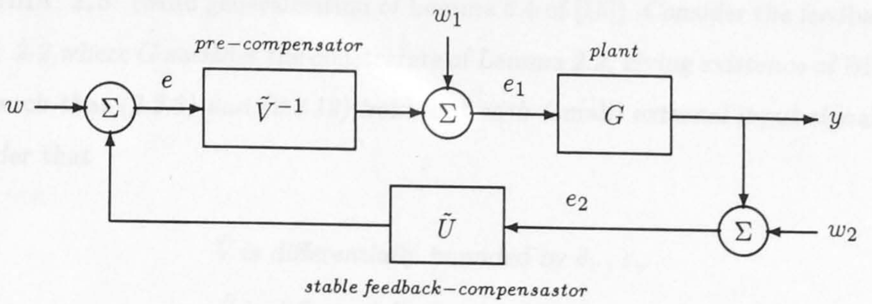

Figure 2-2: Feedback system with external inputs.

elements which are bounded. It is shown in Lemma 3.1 of [49] that under this assumption, G will have a *lcf*. Furthermore, if this condition is violated it can be seen that for any left factorization, either $\tilde{N}$ is not BIBO stable, or $\tilde{M}^{-1}\tilde{N}$ is not a *lcf*. Hence we shall only consider plants G such that this assumption holds.

Left coprime factorizations started out being considered in conjunction with the Bezout identity. By considering feedback systems for injective nonlinear plants with a particular pre-compensator $\tilde{V}^{-1}$ and feedback-compensator $\tilde{U}$, as in Fig. 2-2, Hammer [15] uses the Bezout identity to obtain a method of stabilization.

Lemma 2.4 [From [15]] *Consider a nonlinear plant $G : S_0(R^m) \to S_0(R^n)$, suppose that there exists a feedback controller $K : S_0(R^n) \to S_0(R^m)$, as in Fig. 2-1 such that the closed-loop is well posed. Further, consider that there is a right coprime factorization for the plant, $G = NM^{-1}$, which satisfies a Bezout identity as in Lemma 2.2, (2.3.9). Then the feedback system shown in Fig. 2-2, in the case w_1, $w_2 = 0$, is stable, and the closed-loop transfer mappings as defined under (2.3.9) are*

$$e = \tilde{V}Mw\,,\; e_1 = Mw\,,\; y = Nw \tag{2.3.19}$$

□

However this stability is not robust to small signal injections around the loop, so the resulting closed loop system is not necessarily internally stable. To cope with such small signals, Hammer introduces a differential boundedness constraint on $\tilde{V}$ and $\tilde{U}$. Differential boundedness is defined as follows.

Definition 2.8 [Differential Boundedness] *An operator $F: C(\mathcal{X}) \mapsto C(\mathcal{Y})$ is said to be differentially bounded by θ_F, ε_F iff for all signals a_1, $a_2 \in C(\mathcal{X})$, if $|a_1 - a_2| < \varepsilon_F$ then $|Fa_1 - Fa_2| < \theta_F$.*

Lemma 2.5 (Mild generalization of Lemma 3.4 of [15]) *Consider the feedback system of Fig. 2-2,where G satisfies the constraints of Lemma 2.2, giving existence of BIBO stable $\tilde{V}$, $\tilde{U}$ such that (2.3.9) and (2.3.19) hold, but with (small) external input signals w_1, w_2. Consider that*

$$\tilde{V} \text{ is differentially bounded by } \theta_V, \varepsilon_V \tag{2.3.20}$$

$$\tilde{U} \text{ is differentially bounded by } \theta_U, \varepsilon_U \tag{2.3.21}$$

In addition, consider that N is stable over $S_0(\theta^n)$, where $\theta > \theta_U + \theta_V$. Then the system is internally (bounded-input) stable for $w \in S_0([\theta - \theta_U - \theta_V]^m)$, $w_1 \in S_0(\varepsilon_V^m)$, $w_2 \in S_0(\varepsilon_U^n)$, in that under these constraints all signals are bounded for all possible inputs, or equivalently all the closed-loop transfer mappings are BIBO stable. □

Proof. First consider the case when $w_1 = w_2 = 0$. Then for $w \in S_0(\theta^m)$ we have all internal signals bounded. The transfer mappings of Fig. 2-2 are given implicitly, via (2.3.19) (2.3.9), in

$$e = (I - \tilde{U}G\tilde{V})^{-1}w = \tilde{V}Mw\,, \quad e_1 = \tilde{V}^{-1}e = Mw\,, \quad y = e_2 = Ge_1 = Nw \tag{2.3.22}$$

These are all BIBO stable by Lemma 2.2. Consider now the effect of adding in the small signal $w_2 \in S_0(\varepsilon_U^n)$ with $w_1 = 0$. Then the response at e will be given by

$$e = w + \tilde{U}(w_2 + y) \tag{2.3.23}$$

Define the mapping $\alpha : S_0(\varepsilon_U^n) \to S$ by

$$\alpha(w_2) = \tilde{U}(w_2 + y) - \tilde{U}y \tag{2.3.24}$$

Since $\tilde{U}$ is differentially bounded by θ_U and $w_2 \in S_0(\varepsilon_U^n)$, we have $\alpha(w_2) \in S_0(\theta_U^m)$. Note that the response at e when $w_2 \neq 0$ is the same as if we replace the input signal w with $w + \alpha(w_2)$ and set $w_2 = 0$. Hence we conclude that for $w \in S_0([\theta - \theta_U]^m)$ the introduction of $w_2 \in S_0(\varepsilon_U^n)$ does not affect the boundedness of the signals e , e_1 and y. The signal e_2 will remain bounded as it is the sum of two bounded signals.

Consider now the effect of adding in the small signal $w_1 \in S_0(\theta_V^m)$ and, without loss of generality, as shown above we can take $w_2 = 0$. The response of e_1 will be given by

$$e_1 = w_1 + \tilde{V}^{-1}(w + \tilde{U}e_2) \tag{2.3.25}$$

Define the mapping $\beta : S_0(\varepsilon_V^m) \rightarrow S$ by

$$\beta(w_1) = \tilde{V}[\tilde{V}^{-1}(w + \tilde{U}e_2) + w_1] - \tilde{V}[\tilde{V}^{-1}(w + \tilde{U}e_2)] \tag{2.3.26}$$

Since $\tilde{V}$ is differentially bounded and $w_1 \,\epsilon\, S_0(\epsilon_V^m)$, we have $\beta(w_1) \,\epsilon\, S_0(\theta_V^m)$. If we replace the input w by

$$w + \beta(w_1) = \tilde{V}[\tilde{V}^{-1}(w + \tilde{U}e_2) + w_1] - \tilde{U}e_2 \tag{2.3.27}$$

and set the input at w_1 zero, then it is straightforward to show that the output e_1 is unchanged. Consequently, e_1 is bounded, as then are e, e_2 and y. Likewise, with the input $w \,\epsilon\, S_0([\theta - \theta_U - \theta_V]^m)$ the effects of both $w_1 \,\epsilon\, S_0(\theta_V^m)$ and $w_2 \,\epsilon\, S_0(\theta_v^m)$ can be incorporated into the input signal, under the differential boundedness assumptions on $\tilde{V}, \tilde{U}$. This gives us the result. ∎

Remark 2.11 When using this lemma in the development of the main results of the following chapter, N is taken to be BIBO stable, and we are able to choose $0 < \theta < \infty$ arbitrarily large, so that w is effectively unrestricted.

Remark 2.12 In the linear case $\tilde{U}$, $\tilde{V}$ are differentially bounded by all θ, and $\varepsilon_U \propto \theta$, $\varepsilon_V \propto \theta$. As a consequence the closed-loop system is internally stable, without restriction on the inputs w, w_1, w_2.

Remark 2.13 By considering that in Fig. 2-2 we have $w = 0$, then we can construct a controller $K = \tilde{V}^{-1}\tilde{U}$ which will bounded-input stabilize the plant G. This is more precisely stated in the following corollary.

Corollary 2.2 *Consider the feedback system of Fig. 2-2,where G satisfies the constraints of Lemma 2.2, giving existence of BIBO stable $\tilde{V}$, $\tilde{U}$ such that (2.3.9) and (2.3.19) hold. Further, assume that N is BIBO stable, and that (2.3.20) and (2.3.21) hold, and that the signal $w = 0$. Construct a controller by $K = \tilde{V}^{-1}\tilde{U}$, then the system $\{G, K\}$ thus formed is ε_V, ε_U bounded-input stable.* □

Remark 2.14 If a dual approach to Lemma 2.2 we find a *rcf* of K, which satisfies a

Bezout identity,

$$\tilde{M}V - \tilde{N}U = I \tag{2.3.28}$$

then we can construct a stabilized plant $G = \tilde{M}^{-1}\tilde{N}$. Further, if the *lcf* of G is differentially bounded as follows

$$\tilde{M} \text{ is differentially bounded by } \theta_M,\, \varepsilon_U \tag{2.3.29}$$

$$\tilde{N} \text{ is differentially bounded by } \theta_U,\, \varepsilon_V \tag{2.3.30}$$

and then the system $\{G,\, K\}$ will be stable in the presence of inputs $w_1 \epsilon S_0(\varepsilon_V^n)$ and $w_2 \epsilon S_0(\varepsilon_U^m)$. Equivalently there is ε_V, ε_U bounded-input stability of the system $\{G,\, K\}$.

Thus it is demonstrated that given a Bezout identity and differential boundedness of some of the nonlinear operators, a limited form of stability may be proven. In the linear case, and for nonlinear right coprime factorizations, Theorem 2.1 and Lemma 2.3, it was shown that matrix versions of this result are possible. We are thus motivated to derive similar results for nonlinear left coprime factorizations. There are problems in trying to generalize these results. The relationship between the operator $\begin{bmatrix} \tilde{V} & -\tilde{U} \\ -\tilde{N} & \tilde{M} \end{bmatrix}^{-1}$ and the stability and well-posedness of $\{G,\, K\}$ does not generalize directly from the linear case. In the linear case we have that this operator is stable iff the system $\{G,\, K\}$ is well-posed and stable, as stated in (1.1.28). The first result attainable for a matrix of nonlinear operators of this form is as follows.

Lemma 2.6 [41] *Consider the system $\{G,\, K\}$, where G and K are such that each has stable left coprime factorizations as give in (2.3.2), (2.3.4). Consider the system of Fig. 2-3, with inputs w_1, w_2 zero. Then this system will be well-posed if*

$$\begin{bmatrix} \tilde{V} & -\tilde{U} \\ -\tilde{N} & \tilde{M} \end{bmatrix}^{-1} \textit{ exists.} \tag{2.3.31}$$

in the sense that the inputs and outputs for each of $\tilde{N}$, $\tilde{M}^{-1}$, $\tilde{U}$, $\tilde{V}^{-1}$ will be well-defined. Furthermore this system will be stable in the sense that the inputs and outputs for each of $\tilde{N}$, $\tilde{M}^{-1}$, $\tilde{U}$, $\tilde{V}^{-1}$ will be bounded if s_1 and s_2 are bounded iff

$$\begin{bmatrix} \tilde{V} & -\tilde{U} \\ -\tilde{N} & \tilde{M} \end{bmatrix}^{-1} \textit{ is stable.} \tag{2.3.32}$$

□

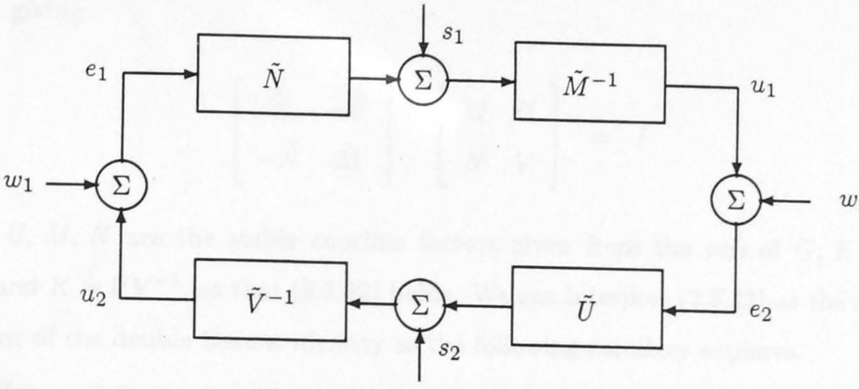

Figure 2-3: System of the *lcfs* of G and K.

Proof. From Fig 2-3,

$$u_1 = \tilde{V}^{-1}(s_2 + \tilde{U}u_2) \tag{2.3.33}$$

$$u_2 = \tilde{M}^{-1}(s_1 + \tilde{N}u_1) \tag{2.3.34}$$

Using simple algebraic manipulations, then under the existence assumption (2.3.32),

$$\begin{pmatrix} u_1 \\ u_2 \end{pmatrix} = \begin{bmatrix} \tilde{M} & -\tilde{N} \\ -\tilde{U} & \tilde{V} \end{bmatrix}^{-1} \begin{pmatrix} s_1 \\ s_2 \end{pmatrix} \tag{2.3.35}$$

This mapping is BIBO stable under (2.3.32), so that u_1, u_2 are bounded if s_1, s_2 are bounded. Furthermore u_1, u_2 bounded gives $e_1 = \tilde{M}u_2$ and $e_2 = \tilde{V}u_1$ both bounded. Hence the result. ∎

Remark 2.15 Note that this result is similar to applying Lemma 2.3 and then Theorem 2.1 to the system $\{\tilde{N}\tilde{V}^{-1}, \tilde{U}\tilde{M}^{-1}\}$.

Remark 2.16 This assumption forms the basis of the following theory. In the following chapter it is found that this assumption allows the characterization of the class of all stabilizing plants and controllers.

Remark 2.17 Further to Remark 2.14, note that if $\tilde{V}$ or $\tilde{N}$ is differentially bounded as in (2.3.20), (2.3.30), respectively, and $\tilde{M}$ or $\tilde{U}$ is differentially bounded as in (2.3.20), (2.3.30), respectively, then $\{G, K\}$ will be ε_V, ε_U bounded-input stable.

Remark 2.18 The assumption (2.3.32) does not seem overly restrictive when considered in the context of the linear theory. In the linear case we have the double Bezout equations holding, giving

$$\begin{bmatrix} \tilde{V} & -\tilde{U} \\ -\tilde{N} & \tilde{M} \end{bmatrix}^{-1} \begin{bmatrix} M & U \\ N & V \end{bmatrix} = I$$

Here V, U, M, N are the stable coprime factors given from the *rcfs* of G, K as $G = NM^{-1}$ and $K = UV^{-1}$, so that (2.3.32) holds. We can interpret (2.3.32) as the nonlinear equivalent of the double Bezout identity as the following corollary explores.

Corollary 2.3 *Consider a plant, G, and controller, K such that each has a lcf, and the conditions of Lemma 2.6 are satisfied. Suppose that the system of Fig 2-3 is well-posed, so that (2.3.31) holds, then it is necessary for the operators $(\tilde{V} - \tilde{U}G)^{-1}$ and $(\tilde{M} - \tilde{N}K)^{-1}$ to exist. Further, there exist right factorizations for G and K as in (2.3.1), (2.3.3), with M, N, U, V not necessarily stable, and the following Bezout Identities hold.*

$$\tilde{V}M - \tilde{U}N = I \tag{2.3.36}$$

$$\tilde{M}V - \tilde{N}U = I \tag{2.3.37}$$

Further if the system is stable so that (2.3.32) holds, the right factorizations of G and K will be stable and the factorizations will thus be coprime. □

Proof. Consider the action of the $\begin{bmatrix} \tilde{V} & -\tilde{U} \\ -\tilde{N} & \tilde{M} \end{bmatrix}$ on the vector $\begin{pmatrix} Ga \\ a \end{pmatrix}$.

$$\begin{bmatrix} \tilde{V} & -\tilde{U} \\ -\tilde{N} & \tilde{M} \end{bmatrix} \begin{pmatrix} Ga \\ a \end{pmatrix} = \begin{pmatrix} \tilde{M}\tilde{M}^{-1}\tilde{N}a - \tilde{N}a \\ -\tilde{U}Ga + \tilde{V}a \end{pmatrix}$$
$$= \begin{pmatrix} 0 \\ (\tilde{V} - \tilde{U}G)a \end{pmatrix}$$
$$\begin{pmatrix} 0 \\ b \end{pmatrix} \tag{2.3.38}$$

Hence under (2.3.31) it is necessary that this be invertible, giving $a = (\tilde{V} - \tilde{U}G)^{-1}b$. Hence it is necessary that $(\tilde{V} - \tilde{U}G)^{-1}$ exists.

Now define the mappings M and N as follows.

$$M = (\tilde{V} - \tilde{U}G)^{-1},\ N = G(\tilde{V} - \tilde{U}G)^{-1} \tag{2.3.39}$$

The Bezout identity (2.3.36) may now be simply proved.

$$\begin{aligned}\tilde{V}M - \tilde{U}N &= \tilde{V}(\tilde{V} - \tilde{U}G)^{-1} - \tilde{U}G(\tilde{V} - \tilde{U}G)^{-1} \\ &= (\tilde{V} - \tilde{U}G)(\tilde{V} - \tilde{U}G)^{-1} \\ &= I\end{aligned}$$

Considering the action of $\begin{bmatrix} \tilde{V} & -\tilde{U} \\ -\tilde{N} & \tilde{M} \end{bmatrix}$ on $\begin{pmatrix} a \\ Ka \end{pmatrix}$ gives the dual results of the existence of $(\tilde{M} - \tilde{N}K)^{-1}$, a right factorization for K

$$V = (\tilde{M} - \tilde{N}K)^{-1},\ U = K(\tilde{M} - \tilde{N}K)^{-1} \tag{2.3.40}$$

and that (2.3.37) holds.

Consider now the result of (2.3.32) holding. Note that

$$\begin{pmatrix} Ga \\ a \end{pmatrix} = \begin{pmatrix} Nb \\ Mb \end{pmatrix} = \begin{bmatrix} \tilde{V} & -\tilde{U} \\ -\tilde{N} & \tilde{M} \end{bmatrix}^{-1} \begin{pmatrix} 0 \\ b \end{pmatrix} \tag{2.3.41}$$

Hence if b is bounded Nb and Mb are both bounded and so M and N are BIBO stable. Further if b is unbounded we have Nb or Mb unbounded, and since $G = NM^{-1}$ this gives a *rcf* for G.

Dually, under (2.3.32), the right factorization of K, (2.3.40), will be coprime. ■

Remark 2.19 The condition (2.3.32) is a stronger one than merely the satisfaction of the double Bezout identities (1.1.18). The additional strength appears to be necessary to deal with both the signals s_1 and s_2 as in Fig. 2-3, rather than just s_1 or s_2 acting alone.

More general results are elusive, although under other assumptions other results may be obtained. In the case that there exist *lcfs* for G and K in which the operators $\tilde{V}$, $\tilde{M}$ are linear, the following result will hold.

Lemma 2.7 *Suppose that for G and K, we have* lcfs *as in (2.3.2), (2.3.4), with $\tilde{V}$ and $\tilde{M}$ linear. Then $\{G, K\}$ is well-posed iff (2.3.31) holds, and is stable if (2.3.32) holds.* □

Proof. Note that with $\tilde{V}$, $\tilde{M}$ linear the following will hold.

$$\begin{bmatrix} I & -K \\ -G & I \end{bmatrix}^{-1} = \begin{bmatrix} I & -\tilde{V}^{-1}\tilde{U} \\ -\tilde{M}^{-1}\tilde{N} & I \end{bmatrix}^{-1} \tag{2.3.42}$$

$$= \left[\begin{bmatrix} \tilde{V}^{-1} & 0 \\ 0 & \tilde{M}^{-1} \end{bmatrix}\begin{bmatrix} \tilde{V} & -\tilde{U} \\ -\tilde{N} & \tilde{M} \end{bmatrix}\right]^{-1} \tag{2.3.43}$$

$$= \begin{bmatrix} \tilde{V} & -\tilde{U} \\ -\tilde{N} & \tilde{M} \end{bmatrix}^{-1}\begin{bmatrix} \tilde{V} & 0 \\ 0 & \tilde{M} \end{bmatrix} \tag{2.3.44}$$

It is straightforward to see that $\{G,\, K\}$ is well-posed iff (2.3.31) holds. As $\tilde{V}$ and $\tilde{M}$ are BIBO stable we have that $\begin{bmatrix} I & -K \\ -G & I \end{bmatrix}^{-1}$ is stable if $\begin{bmatrix} \tilde{V} & -\tilde{U} \\ -\tilde{N} & \tilde{M} \end{bmatrix}^{-1}$ is stable. ∎

Remark 2.20 Currently there does not appear to be any way to link stability and well-posedness of $\{G,\, K\}$ to equations (2.3.31), (2.3.32) without this linearity assumption.

2.4 Conclusion

In this chapter we have introduced the stability definition for our work and attempted to develop a clear picture of why coprime factorizations are important in nonlinear systems.

The definition of right coprimeness is shown to link closely with the existence of a generalized Bezout identity, and give a simple characterization for the well-posedness and stability of a system $\{G,\, K\}$. Furthermore, given a well posed system, it is possible to find a right factorization for the plant or controller, with a corresponding Bezout identity. It is interesting to note that given a right factorization of G, a bounded-input stabilizing controller may be constructed through the Bezout identity $\tilde{V}M - \tilde{U}N = I$.

Left coprime factorizations do not give the same straightforward characterizations as *rcfs*, but when combined with differential boundedness, prove to have close links with the bounded-input stability of the $\{G,\, K\}$. Additionally the use of left factorizations allows the injection of signals within the plant and controller, as in Fig. 2-3, which do not interfere with the stability of the system. This proves to be crucial in characterizing the classes of stabilizing plant and controller pairs, as seen in the following chapter.

This brings us to the point where the ideas developed in forming the Youla-Kucera parameterization for linear systems, as in Section 1.1.3, may now be generalized to nonlinear systems.

Chapter 3

Youla-Kucera parameterization for Nonlinear Systems

3.1 Introduction

In this chapter we develop the Youla-Kucera parameterization for nonlinear systems, generalizing the results of Section 1.1.3. Most earlier work assumes linearity in the plant or controller, or applies only to systems with a certain structure. Here we attempt to develop a general parameterization which provides for the generalization of the existing linear results to the nonlinear arena.

In his work Hammer [12, 14] derives a stabilization scheme for injective nonlinear plants having right coprime factorizations. This is achieved through the construction of a pre- and feedback-compensator pair such that a Bezout identity is satisfied. Further work done by Tay and Moore [49] shows that for a wider class of systems the same procedure can be followed and the class of all stabilizing pre- and feedback-compensators satisfying the Bezout identity can be constructed. Through the introduction of the concept of differential boundedness Hammer [15] shows how to derive internal stability results for such a system. Paice [40] was able to combine these results to derive a class of controllers which bounded-input stabilize a given plant. In particular, the characterizations are such that the bounded-input, bounded-output stable system parameter can be realized in a single feedback loop, as in the linear theory of [8]. This work was further developed in Paice [41] to generate a result giving the classes of all plants stabilized by a given controller. Note that these papers worked mainly with the left factorizations of a given plant, controller pair, and worked within a purely input-output framework.

In other work Desoer [6] and then Verma [55] have developed an approach based on the right coprime factorizations of the plant and controller in an input-output framework. However, in order to construct the class of controllers stabilizing a given plant in a manner similar to that of the Youla-Kucera parametrization, it is necessary that linearity be assumed for the plant. By taking a left coprime approach to the problem the need of assuming linearity is avoided, however differential boundedness assumptions become necessary.

In this chapter we build on the results of the previous chapter to develop a nonlinear version of the Youla-Kucera parameterization for nonlinear systems. By considering the left coprime factorization of G, it is shown that a class of stabilizing pre- and feedback-compensator pairs can be constructed. It is shown that the class of pre- and feedback-compensator pairs each parametrized by BIBO stable maps Q generates a class of feedback controllers for G. Further it is shown that this class can be generated by a single BIBO stable map Q_r, which can be calculated in terms of the original map Q. It is then shown that a necessary and sufficient condition on Q_r for the system to be bounded-input stable, under certain differential boundedness conditions on factorizations of G and K, is that Q_r is BIBO stable. Serendipitously , the differential boundedness assumptions do not involve Q_r.

These results may be readily dualized to give a parameterization for the class of plants G_{S_r}, stabilized by a given controller. This new structure, which only requires single maps S_r and Q_r to parameterize these classes of plants, allows a simple characterization of the class of systems $\{G_{S_r}, K_{Q_r}\}$ which are stable. Some robust stabilization results follow.

3.2 A Class of Stabilizing Controllers for G

Recall the connections between *rcfs* and the Bezout identity from Section 2.3.1. Specifically, in Lemma 2.2, p. 20 and Lemma 2.4, p. 24 it is shown that a plant with a right coprime factorization may be stabilized by a pre- and feedback-compensator pair. This results from the Bezout identity (2.3.9). It is now shown how the techniques of Section 1.1.3 may be applied to give the class of controllers K_Q which stabilize a given plant.

The following theorem is based on Theorem 3.1 of Tay [49], and has been generalized using the techniques presented in Hammer [15].

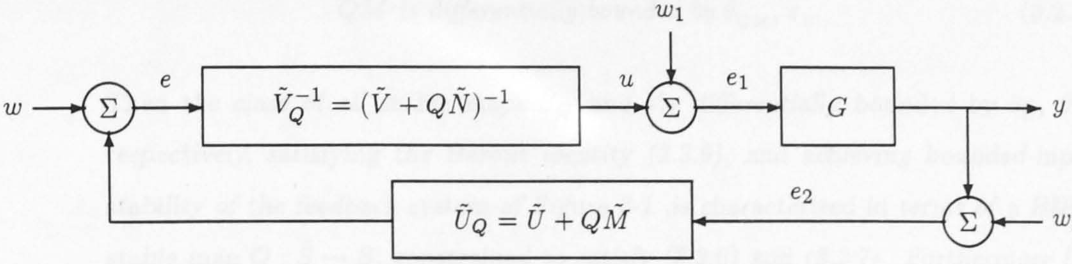
w_1
w
Σ
e
$\tilde{V}_Q^{-1} = (\tilde{V} + Q\tilde{N})^{-1}$
u
Σ
e_1
G
y
e_2
$\tilde{U}_Q = \tilde{U} + Q\tilde{M}$
Σ
w_2

Figure 3-1: The class of all bounded-input stabilizers of G.

Theorem 3.1 (Mild generalization of Theorem 3.1 of [49]) *Consider a nonlinear plant $G : S_0(R^m) \to S_0(R^n)$, satisfying the assumptions of Lemma 2.2, with right and left coprime factorizations, $G = NM^{-1} = \tilde{M}^{-1}\tilde{N}$ over the factorization spaces $S, \tilde{S}$. Consider also BIBO stable mappings*

$$\tilde{V} : S_0(R^m) \to S, \textit{ invertible}, \;\; \tilde{U} : S_0(R^n) \to S \tag{3.2.1}$$

such that the feedback system shown in Figure 2-2, p. 24 has stable transfer mappings of (2.3.19). Then

(i). [49] the class of all stable maps $\tilde{V}_Q$, $\tilde{U}_Q$ satisfying

$$\tilde{V}_Q M - \tilde{U}_Q N = I \tag{3.2.2}$$

is characterized in terms of an arbitrary BIBO stable nonlinear map $Q : \tilde{S} \to S$ as

$$\tilde{U}_Q \;=\; (\tilde{U} + Q\tilde{M}) : S_0(R^n) \to S \tag{3.2.3}$$

$$\tilde{V}_Q \;=\; (\tilde{V} + Q\tilde{N}) : S_0(R^m) \to S \tag{3.2.4}$$

Moreover, the feedback system of Figure 3-1 for the case w_1 , $w_2 = 0$ is well-posed and has stable input-output transfer mappings given from

$$e \,=\, \tilde{V}_Q M w \,,\, e_1 \,=\, Mw \,,\, y \,=\, Nw \tag{3.2.5}$$

(ii). (Generalization of (i)) *Moreover, consider that $\tilde{U}$ and $\tilde{V}$ satisfy the differential boundedness constraints of (2.3.20), (2.3.21) and $\tilde{M}$ and $\tilde{N}$ are such that there exist BIBO stable maps $Q : \tilde{S} \to S$ achieving*

$$Q\tilde{N} \textit{ is differentially bounded by } \theta_{QN}, \varepsilon_V \tag{3.2.6}$$

$$Q\tilde{M} \textit{ is differentially bounded by } \theta_{QM}, \varepsilon_U \qquad (3.2.7)$$

Then the class of all stable maps $\tilde{U}_Q$ and $\tilde{V}_Q$ differentially bounded by θ_U, θ_V, respectively, satisfying the Bezout identity (2.3.9), and achieving bounded-input stability of the feedback system of Figure 3-1 ,is characterized in terms of a BIBO stable map $Q : \tilde{S} \rightarrow S$, constrained to satisfy (3.2.6) and (3.2.7). Furthermore $\tilde{U}_Q$ and $\tilde{V}_Q$ are given by (2.16).

(iii). If the system of Figure 3-1 is to be structurally stable then, whether or not (3.2.6) and (3.2.7) holds, it is necessary that Q be BIBO stable. [By structural stability we mean that the mappings $\tilde{V}_{Q_V}$, $\tilde{U}_{Q_U}$ will bounded-input stabilize the system for arbitrary Q_U, Q_V in some "small" neighbourhood of Q, without the constraint $Q_U = Q_V$.]

□

Proof. See [49] for a proof of (i).

Proof of (ii). Suppose Q is BIBO stable and makes $Q\tilde{M}$ and $Q\tilde{N}$ differentially bounded, as above, then $\tilde{U}_Q$ and $\tilde{V}_Q$, given by (2.15) will be differentially bounded by θ_U and θ_V, respectively. Substituting $\tilde{U}_Q$ and $\tilde{V}_Q$ into (2.3.9) shows that they satisfy the Bezout identity, hence the closed-loop transfer mappings given by (3.2.5) will be stable. Applying Lemma 2.5 shows that $\tilde{U}_Q$ and $\tilde{V}_Q$ bounded-input stabilize the system.

Now suppose that $\tilde{U}^*$ and $\tilde{V}^*$ are differentially bounded by θ_U and θ_V, respectively, and satisfy (2.3.9), stabilizing the system. Then as both they and $\tilde{U}$, $\tilde{V}$ satisfy (2.3.9) we get

$$(\tilde{V}^* - \tilde{V})M = (\tilde{U}^* - \tilde{U})N \qquad (3.2.8)$$

Now define Q by the equation

$$Q\tilde{M} = \tilde{U}^* - \tilde{U} \qquad (3.2.9)$$

which is differentially bounded by θ_U. Substituting into (3.2.8) gives

$$(\tilde{V}^* - \tilde{V})M = Q\tilde{M}N = Q\tilde{N}M\,, \ \ \tilde{V}^* - \tilde{V} = Q\tilde{N} \qquad (3.2.10)$$

which is differentially bounded by θ_V under (3.2.10). Note that (3.2.9) is in the form of (3.2.3) and (3.2.10) is of the form of (3.2.4), and so we have the required result.

Proof of (iii). Suppose that the system of Figure 3-1 is structurally stable and that Q is unstable. Then for unstable Q_U, Q_V in the neighbourhood of Q the system is stable, and e_2, u are bounded. Note that since Q_U, Q_V are unstable $e = Q_U\tilde{M}e_2 - Q_V\tilde{N}u + w$ is bounded only if $(Q_U\tilde{M}e_2 - Q_V\tilde{N}u)$ is bounded. This condition generically fails for Q_U, Q_V pairs in the neighbourhood of Q, and the result obtained follows. ■

Remark 3.1 In the linear case, the conditions requiring differential boundedness evanesce, as do the restrictions on the magnitudes of the inputs w_1, w_2.

Remark 3.2 The differential boundedness conditions (3.2.6) and (3.2.7) appear to be overly restrictive, however we are unable to give sufficiency of Q BIBO without it. This motivates, to some extent, the work of the next section.

Remark 3.3 Referring to result (iii), when Q_U and Q_V are unstable and, $Q_U = Q_V$, then it appears difficult to show that $(Q_U\tilde{M}e_2 - Q_V\tilde{N}u)$ is bounded for all possible u, e_2 bounded. Of course in the linear case, where superposition holds, this situation is excluded by well-posedness assumptions.

Remark 3.4 Applying Corollary 2.2, p. 26, to Theorem 3.1 and assuming $w = 0$ gives the Youla-Kucera parametrization for a class of stabilizing controllers for a linear plant G. This is more precisely stated in the following lemma.

Lemma 3.1 *Consider a possibly noninjective plant G with right and left coprime factorizations as in (2.3.1), (2.3.2). Suppose that there exist mappings $\tilde{V}$, $\tilde{U}$ which are differentially bounded as in (2.3.20), (2.3.21), respectively, and the Bezout identity (2.3.9) holds, leading to a controller class K_Q, constructed as in Figure 3-2 with $w = 0$, where Q is a BIBO stable mapping, and given by*

$$K_Q = \tilde{V}_Q^{-1}\tilde{U}_Q = (\tilde{V} + Q\tilde{N})^{-1}(\tilde{U} + Q\tilde{M}) \tag{3.2.11}$$

Then the system $\{G,\, K_Q\}$ will be ε_V, ε_U bounded-input stable when Q is a BIBO stable mapping constrained so that (3.2.6) and (3.2.7) are satisfied. □

Remark 3.5 Note that this is a sufficient, but not necessary, condition. It may be possible for a mapping Q which does not satisfy (3.2.6) or (3.2.7) to bounded-input stabilize the system $\{G,\, K_Q\}$.

Remark 3.6 The map from Q is well defined, but the map from K_Q to Q is not, so it is difficult to use this lemma to generate a stability result.

Remark 3.7 Note that in this lemma we have assumed that $w = 0$. As the stability is based on the Bezout identity (2.3.9), if $w \neq 0$ the stability of the system will not be

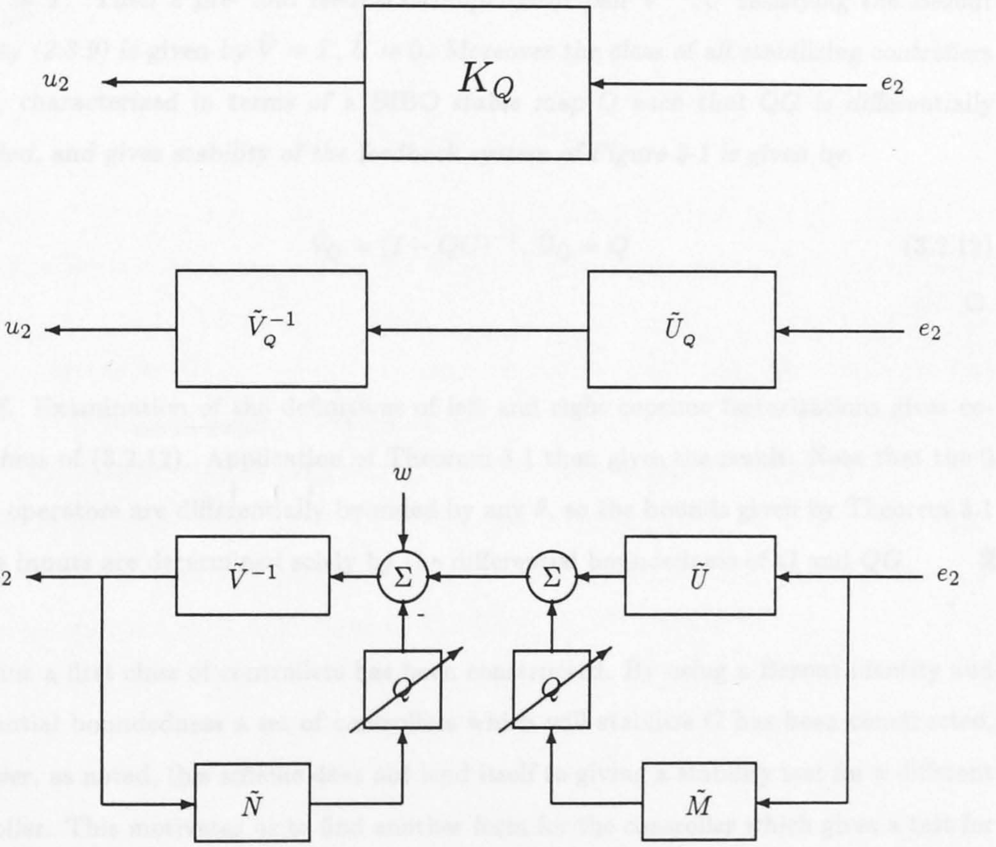

Figure 3-2: The controller K_Q.

disturbed.

In the sequel it will be assumed that $w = 0$, however in general this is not necessary, and is assumed so as to rearrange the feedback system of Figure 2-2, p. 24, to that of Figure 2-1, p. 16.

The following corollary to Theorem 3.1, giving a class of stabilizing pre- and feedback-compensator pairs for a stable plant, will be useful in later sections.

Corollary 3.1 *Consider that the conditions of Theorem 3.1 apply, and in addition G is stable, with right and left coprime factorization pairs $N = G$, $M = I$ and $\tilde{N} = G$, $\tilde{M} = I$. Then a pre- and feedback-compensator pair $\tilde{V}^{-1}, \tilde{U}$ satisfying the Bezout identity (2.3.9) is given by $\tilde{V} = I$, $\tilde{U} = 0$. Moreover the class of all stabilizing controllers for G, characterized in terms of a BIBO stable map Q such that QG is differentially bounded, and gives stability of the feedback system of Figure 3-1 is given by*

$$\tilde{V}_Q = (I - QG)^{-1},\ \tilde{U}_Q = Q \tag{3.2.12}$$

□

Proof. Examination of the definitions of left and right coprime factorizations gives coprimeness of (3.2.12). Application of Theorem 3.1 then gives the result. Note that the 0 and I operators are differentially bounded by any θ, so the bounds given by Theorem 3.1 on the inputs are determined solely by the differential boundedness of Q and QG. ∎

Thus a first class of controllers has been constructed. By using a Bezout identity and differential boundedness a set of controllers which will stabilize G has been constructed. However, as noted, this scheme does not lend itself to giving a stability test for a different controller. This motivates us to find another form for the controller which gives a test for the stability of a system when a different controller is used.

3.3 A Second Class of Stabilizing Controllers for G

Consider again the class of stabilizing controllers for a nonlinear plant G which satisfies the conditions of Theorem 3.1. This gives a system with the structure of the system $\{G,\ K_Q\}$, where K_Q is as in Figure 3-2. This class of feedback controllers K_Q stabilizing G is characterized in terms of a BIBO stable mapping Q, restricted as in Theorem 3.1. In the linear case, the principle of superposition applies to allow re-configuration of the

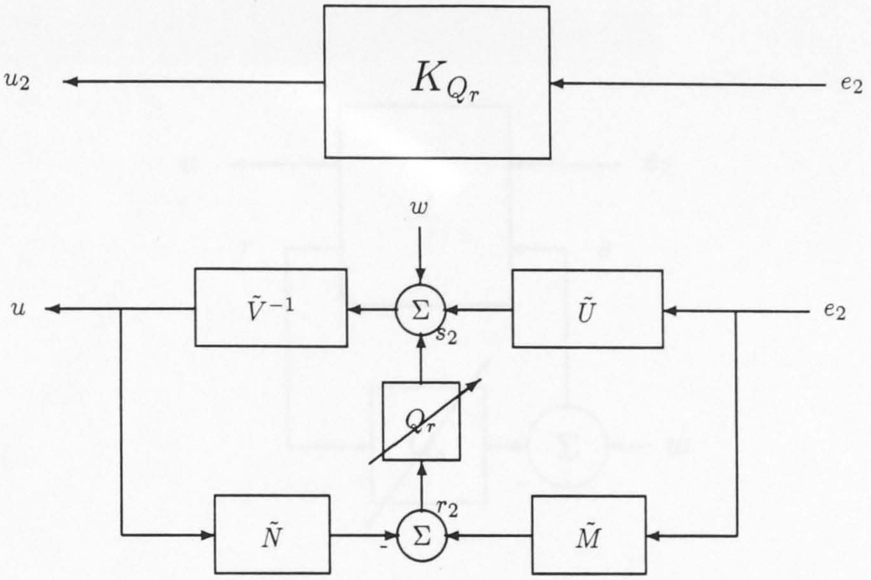

Figure 3-3: The controller K_{Q_r}.

controller of Figure 3-2, now denoted K_{Q_r}, as in Figure 3-3, where $Q_r = Q$. Notice that the controller class of Figure 3-3 has the form of Figure 3-4 for some operator J, whereas the arrangement of Figure 3-2 does not.

Our purpose in this section is to examine for the nonlinear case, where superposition does not hold, the controller class K_{Q_r} of Figure 3-3 and 3-4, parametrized in terms of Q_r. Is K_{Q_r} stabilizing for arbitrary stable Q_r? Is there some stable Q_r such that $K_Q = K_{Q_r}$ for arbitrary stable Q? In other words, is there a natural generalization to the linear results where the class of all stabilizing controllers can be conveniently parametrized as in Figure 3-4 with the block Q implemented in a single feedback loop?

To proceed, let us note that for the controller shown in Figure 3-3,

$$u = \tilde{V}^{-1}(\tilde{U}e_2 + Q_r(\tilde{M}e_2 - \tilde{N}u))$$

Since $u = K_Q e_2$ we substitute for u and rearrange to get

$$Q_r = (\tilde{V}K_Q - \tilde{U})(\tilde{M} - \tilde{N}K_Q)^{-1} \tag{3.3.1}$$

However, from (3.2.11), we have that $(\tilde{V}K_Q - \tilde{U}) = Q\tilde{M} - Q\tilde{N}K_Q$, so that substitution into (3.3.1) gives

$$Q_r = (Q\tilde{M} - Q\tilde{N}K_Q)(\tilde{M} - \tilde{N}K_Q)^{-1} \tag{3.3.2}$$

and the following lemma is established.

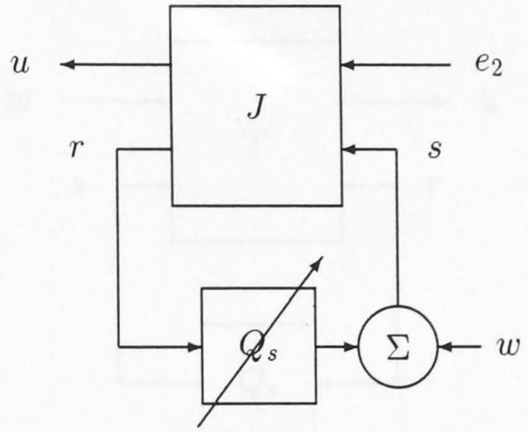

Figure 3-4: Reconfiguration of the controller K_{Q_r}.

Lemma 3.2 *Consider a nonlinear plant $G = \tilde{M}^{-1}\tilde{N}$ bounded-input internally stabilized by the controller class K_Q of (3.2.11), Figure 3-2, under the conditions of Theorem 3.1, with $w = 0$. Then for each Q, there exists a nonlinear mapping Q_r such that*

$$K_{Q_r} = K_Q \tag{3.3.3}$$

Further, Q_r is given by equations (3.3.1) and (3.3.2). □

Remark 3.8 Note that from a comparison of Figure 3-2 and Figure 3-3 it is straightforward to conclude that Q_r is linear if and only if Q is linear, and in this case $Q_r = Q$. Moreover, in the case where all operators are linear and $Q_r = Q$, then the controller classes of Figure 3-2, 3-3 and 3-4 are equivalent with J defined from

$$\begin{pmatrix} u \\ r \end{pmatrix} = \begin{bmatrix} K & \tilde{V}^{-1} \\ \tilde{M}(I - GK) & -\tilde{N}\tilde{V}^{-1} \end{bmatrix} \begin{pmatrix} e_2 \\ s \end{pmatrix} \tag{3.3.4}$$

Remark 3.9 When $(\tilde{M} - \tilde{N}K_Q)^{-1}$ is BIBO stable it may be shown that Q BIBO stable implies Q_r BIBO stable. In the linear case this condition is trivially satisfied as $Q = Q_r$, however it is not clear whether this result carries over to the general nonlinear case. Thus we cannot currently guarantee stability of Q_r when given stability of Q.

Remark 3.10 In the case $w \neq 0$ the controllers K_Q and K_{Q_r} of Figure 3-2, 3-3 will bounded-input stabilize the system, although there is no general relationship between Q_r

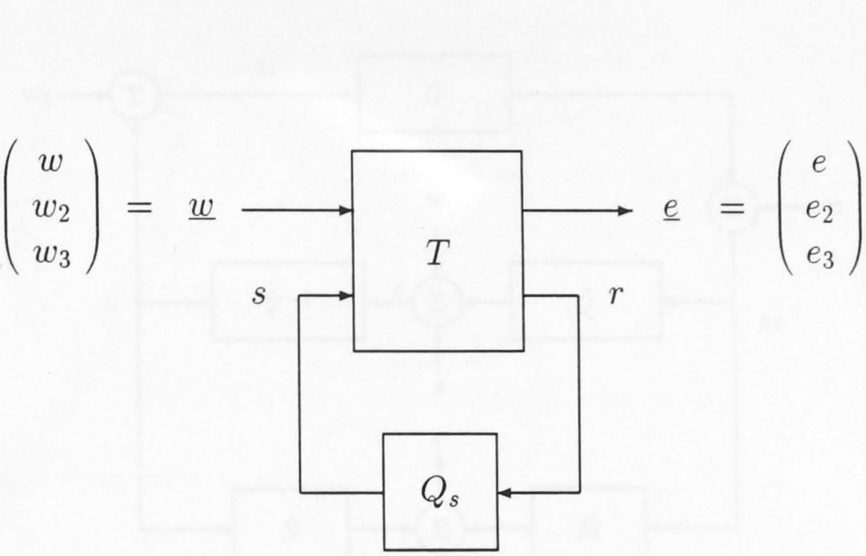

Figure 3-5: Reconfiguration of the system $\{G,\, K_{Q_r}\}$.

and Q which gives $K_{Q_r} = K_Q$. Conditions on Q_r giving bounded-input stability of the system are yet to be derived.

Motivated by the linear results we now look for conditions on Q_r to achieve bounded-input internal stability of the closed loop system with plant G and controller K_{Q_r}. Lemma 3.2 shows that when $w = 0$ the class of bounded-input stabilizing controllers for G may be parametrized in terms of a single Q_r. This allows us to restructure the nonlinear system of Figure 3-1 into that of Figure 3-4 and 3-5, where $\underline{e} = [e, e_1, e_2]'$ and $\underline{w} = [w, w_1, w_2]'$. In this case we can obtain an expression for J in terms of the composition of two nonlinear operators. This may be seen from the examination of the following

$$\begin{aligned}\begin{pmatrix} w \\ r \end{pmatrix} &= \begin{pmatrix} \tilde{V}^{-1}(s + \tilde{U}e_2) \\ \tilde{M}e_2 - \tilde{N}\tilde{V}^{-1}(s + \tilde{U}e_2) \end{pmatrix} \\ &= \begin{bmatrix} 0 & \tilde{V}^{-1} \\ \tilde{M} & -\tilde{N}\tilde{V}^{-1} \end{bmatrix} \circ \begin{bmatrix} I & 0 \\ \tilde{U} & I \end{bmatrix} \begin{pmatrix} e_1 \\ s \end{pmatrix} \end{aligned} \tag{3.3.5}$$

Where $\circ$ denotes composition of operators.

We now look for conditions on Q_r that will give stability of the system. By studying this structure, and using Corollary 3.1 the following result is derived.

Lemma 3.3 *Consider the feedback system of Figure 3-5, or equivalently Figure 3-6 with $s = Q_r s$, where G, $\tilde{U}$ and $\tilde{V}$ satisfy the conditions of Theorem 3.1. Also consider that s is bounded, w_1, w_2 are bounded by ε_U and ε_V, respectively, and w is bounded. Then*

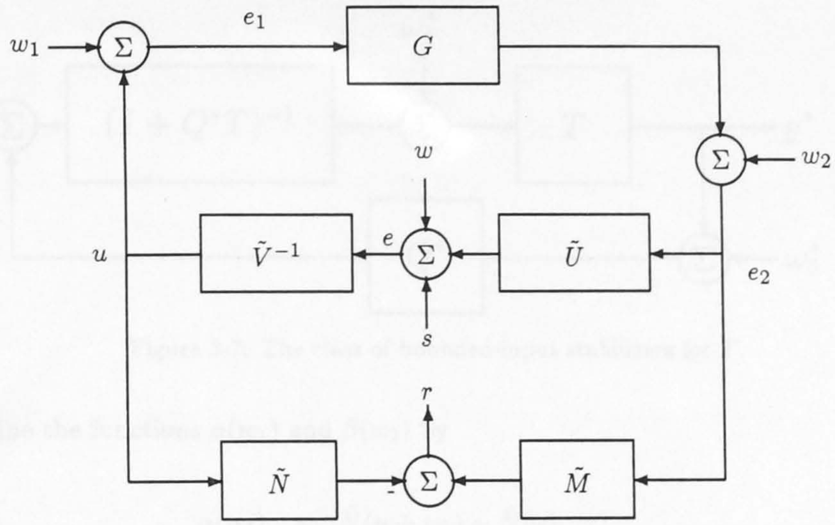

Figure 3-6: Structure of the operator T.

(i). the mapping

$$
\begin{aligned}
T &: S_0(\theta^m) \times S_0(\varepsilon_U^m) \times S_0(\varepsilon_V^n) \to S_0(R^m) \times S_0(R^m) \times S_0(R^n) \times S_0(R^m) \\
T &: (w, w_1, w_2) \mapsto (e, e_1, e_2, r) \quad [\underline{w} \mapsto (\underline{e}, r)] \qquad (3.3.6)
\end{aligned}
$$

is BIBO stable.

(ii). Moreover, if $\tilde{M}$ and $\tilde{N}$ are differentially bounded, as in (2.3.29), (2.3.30), with $\mid w_1 \mid < \varepsilon_V$ and $\mid w_2 \mid < \varepsilon_U$, then r is bounded by $\theta_M + \theta_N$ □

Proof. (i) The subsystem of T with inputs $(s, \underline{w})$ and outputs $\underline{e}$ is itself a re-organisation of the scheme of Figure 2-2, p. 24, where the input w of Figure 2-2, p. 24 is replaced by $s + w$. Thus under the conditions of the lemma, by Theorem 3.1 the outputs $\underline{e}$ will be bounded.

Now $\tilde{M}$ is BIBO stable, hence $\tilde{M}e_2$ is bounded. Also $\tilde{V}^{-1}e = e_1 - w_1$, hence $\tilde{V}^{-1}e$ is bounded, and since $\tilde{N}$ is BIBO stable $\tilde{N}\tilde{V}^{-1}e$ is bounded. Consequently $r = \tilde{M}e_2 - \tilde{N}\tilde{V}^{-1}e$ is bounded. Hence for inputs $(s, \underline{w})$ bounded as given in the lemma, the outputs $(r, \underline{e})$ are bounded, giving the result, (i).

(ii) Referring to Figure 3-6, clearly r can be expressed as

$$
\begin{aligned}
r &= \tilde{M}e_2 - \tilde{N}u \\
&= \tilde{M}(w_2 + G(w_1 + u) - \tilde{N}u \qquad (3.3.7)
\end{aligned}
$$

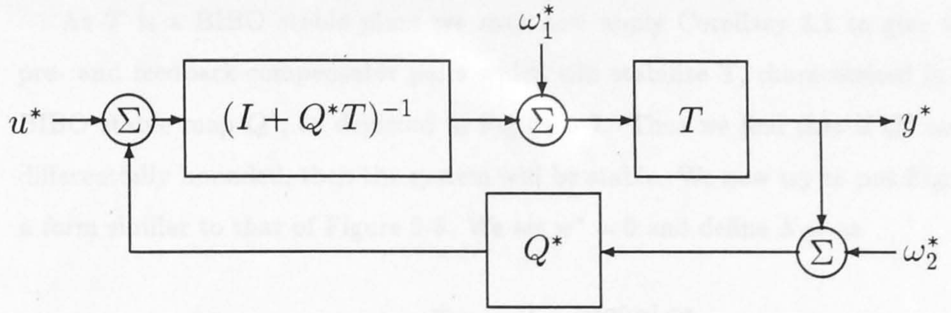

Figure 3-7: The class of bounded-input stabilizers for T

Now define the functions $\alpha(w_1)$ and $\beta(w_2)$ by

$$\begin{aligned} \alpha(w_1) &= \tilde{N}(u+w_1) - \tilde{N}(u) && (3.3.8) \\ \beta(w_2) &= \tilde{M}(w_2 + \tilde{M}^{-1}b) - \tilde{M}(\tilde{M}^{-1}b) && (3.3.9) \end{aligned}$$

where $b = \tilde{N}u + \alpha(w_1)$. Since $\tilde{N}$, $\tilde{M}$ are differentially bounded by θ_V,θ_U respectively, then $\alpha(w_1)$ and $\beta(w_2)$ are also bounded by θ_N, θ_M. Further, (3.3.7) can be rewritten as

$$\begin{aligned} r &= \tilde{M}(w_2 + \tilde{M}^{-1}(\tilde{N}u + \alpha(w_1))) - \tilde{N}u \\ &= \tilde{M}(\tilde{M}^{-1}(\tilde{N}u + \alpha(w_1) + \beta(w_2)) - \tilde{N}u \\ &= \alpha(w_1) + \beta(w_2) && (3.3.10) \end{aligned}$$

Since $\alpha(w_1)$ and $\beta(w_2)$ are bounded by θ_N and θ_M, respectively, r is bounded by $\theta_N + \theta_M$. This completes the proof. ∎

Remark 3.11 Note that we are assuming N is BIBO, so the assumption that s be bounded may be dropped as noted Remark 2.11.

Remark 3.12 In the case $w_1 = w_2 = 0$ we have $r \equiv 0$. When w_1 and w_2 are not zero, but suitably small, we have r non-zero, but bounded by $\theta_N + \theta_M$. The value of r will, in general, depend on the value of s, but it will remain bounded for all input signals s. In the linear case, the terms of $\alpha(w_1)$ and $\beta(w_2)$ depend on s, but $r = \alpha(w_1) + \beta(w_2)$ does not, giving the result $T_{22} = 0$. The bound on r that we have obtained here, depending on w_1, w_2 and s is the nonlinear version of the result $T_{22} = 0$.

Remark 3.13 Note that we have not assumed $w = 0$ in this lemma. This is due to the fact that since $\tilde{N}$ is BIBO stable, the boundedness of the system will be invariant of arbitrary inputs prior to the pre-compensator.

As T is a BIBO stable plant we may now apply Corollary 3.1 to give the class of pre- and feedback-compensator pairs which will stabilize T, characterized in terms of a BIBO stable map Q^*, as depicted in Figure 3-7. Thus we find that if Q^* and Q^*T are differentially bounded, then the system will be stable. We now try to put Figure 3-7 into a form similar to that of Figure 3-5. We set $w^* = 0$ and define K_{Q^*} as

$$K_{Q^*} = (I + Q^*T)^{-1}Q^* \tag{3.3.11}$$

note that if we set $w_1^* = (\underline{w}, r)$, $w_2^* = 0$ and constrain K_{Q^*} to be of the form

$$K_{Q^*} : \begin{pmatrix} \underline{e} \\ r \end{pmatrix} \mapsto \begin{pmatrix} \underline{w} \end{pmatrix} = \begin{pmatrix} Q_r r \\ 0 \\ 0 \end{pmatrix} \tag{3.3.12}$$

we have put the system into a form similar to Figure 3-5. We now find a Q^* which satisfies this constraint.

Lemma 3.4 *A Q^* satisfying* (3.3.12) *is*

$$Q^* = \begin{pmatrix} Q_r r \\ 0 \\ 0 \end{pmatrix} \tag{3.3.13}$$

□

Proof. We give a proof by substitution. For the lemma to hold we must have

$$\begin{aligned} \begin{pmatrix} Q_r r \\ 0 \\ 0 \end{pmatrix} &= (I + Q^*T)^{-1}Q^* \begin{pmatrix} \underline{e} \\ r \end{pmatrix} \\ (I + Q^*T) \begin{pmatrix} Q_r r \\ 0 \\ 0 \end{pmatrix} &= Q^* \begin{pmatrix} \underline{e} \\ r \end{pmatrix} \qquad (3.3.14) \\ &= \begin{pmatrix} Q_r r \\ 0 \\ 0 \end{pmatrix} \end{aligned}$$

Recalling Remark 3.12 we have

$$T\begin{pmatrix} Q_r r \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} \underline{e} \\ 0 \\ 0 \end{pmatrix}$$

Therefore

$$Q^{*}T\begin{pmatrix} Q_r r \\ 0 \\ 0 \end{pmatrix} = 0 \tag{3.3.15}$$

Substituting this into (3.3.14) completes the proof. ∎

Remark 3.14 The most important result from this lemma is that the pre-compensator $\tilde{V}_{Q^{*}}^{-1}$ is always equivalent to the identity when there is no input between $\tilde{V}_{Q^{*}}^{-1}$ and $\tilde{U}_{Q^{*}}$. This would seem to indicate that in the case depicted in Figure 3-5 we need only require Q_r stable and differentially bounded to give stability of the system. Even this is a stronger condition than required, as is now explored.

Theorem 3.2

Consider the system of Figure 3-5, where the operators $\tilde{N}, \tilde{M}, \tilde{V}, \tilde{N}$ are all differentially bounded as given by (2.3.20), (2.3.21), (2.3.29) and (2.3.30) respectively, and

$$w_1 \textit{ and } w_2 \textit{ bounded by } \varepsilon_V,\ \varepsilon_U, \textit{ respectively.} \tag{3.3.16}$$

and $w = 0$. The closed-loop system is bounded-input stable iff the operator Q_r is BIBO stable for all inputs r bounded by $\theta_M + \theta_N$. i.e. *the system $\{G, K_{Q_r}\}$ will be ε_V, ε_U bounded-input stable iff Q_r is $(\theta_M + \theta_N)$ bounded-input stable.* □

Proof. By Lemma 3.3 the conditions of the theorem give the result that for s bounded the outputs $(\underline{e}, r)$ of the system are bounded. Due to the restrictions on the inputs w_1 and w_2 given by(3.3.16) the value of the output r is bounded by $\theta_M + \theta_N$. If Q_r is stable for all inputs r bounded by $\theta_N + \theta_M$ then the value of θ will be well defined, where θ is given by

$$\theta = \sup_{|x| < \theta_M + \theta_N} |\, Q_r x \,| \tag{3.3.17}$$

Therefore for all inputs, $\underline{w}$, bounded as above s will be bounded by θ. Hence the outputs will be bounded, and the closed-loop system is bounded-input stable.

If Q_r is unstable, then for some r, $s = Q_r r$ will be unbounded. If the signals e, e_1 and e_2 remain bounded the system would be stable. Suppose that e_2 is bounded, then since $\tilde{U}$ is BIBO stable $\tilde{U}e_2$ is bounded, therefore $e = s + \tilde{U}e_2$ is unbounded. Furthermore as $\tilde{V}$ is BIBO, if $\tilde{V}^{-1}$ has an unbounded input it will have an unbounded output, so e will be unbounded. Now suppose that $e = s + \tilde{U}e_2$ is bounded, then $\tilde{U}e_2$ is unbounded, and as $\tilde{U}$ is BIBO, this implies that the signal e_2 is unbounded. We have shown that if the signal s is unbounded then one of the signals e, e_1 and e_2 must also be unbounded. Therefore the system is unstable. This gives us the result. ∎

Remark 3.15 Notice that in this theorem the differential boundedness assumption 3.2.7 is absent, so that in this respect the characterization of this section are more elegant than those in the Youla-Kucera formulation of the previous section.

Remark 3.16 The introduction of an arbitrary bounded signal w will not disturb stability of the system. This follows since N is BIBO stable, and using arguments from Remarks 3.11-3.13.

Remark 3.17 In the case that the mappings $\tilde{V}$, $\tilde{U}$, $\tilde{N}$ and $\tilde{M}$ satisfy a Lipschitz condition instead of satisfying the differential boundedness constraints, the theorem again holds, although the bounds on the inputs w_1, w_2 will be different. Proof details on this result are straightforward, following closely the above proof, and are therefore omitted.

Remark 3.18 This result specializes directly to known linear results, since in the linear case the bounds on w_1, w_2 may be arbitrarily large.

Remark 3.19 Further to Remark 3.9, following Lemma 3.2, we may now show that BIBO stability of Q implies Q_r is BIBO stable. When Q is BIBO stable, then K_Q will bounded-input stabilize G, and by Lemma 3.2 the Q_r given by (3.3) will ensure $K_{Q_r} = K_Q$. Hence K_{Q_r} will bounded-input stabilize G, and so Q_r will stabilize T. Application of the theorem gives BIBO stability of Q_r.

3.4 The Class of Stabilized Plants

In the previous section the class of all bounded-input stabilizers K_{Q_r}, for a nonlinear plant G was characterized. Here we find the dual result which characterizes the class of all plants which are bounded-input stabilized by a given nonlinear controller K, and thereby achieve a first robust stabilization result. In the next section a more general robust stabilization

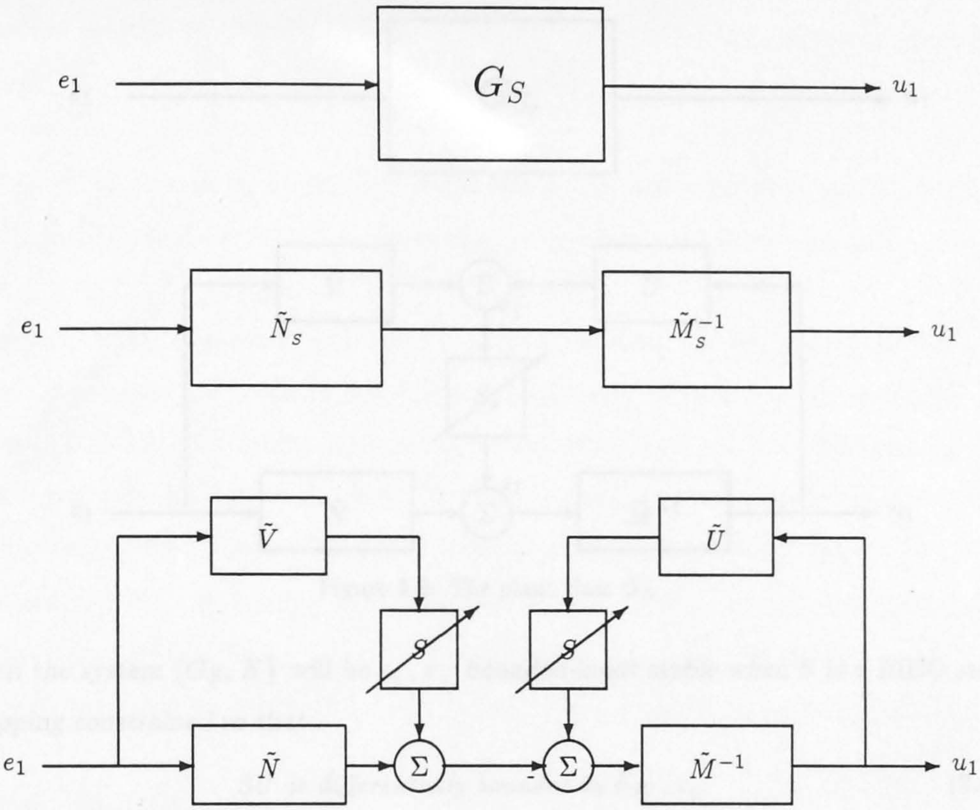

Figure 3-8: The plant G_S.

result is developed.

The dual procedure to constructing the class of stabilizers K_Q of Figure 3-2, is followed to produce the class of all plants bounded-input stabilized by a given controller. Suppose that $K : S_0(R^m) \rightarrow S_0(R^n)$ has right and left coprime factorizations, as in (2.3.3), (2.3.4), and that the following Bezout identity holds,

$$\tilde{M}V - \tilde{N}U = \tilde{Z}, \textit{ unimodular} \tag{3.4.1}$$

with $G = \tilde{M}^{-1}\tilde{N}$. Then dualizing Lemma 3.1 we have.

Lemma 3.5 *Consider an ε_V, ε_U bounded-input stable system $\{G, K\}$, such that G has a lcf, as given by (2.3.2), which is differentially bounded, as in (2.3.29), (2.3.30), and K has both right and left coprime factorizations, as in (2.3.3) and (2.3.4). Suppose further that the Bezout identity (3.4.1) holds, leading to a class of plants G_S, constructed as in Figure 3-8, where S is a BIBO stable mapping, and G_S is given by*

$$G_S = \tilde{M}_S^{-1}\tilde{N}_S = (\tilde{M} + S\tilde{U})^{-1}(\tilde{N} + S\tilde{V}) \tag{3.4.2}$$

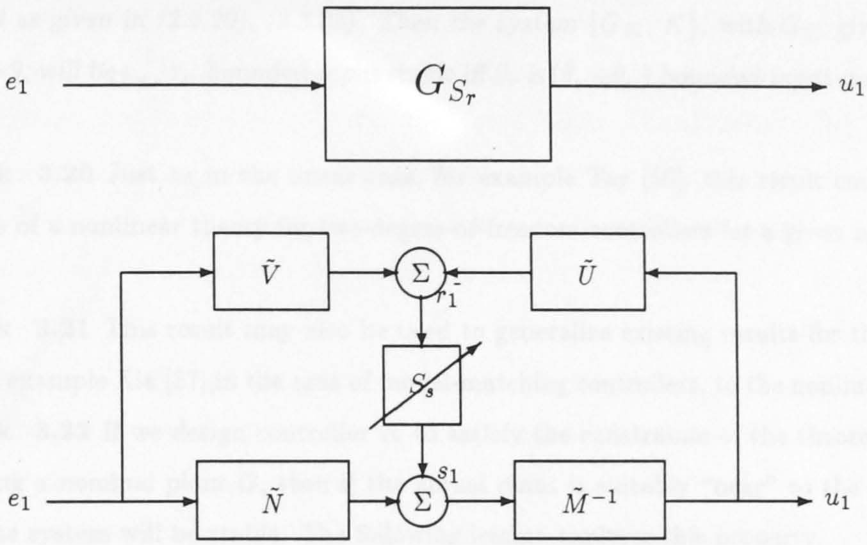

Figure 3-9: The plant class G_{S_r}.

Then the system $\{G_S,\, K\}$ will be ε_V, ε_U bounded-input stable when S is a BIBO stable mapping constrained so that

$$S\tilde{U} \text{ is differentially bounded by } \theta_{SU},\, \varepsilon_U \tag{3.4.3}$$

$$S\tilde{V} \text{ is differentially bounded by } \theta_{SV},\, \varepsilon_V \tag{3.4.4}$$

□

In order to encompass a wider class of plants stabilized by the controller K, we dualize the results of Lemma 3.2, and Theorem 3.2, thus constructing the class of plants G_{S_r} as shown in Figure 3-9.

Lemma 3.6 *For every BIBO stable S such that (3.4.3) and (3.4.4) hold, there exists a stable S_r such that the controllers of Figure 3-8 and 3-9 are equivalent, in that $G_{S_r} = G_S$. Furthermore, S_r is given by*

$$\begin{aligned} S_r &= (\tilde{M}G_S - \tilde{N})(\tilde{V} - \tilde{U}G_S)^{-1} & (3.4.5)\\ &= (S\tilde{V} - S\tilde{U}G_S)(\tilde{V} - \tilde{U}G_S)^{-1} & (3.4.6)\end{aligned}$$

□

Theorem 3.3

Consider an ε_V, ε_U bounded-input stable system $\{G,\, K\}$, such that G has a lcf, as given by (2.3.2), which is differentially bounded, as in (2.3.29), (2.3.30), and K has both right and left coprime factorizations, as in (2.3.3) and (2.3.4), with the lcf being differentially *bounded as given in (2.3.20), (2.3.21). Then the system $\{G_{S_r}, K\}$, with G_{S_r} given as in Figure 3-9, will be ε_V, ε_U bounded-input stable iff S_r is $(\theta_V+\theta_U)$ bounded-input stable.* □

Remark 3.20 Just as in the linear case, for example Tay [50], this result could form the basis of a nonlinear theory for two-degree-of-freedom controllers for a given nonlinear plant.

Remark 3.21 This result may also be used to generalize existing results for the linear case, for example Xia [37],in the area of model-matching controllers, to the nonlinear case.

Remark 3.22 If we design controller K to satisfy the constraints of the theorem when stabilizing a nominal plant G, then if the actual plant is suitably "near" to the nominal plant, the system will be stable. The following lemma explores this property.

Lemma 3.7 *Consider that the conditions of Theorem 3.3 hold and that the difference between G_S and G is "small", in the sense that $\mid (G_S-G)u \mid< \varepsilon_U$ for all inputs $u \,\epsilon\, S_0(R^M)$. Then S_r given by (3.4.5) is BIBO stable, moreover, all outputs of S_r are bounded by θ_U.* □

Proof. First note that (3.4.5) can be rewritten as follows,

$$\begin{aligned} S_r &= (\tilde{M}G_S - \tilde{N})(\tilde{V} - \tilde{U}G_S)^{-1} \\ &= (\tilde{M}(G + (G_S - G)) - \tilde{M}(G))(\tilde{V} - \tilde{U}G_S)^{-1} \end{aligned} \tag{3.4.7}$$

Now define the mapping $\alpha : S_0(R^M) \rightarrow S_0(R^n)$ as follows,

$$\alpha(u) \;=\; \tilde{M}(G + \Delta G)u - \tilde{M}(G)u \tag{3.4.8}$$

Under the differential boundedness assumption on $\tilde{M}$, (2.3.29), note that if $\Delta G u < \varepsilon_U$, then $\alpha(u) < \varepsilon_U$ for all inputs u. Setting $\Delta G \equiv G_S - G$, then the conditions of the lemma give the required restriction, so that the lemma is proved. ∎

3.5 Stability of $\{G_S,\, K_Q\}$

In this section the results of the previous sections are generalized to obtain a more complete robust stabilization result. In the notation of the previous sections, we show that under an appropriate double Bezout condition, K_{Q_r} "stabilizes" G_{S_r} iff Q_r "stabilizes" S_r. Thus when $S_r \equiv 0$, the result specializes to that of Section 3.2, and when $Q_r \equiv 0$, the results specialize to those of Section 3.4. In adaptive control, for example, when the plant is uncertain or changing, then an adaptive operator Q_r in the otherwise non-adaptive controller will stabilize the system iff Q_r "stabilizes" S_r. The stability result also is useful in coping with controller uncertainties, or implementation artifacts in the presence of plant uncertainties.

We follow an approach similar to that taken by Verma in [55], in considering the stability of the inverse of a matrix of nonlinear mappings as the basis of a stability result. In his work Verma considered a matrix consisting of the *rcfs* of the plant, G, and controller, K. Here, the dual approach is presented, in that we first consider the stability of a matrix constructed from the *lcfs* of G and K.

Recall the results of Lemma 2.6, which gives stability of the system of Figure 2-3. Consequently, under (2.3.32) we can achieve stability results for the system $\{G_{S_r}, K_{Q_r}\}$ of Figure 3.5, as follows.

Theorem 3.4

Consider the system $\{G_{S_r}, K_{Q_r}\}$ of Figure 3.5, where the maps $\tilde{N}$, $\tilde{M}$, $\tilde{U}$, $\tilde{V}$ are lcfs of G and K, and satisfy (2.3.32), (2.3.20), (2.3.21), (2.3.29) and (2.3.30). Then the system is ε_V, ε_U bounded-input stable iff the system $\{S_r, Q_r\}$ of Figure 3.5 is $(\theta_U + \theta_V)$, $(\theta_M + \theta_N)$ bounded-input stable. □

Proof. Under the conditions of the theorem first apply Lemma 2.6 to give boundedness of the outputs e_1, e_2 and y_1, y_2 when s_1, s_2 are bounded, and the signals w_1, w_2 are bounded by ε_1, ε_2 respectively. Hence the system will be stable iff s_1, s_2 are bounded. Now the boundedness of s_1, s_2 is dependent on the mappings S_r, Q_r and their inputs r_1, r_2, so let us next consider the response of the signals r_1, r_2 to the inputs s_1, s_2 and w_1, w_2.

$$r_1 = \tilde{V}(w_1 + \tilde{V}^{-1}e_2) - \tilde{U}\tilde{M}^{-1}e_1 \tag{3.5.1}$$

$$r_2 = \tilde{M}(w_2 + \tilde{M}^{-1}e_1) - \tilde{N}\tilde{V}^{-1}e_2 \tag{3.5.2}$$

$$e_1 = s_1 + \tilde{N}(w_1 + \tilde{V}^{-1}e_2) \tag{3.5.3}$$

$$e_2 = s_2 + \tilde{U}(w_2 + \tilde{M}^{-1}e_1) \tag{3.5.4}$$

In order to take advantage of the differential boundedness properties of $\tilde{N}$, $\tilde{M}$, $\tilde{U}$, $\tilde{V}$ we

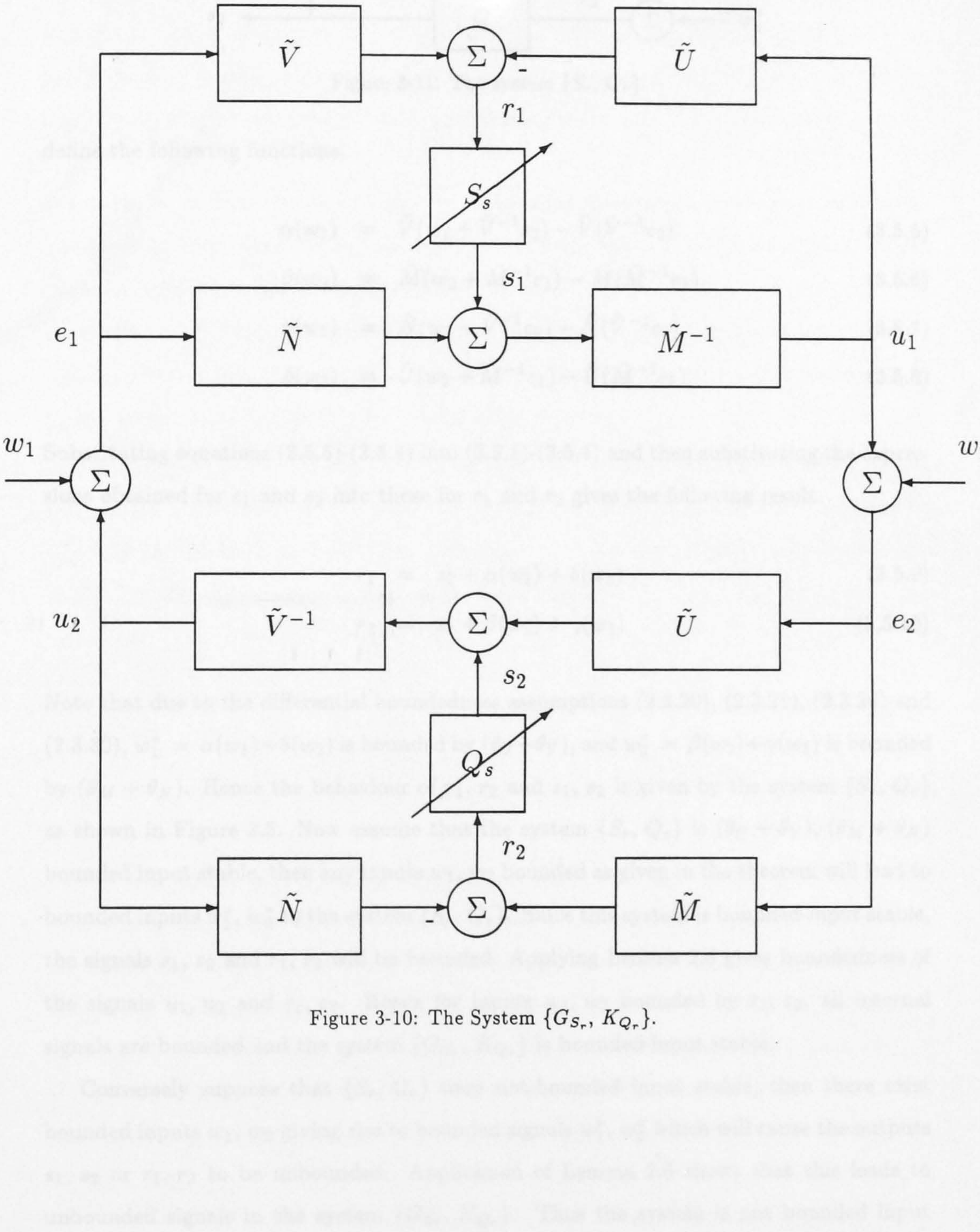

Figure 3-10: The System $\{G_{S_r}, K_{Q_r}\}$.

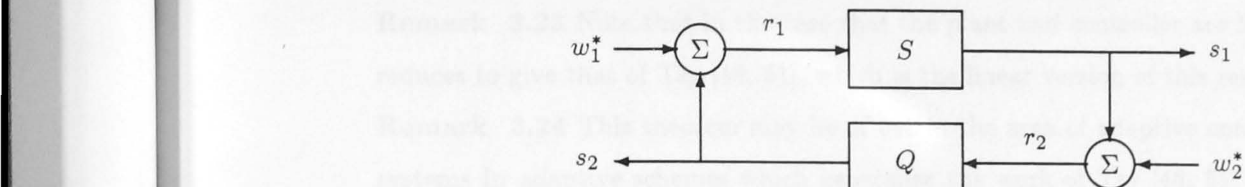

Figure 3-11: The system $\{S_r, Q_r\}$.

define the following functions.

$$\begin{aligned}
\alpha(w_1) &= \tilde{V}(w_1 + \tilde{V}^{-1}e_2) - \tilde{V}(\tilde{V}^{-1}e_2) && (3.5.5)\\
\beta(w_2) &= \tilde{M}(w_2 + \tilde{M}^{-1}e_1) - \tilde{M}(\tilde{M}^{-1}e_1) && (3.5.6)\\
\gamma(w_1) &= \tilde{N}(w_1 + \tilde{V}^{-1}e_2) - \tilde{N}(\tilde{V}^{-1}e_2) && (3.5.7)\\
\delta(w_2) &= \tilde{U}(w_2 + \tilde{M}^{-1}e_1) - \tilde{U}(\tilde{M}^{-1}e_1) && (3.5.8)
\end{aligned}$$

Substituting equations (3.5.5)-(3.5.8) into (3.5.1)-(3.5.4) and then substituting the expressions obtained for e_1 and e_2 into those for r_1 and r_2 gives the following result.

$$\begin{aligned}
r_1 &= s_2 + \alpha(w_1) + \delta(w_2) && (3.5.9)\\
r_2 &= s_1 + \beta(w_2) + \gamma(w_1) && (3.5.10)
\end{aligned}$$

Note that due to the differential boundedness assumptions (2.3.20), (2.3.21), (2.3.29) and (2.3.30), $w_1^* = \alpha(w_1)+\delta(w_2)$ is bounded by $(\theta_U+\theta_V)$, and $w_2^* = \beta(w_2)+\gamma(w_1)$ is bounded by $(\theta_M + \theta_N)$. Hence the behaviour of r_1, r_2 and s_1, s_2 is given by the system $\{S_r, Q_r\}$ as shown in Figure 3.5. Now assume that the system $\{S_r, Q_r\}$ is $(\theta_U + \theta_V)$, $(\theta_M + \theta_N)$ bounded input stable, then any inputs w_1, w_2 bounded as given in the theorem will lead to bounded inputs w_1^*, w_2^* to the system $\{S_r, Q_r\}$. Since this system is bounded-input stable, the signals s_1, s_2 and r_1, r_2 will be bounded. Applying Lemma 2.6 gives boundedness of the signals u_1, u_2 and e_1, e_2. Hence for inputs w_1, w_2 bounded by ε_1, ε_2, all internal signals are bounded and the system $\{G_{S_r}, K_{Q_r}\}$ is bounded-input stable.

Conversely suppose that $\{S_r, Q_r\}$ were not-bounded input stable, then there exist bounded inputs w_1, w_2 giving rise to bounded signals w_1^*, w_2^* which will cause the outputs s_1, s_2 or r_1, r_2 to be unbounded. Application of Lemma 2.6 shows that this leads to unbounded signals in the system $\{G_{S_r}, K_{Q_r}\}$. Thus the system is not bounded input stable and there is a contradiction. ∎

Remark 3.23 Note that in the case that the plant and controller are linear, this result reduces to give that of Tay [48, 51], which is the linear version of this result.

Remark 3.24 This theorem may be of use in the area of adaptive control of nonlinear systems In adaptive schemes which generalize the work of Tay [48, 51, 47] to nonlinear plants, then it is reasonable that Q_r be an adaptive operator. Stability analysis of such adaptive Q_r schemes are then possible, in that there is stability if $Q_r$stabilizes the operator S_r.

Remark 3.25 Note that in this chapter we have considered robust stabilization from an input-output framework, so that although care must be taken of initial conditions, we can allow for time-variations of the plant and controller.

Remark 3.26 This result may be used to produce a link with the problem of simultaneously stabilizing m+1 nonlinear plants with the problem of strongly stabilizing m nonlinear plants, as is explored by the following corollary.

Corollary 3.2 *Consider the system $\{G_0, K_0\}$, which is bounded-input stable and satisfies the assumptions of Theorem 3.4. Then the problem of finding a single controller K_Q that will stabilize the m+1 plants $G_0, G_1, \ldots . G_m$ is equivalent to that of finding a single controller Q for each member of the set of m plants $S_1, S_2, \ldots S_m$, which are given as follows*

$$S_i = (\tilde{M}G_i - \tilde{N})(\tilde{V} - \tilde{U}G_i)^{-1} \tag{3.5.11}$$

Where $\tilde{V}$, $\tilde{U}$, $\tilde{M}$, $\tilde{N}$ are the lcfs *of K_0 and G_0, respectively.* □

Proof. Comparing (3.5.11) and (3.4.5), observe that $G_{S_i} \equiv G_i$, where G_{S_i} is constructed as shown in Figure 3-9, with the mapping $S \equiv S_i$. Let us seek to construct a controller K_Q of the form of Figure 3-3 that will stabilize all of the G_{S_i}. By Theorem 3.4, the system $\{G_{S_i}, K_Q\}$ is stable iff the system $\{S_i, Q\}$ is stable. Restricting Q to be BIBO stable gives stability of the system $\{G_0, K_Q\}$. Thus to stabilize the set of plants $\{S_i\}$ we need only find a stable mapping Q such that the systems $\{S_i, Q\}$ are stable. Hence the problem has reduced to that of finding a single stable mapping Q that will stabilize the set of m plants $S_1, S_2, \ldots S_m$. ■

Remark 3.27 Note that in the case when m=1, we have the nonlinear version of the well known result for the linear case that the problem of simultaneously stabilizing two plants is equivalent to the strong stabilization of a single plant.

In the following section and the next chapter there will be a slight abuse of notation. We shall refer to a class of plants G_S without specifying whether it is the scheme of Figure 3-8 or Figure 3-9, however, we will usually be referring to the later. The scheme being referred to will generally be clear from the context. The same abuse of notation will be used when referring to the class of plants K_Q.

3.6 Fractional Maps

In order to fully explore the relationships between G_S and S, and dually between K_Q and Q, we now study a nonlinear equivalent of the idea of linear fractional maps. The idea is to develop a framework to characterize the class of stabilizing controllers for a given plant, and the class of plants stabilized by a given controller. The first result concerns left coprime factorizations for G_S, "stabilized" by K in a restricted sense.

Theorem 3.5

Consider a well-posed and internally stable system $\{G, K\}$ with left coprime factorizations (2.3.2), (2.3.4). Consider also any plant G_S such that

$$(\tilde{V} - \tilde{U}G_S)^{-1}\text{exists,} \tag{3.6.1}$$

then G_S has a right factorization

$$G_S = N_S M_S^{-1} \quad , \quad \begin{bmatrix} M_S \\ N_S \end{bmatrix} = \begin{bmatrix} I \\ G_S \end{bmatrix} (\tilde{V} - \tilde{U}G_S)^{-1} \tag{3.6.2}$$

and satisfies the Bezout identity

$$\tilde{V}M_S - \tilde{U}N_S = I \tag{3.6.3}$$

If M_S, N_S are stable they are coprime. Moreover defining an operator S from

$$S = \tilde{M}N_S - \tilde{N}M_S = (\tilde{M}G_S - \tilde{N})(\tilde{V} - \tilde{U}G_S)^{-1} \tag{3.6.4}$$

then under existence of the relevant inverse as in (2.3.31), M_S, N_S can be characterized *by a mapping on S as*

$$\begin{bmatrix} M_S \\ N_S \end{bmatrix} = \begin{bmatrix} \tilde{V} & -\tilde{U} \\ -\tilde{N} & \tilde{M} \end{bmatrix}^{-1} \begin{bmatrix} I \\ S \end{bmatrix} \tag{3.6.5}$$

Additionally, when K "stabilizes" G in the restricted sense of (2.3.32), then M_S, N_S will be stable iff S is stable. Furthermore, (3.6.4)-(3.6.5) give a bijection between the set of all plants G_S such that (3.6.1) holds, and the set of all operators S such that

$$M_S^{-1} = \left(\begin{bmatrix} I & 0 \end{bmatrix} \begin{bmatrix} \tilde{V} & -\tilde{U} \\ -\tilde{N} & \tilde{M} \end{bmatrix}^{-1} \begin{bmatrix} I \\ S \end{bmatrix} \right)^{-1} \quad \textit{exists.} \tag{3.6.6}$$

□

Proof. Note that under existence of $(\tilde{V} - \tilde{U}G_S)^{-1}$ we have

$$G_S = G_S(\tilde{V} - \tilde{U}G_S)^{-1}(\tilde{V} - \tilde{U}G_S) = N_S M_S^{-1} \tag{3.6.7}$$

Thus verifying (3.6.2). Now show (3.6.3)

$$\tilde{V}M_S - \tilde{U}N_S = \tilde{V}(\tilde{V} - \tilde{U}G_S)^{-1} - \tilde{U}G(\tilde{V} - \tilde{U}G_S)^{-1} = (\tilde{V} - \tilde{U}G_S)(\tilde{V} - \tilde{U}G_S)^{-1} = I \tag{3.6.8}$$

Combining $G_S = N_S M_S^{-1}$ and (3.6.3) proves (3.6.4). Now note that

$$\begin{bmatrix} \tilde{V} & -\tilde{U} \\ -\tilde{N} & \tilde{M} \end{bmatrix} \begin{bmatrix} M_S \\ N_S \end{bmatrix} = \begin{bmatrix} \tilde{V}M_S - \tilde{U}N_S \\ \tilde{M}N_S - \tilde{N}M_S \end{bmatrix} = \begin{bmatrix} I \\ S \end{bmatrix} \tag{3.6.9}$$

So under (2.3.31), (3.6.5) holds as claimed.

Under our assumptions, including (2.3.32) we have $\begin{bmatrix} \tilde{V} & -\tilde{U} \\ -\tilde{N} & \tilde{M} \end{bmatrix}$ unimodular. Hence (3.6.5) gives $\begin{bmatrix} M_S \\ N_S \end{bmatrix}$ stable iff $\begin{bmatrix} I \\ S \end{bmatrix}$ is stable. The identity mapping I is trivially stable, hence M_S, N_S are stable iff S is stable.

Now let us prove bijectivity of the maps (3.6.4)-(3.6.5). It is evident from the equations that given an S such that (3.6.6) holds, $G_S = N_S M_S^{-1}$ is constructed from (3.6.5), and (3.6.1) holds. Similarly given G_S such that (3.6.1) holds, the S obtained from (3.6.4) will satisfy (3.6.6), as $M_S = (\tilde{V}G_S - \tilde{U})^{-1}$ is invertible. Hence the mapping from each class to the other is well defined, and thus onto. To prove bijectivity it remains to prove that the images under the maps are unique.

Note that (3.6.5) shows that for each S there exists a unique pair M_S, N_S.

$$M_S^{-1} = \left(\begin{bmatrix} I & 0 \end{bmatrix} \begin{bmatrix} \tilde{V} & -\tilde{U} \\ -\tilde{N} & \tilde{M} \end{bmatrix}^{-1} \begin{bmatrix} I \\ S \end{bmatrix} \right)^{-1} \tag{3.6.10}$$

so it is necessary that S satisfy (3.6.6) for M_S^{-1} to exist. Further the plant G_S so obtained will satisfy (3.6.1). Hence the conditions (3.6.1) and (3.6.6) are equivalent. The bijectivity of the maps (3.6.4)-(3.6.5) is now established.

The map from S to G as defined by (3.6.2) is onto (surjective), as for all G such that (3.6.1) holds we have an S as given by (3.6.4) which maps to G. To prove one to oneness (injectivity) consider that there exist S_1 and S_2 such that $G_{S_1} = G_{S_2}$. Then the S's of (3.6.4) are the same, giving

$$\begin{aligned}
(\tilde{M}N_{S_1} - \tilde{N}M_{S_1}) &= (\tilde{M}N_{S_2} - \tilde{N}M_{S_2}) \\
LHS &= \begin{bmatrix} \tilde{M} & -\tilde{N} \end{bmatrix} \begin{bmatrix} N_{S_1} \\ M_{S_1} \end{bmatrix} \\
&= \begin{bmatrix} 0 & I \end{bmatrix} \begin{bmatrix} \tilde{V} & -\tilde{U} \\ -\tilde{N} & \tilde{M} \end{bmatrix} \begin{bmatrix} \tilde{V} & -\tilde{U} \\ -\tilde{N} & \tilde{M} \end{bmatrix}^{-1} \begin{bmatrix} I \\ S_1 \end{bmatrix} \\
&= \begin{bmatrix} 0 & I \end{bmatrix} \begin{bmatrix} I \\ S_1 \end{bmatrix} \\
&= S_1 \\
RHS &= \begin{bmatrix} \tilde{M} & -\tilde{N} \end{bmatrix} \begin{bmatrix} N_{S_2} \\ M_{S_2} \end{bmatrix} \\
&= \begin{bmatrix} 0 & I \end{bmatrix} \begin{bmatrix} \tilde{V} & -\tilde{U} \\ -\tilde{N} & \tilde{M} \end{bmatrix} \begin{bmatrix} \tilde{V} & -\tilde{U} \\ -\tilde{N} & \tilde{M} \end{bmatrix}^{-1} \begin{bmatrix} I \\ S_2 \end{bmatrix} \\
&= \begin{bmatrix} 0 & I \end{bmatrix} \begin{bmatrix} I \\ S_2 \end{bmatrix} \\
&= S_2 \\
S_1 &= S_2
\end{aligned}$$

Hence the mapping is injective. This gives bijectivity of the maps, thus completing the proof. ■

Remark 3.28 A dual result to this involving right factorizations of G and K, and giving a left factorization for G_S is elusive at the moment, unless additional assumptions are made. However, dualizing in terms of interchanging the roles of G and S gives an expression for S when G_S is expressed in terms of *rcfs* of G, K, as shown in the following theorem.

Remark 3.29 In the case that $(\tilde{V} - \tilde{U}G)^{-1}$ exists, then the theorem gives $G_S = G$ iff $S = 0$. ie, given a left factorization of G, K we can get a rcf for G, also for K as is shown in the dual to this theorem, Theorem 3.7.

Remark 3.30 In the linear S case the expression for G_S simplifies to give $G_S = (\tilde{M} + S\tilde{U})^{-1}(\tilde{N} + S\tilde{V})$.

Remark 3.31 This theorem is of interest in the work done by Hammer [15], and by Tay and Moore [49]. In this work the plant G is stabilized by a pre-, post-compensator pair $\tilde{V}^{-1}$, $\tilde{U}$, so that the question of well-posedness and stability of the system is reduced to that of the existence and stability of the operator $(\tilde{V} - \tilde{U}G)^{-1}$. This theorem shows that any plant G_S for which this system is well posed is related to a nominal plant G by means of (3.6.5), and is parameterized by the operator S. Furthermore, as $(\tilde{V} - \tilde{U}G_S)^{-1} = M_S$, the system is stable iff S is stable. Thus the theorem gives the class of all plants stabilized by the pre-, post-compensator pair $\tilde{V}^{-1}$, $\tilde{U}$.

Theorem 3.6

Consider a well-posed and stable system $\{G, K\}$ with right coprime factorizations (2.3.1), (2.3.3), so that existence and stability conditions (2.3.10) and (2.3.11) hold. Consider a map S such that $(M - US)^{-1}$ exists. Then S has a right factorization $S = P_G D_G^{-1}$ given by

$$\begin{bmatrix} D_G \\ P_G \end{bmatrix} = \begin{bmatrix} I \\ S \end{bmatrix} (M - US)^{-1}, \text{ and } MD_G - UP_G = I \qquad (3.6.11)$$

Further there exists a plant G_S such that

$$G_S = ND_G - VP_G = (N - VS)(M - US)^{-1} \qquad (3.6.12)$$

$$\begin{bmatrix} D_G \\ P_G \end{bmatrix} = \begin{bmatrix} M & -U \\ -N & V \end{bmatrix}^{-1} \begin{bmatrix} I \\ -G_S \end{bmatrix} \qquad (3.6.13)$$

Moreover this gives a bijection between the class of all operators S such that $(M - US)^{-1}$

exists and the class of all plants such that

$$D_G^{-1} = \left(\begin{bmatrix} I & 0 \end{bmatrix} \begin{bmatrix} M & -U \\ -N & V \end{bmatrix}^{-1} \begin{bmatrix} I \\ -G_S \end{bmatrix} \right)^{-1} \quad \textit{exists} \qquad (3.6.14)$$

□

Proof. The details of the proof of this result are the same of those of the previous theorem, as they are dual results, interchanging the roles of S and G. ∎

Remark 3.32 In the case that the plant and controller, and their factorizations, are linear, the conditions (3.6.1) and (3.6.14) are equivalent, as are (3.6.6) and the existence of $(M + US)^{-1}$. The theorems then give the same result.

Remark 3.33 Note that these theorems provide a natural setting for generating the class of all plants stabilized by the controller K, and in the dual case, the class of all controllers stabilizing a given plant. Theorem 3.5 may be applied to the main results of [41] to generate the class of all plants bounded-input stabilized by a given controller. By assuming linearity of K, it is possible to show that the class of all controllers stabilized by K can be generated by Theorem 3.5. More general results are elusive at this time, so that it is not possible to say whether G_S will be stabilized by K.

Remark 3.34 These results may be readily dualized, interchanging the roles of the plant and controller, as is explored in the following theorems.

Theorem 3.7

Consider a well-posed and stable system $\{G, K\}$ with left coprime factorizations (2.3.2), (2.3.4). Consider also any controller K_Q such that

$$(\tilde{M} - \tilde{N}K_Q)^{-1} \quad \textit{exists,} \qquad (3.6.15)$$

then K_Q has a right factorization $K_Q = U_Q V_Q^{-1}$, not necessarily stable, given by

$$\begin{bmatrix} U_Q \\ V_Q \end{bmatrix} = \begin{bmatrix} K_Q \\ I \end{bmatrix} (\tilde{M} - \tilde{N}K_Q)^{-1} \qquad (3.6.16)$$

and satisfies the Bezout identity

$$\tilde{M}V_Q - \tilde{N}U_Q \;=\; I \tag{3.6.17}$$

If V_Q, U_Q are stable they are coprime. Moreover V_Q, U_Q can be characterized in terms of an operator Q, defined as

$$\begin{aligned} Q &= \tilde{V}U_Q - \tilde{U}V_Q & (3.6.18)\\ &= (\tilde{V}K_Q - \tilde{U})(\tilde{M} - \tilde{N}K_Q)^{-1} & (3.6.19)\end{aligned}$$

Under existence of the inverse, as in (2.3.31),

$$\begin{bmatrix} U_Q \\ V_Q \end{bmatrix} = \begin{bmatrix} \tilde{V} & -\tilde{U} \\ -\tilde{N} & \tilde{M} \end{bmatrix}^{-1} \begin{bmatrix} Q \\ I \end{bmatrix} \tag{3.6.20}$$

Additionally if $\{G,\, K\}$ is stable so that (2.3.32) holds, V_Q, U_Q are stable iff Q is stable. Moreover, equations (3.6.18)-(3.6.20) give a bijection between the set of all controllers K_Q such that (3.6.15) holds, and the set of all operators Q such that

$$V_Q^{-1} = \left(\begin{bmatrix} 0 & I \end{bmatrix} \begin{bmatrix} \tilde{V} & -\tilde{U} \\ -\tilde{N} & \tilde{M} \end{bmatrix}^{-1} \begin{bmatrix} Q \\ I \end{bmatrix} \right)^{-1} \quad \textit{exists.} \tag{3.6.21}$$

□

Theorem 3.8

Consider a well-posed and stable system $\{G,\, K\}$ with right coprime factorizations (2.3.1), (2.3.3), so that (2.3.10) and (2.3.11) hold. Consider a map Q such that

$$(V + NQ)^{-1} \quad \textit{exists.} \tag{3.6.22}$$

Then Q has a right factorization $Q = T_K R_K^{-1}$ given by

$$\begin{bmatrix} T_K \\ R_K \end{bmatrix} = \begin{bmatrix} Q \\ I \end{bmatrix} (V + NQ)^{-1} \tag{3.6.23}$$

and satisfies

$$VR_K + NT_K \;=\; I \tag{3.6.24}$$

Further there exists a controller K_Q such that

$$\begin{aligned} K_Q &= UR_K + MT_K & (3.6.25)\\ &= (U+MQ)(V+NQ)^{-1} & (3.6.26)\end{aligned}$$

and

$$\begin{bmatrix} T_K \\ R_K \end{bmatrix} = \begin{bmatrix} M & U \\ N & V \end{bmatrix}^{-1} \begin{bmatrix} Q \\ I \end{bmatrix} \tag{3.6.27}$$

Moreover (3.6.25)-(3.6.27) give a bijection between the class of all operators Q such that (3.6.22) holds and the class of all plants such that

$$R_K^{-1} = \left(\begin{bmatrix} 0 & I \end{bmatrix} \begin{bmatrix} M & -U \\ -N & V \end{bmatrix}^{-1} \begin{bmatrix} Q \\ I \end{bmatrix} \right)^{-1} \quad \textit{exists.} \tag{3.6.28}$$

□

3.7 Conclusion

In this chapter we have developed a left factorization approach to a nonlinear generalization of the Youla-Kucera parameterization for nonlinear systems. By starting with a Bezout identity, and the assumption of differential boundedness, the class of all controllers which bounded-input stabilize a given plant was derived. This dualizes readily to give the class of all plants stabilized by a given controller. These classes could be combined, and a simple characterization of the class of all bounded-input stable plane controller pairs was derived. These results specialize readily to the linear case. Robust stabilization results were thus obtained.

The theorems of the last section on a generalization of linear fractional mappings completes the relationship between G_{S_r} and S_r, and dually K_{Q_r} and Q_r. Given an original stable plant and controller, it is now possible to find the S_r, Q_r pair that corresponds to any other plant, controller pair, and thus the stability properties of the new system may be deduced. Additionally the deviations of the actual plant and controller from the nominal plant and controller may be accounted for in some way.

The question to be tackled now is whether it is possible to implement these results. Many of the preliminary results were based on work by Hammer and use a geometric approach. The later results all assume the existence of the factorizations. In the next chapter a state space approach to the problem is taken, and we attempt to derive realizations of the factors of the plant and controller.

Chapter 4

State Space and Factorizations

4.1 Introduction

Interest in finding state-space realizations for the factorizations of nonlinear systems is relatively new. Initial studies in the area were carried out by Sontag [44], who presented results giving a right factorization for a class of nonlinear plants, and linked them to the problem of finding a smooth stabilizing state feedback map for the plant of interest. Krener [28] presented results showing that right and left factorizations could be obtained for nonlinear plants with controller and observer normal forms. Of particular interest here was the augmentation of the plant by a unity feedthrough term, which appeared necessary to obtain a left factorization. This is also required in the work by Moore and Irlicht [36], in which a factorization theory is developed for a quite general form of nonlinear plants, giving right factorizations, and left factorizations for an augmented version of the plant. In [56] Verma presented a construction of the right coprime factorization of a general continuous time nonlinear plant, while in [17] Hammer gives a construction for discrete time systems.

Throughout the previous chapters, while considering the input-output approach to nonlinear factorizations, it has become apparent that by making very few simplifying assumptions, a framework which closely mimics the linear factorization theory may be developed. It is now of interest to see how closely the state-space approach to nonlinear factorizations mimics the linear theory. It is the purpose of this chapter to examine this.

Starting with a general state space description of a continuous time plant, stable right factorizations are developed, based on the assumption that the state equation of the plant is stabilizable by nonlinear state feedback. A stabilizing controller for a given plant is derived, and some of the results derived in previous chapters are applied giving an approach to the stabilization problem which allows for differing initial conditions and unmodelled dynamics. The development parallels the development of the theory for linear systems.

Although we work in continuous time in this chapter, the results for discrete time are very similar. The main difference being that instead of appealing to Theorem 4.1, an inductive proof may be used.

In Section 4.2 the stage is set for the rest of the chapter. The class of nonlinear plants which we are interested in is defined, and some useful results concerning the algebra of nonlinear operators are proved. A right factorization for a plant for which there exists a stabilizing state feedback map is derived in Section 4.3 we derive right factorizations for a plant for which there exists a stabilizing state feedback map. A controller is also designed, based on the idea of constructing a stable state estimator for the plant, a right factorization for this controller is also presented. Through the use of some of the results of Chapter 2 it is shown that these factorizations are coprime and that the plant controller feedback loop is stable. An approach to the stabilization of a plant with different initial conditions to those of the controller through the use of Theorem 3.6, p. 57, is also presented. In Section 4.4.1 a special form of the nonlinear system is considered as a means of obtaining left factorizations is presented. Theorem 3.4, p. 50, may then be applied to give the class of all bounded input stable plants and controllers. In Section 4.5 some concrete examples are given. The universal stabilizing controller of Nussbaum [39] is factorized.

4.2 Preliminaries

4.2.1 Continuous Time Nonlinear Operators.

Given a real vector space $\mathcal{X}$, define the space of trajectories within $\mathcal{X}$, $C(\mathcal{X})$ as in Section 2.2. Any function which is continuous and has continuous first derivative is called $\mathcal{C}^1$.

In this chapter slightly different notation to that of the previous chapters is used. Assume that the state space realization of a general nonlinear operator, $G(x_0): C(\mathcal{U}) \mapsto C(\mathcal{Y})$, which maps inputs $u(\cdot)$ to outputs $y(\cdot)$, is of the form,

$$G(x_0) : \begin{array}{rcl} \dot{x} & = & f(x,u) \\ y & = & h(x,u) \end{array} \qquad x(0) = x_0 \tag{4.2.1}$$

As a plant with different initial conditions is almost guaranteed to give a different map from the input space to the output space, this dependency is made explicit. Thus the operator $G(x_0)$ is different to $G(x_1)$ for $x_0 \neq x_1$.

Note that we are implicitly assuming causality of the plant by choosing a state space realization of the form of (4.2.1).

A fundamental property of differential equations that we shall be exploiting is the existence and uniqueness of solutions of the differential equation. A brief review of the results which will be required is now presented. The following theorem, adapted from Hirsh and Smale [21] and stated without proof is useful.

Theorem 4.1

Let $f: \mathcal{X} \mapsto \mathcal{X}$ be a $\mathcal{C}^1$map and let $x_0 \in \mathcal{X}$. Then there exists a unique maximal open interval (a, b) containing 0, and a unique function $x: (a, b) \mapsto \mathcal{X}$ satisfying

$$\dot{x} = f(x) \qquad , \; x(0) = x_0 \tag{4.2.2}$$

□

Remark 4.1 Note that a, b may be equal to plus or minus infinity. In the case that b is finite, the system is unstable, with finite escape time. Similarly if a is finite, the reverse time system has finite escape time.

Remark 4.2 This theorem also gives results for the time varying case, and for systems of the form of (4.2.1), as is explored in the following corollaries.

Corollary 4.1 *Let $f: \mathcal{X} \times \mathcal{R} \mapsto \mathcal{X}$ be a $\mathcal{C}^1$map, and let $x_0 \in \mathcal{X}$. Then there exists a unique maximal open interval (a, b) containing 0, and a unique function $x: (a, b) \mapsto \mathcal{X}$ satisfying*

$$\dot{x} = f(x, t) \qquad , \; x(0) = x_0 \tag{4.2.3}$$

□

Proof. Let $y = \begin{pmatrix} x \\ t \end{pmatrix}$, and $g(y) = \begin{pmatrix} f(x,t) \\ 1 \end{pmatrix}$. As f is $\mathcal{C}^1$, g is $\mathcal{C}^1$, now apply Theorem 4.1. ∎

Corollary 4.2 *Let $f: \mathcal{X} \times \mathcal{U} \mapsto \mathcal{X}$ be a $\mathcal{C}^1$map, and let $x_0 \in \mathcal{X}$. Then given $u \in C(\mathcal{U})$ there exists a unique maximal open interval (a, b) containing 0, and a unique* *function* $x\colon (a,b) \mapsto \mathcal{X}$ *satisfying*

$$\dot{x} = f(x, u(t)) \qquad , \; x(0) = x_0 \tag{4.2.4}$$

□

Proof. Given $u \in C(\mathcal{U})$ set $g(x,t) = f(x,u(t))$ which is $\mathcal{C}^1$as f and u are both $\mathcal{C}^1$, and apply the previous corollary. ■

In order to guarantee existence and uniqueness of solutions it is assumed that the map $f(\cdot,\cdot)$ of (4.2.1) is $\mathcal{C}^1$. Unless otherwise stated all functions in the work to follow shall be assumed to be $\mathcal{C}^1$.

4.2.2 Algebra with Nonlinear Operators.

Consider two operators of the form of (4.2.1), $A(x_0)\colon C(\mathcal{U}) \mapsto C(\mathcal{Y})$ and $B(v_0)\colon C(\mathcal{Y}) \mapsto C(\mathcal{Z})$.

$$A(x_0) : \begin{array}{rcl} \dot{x} &=& f_A(x,u) \\ y &=& h_A(x,u) \end{array} \qquad x(0) = x_0 \tag{4.2.5}$$

$$B(v_0) : \begin{array}{rcl} \dot{v} &=& f_B(v,y) \\ z &=& h_B(v,y) \end{array} \qquad v(0) = v_0 \tag{4.2.6}$$

Then the operator $C(x_0,v_0) \;=\; B(v_0)A(x_0)\colon C(\mathcal{U}) \mapsto C(\mathcal{Z})$ will have state space description

$$C(x_0,v_0) \; : \; \begin{pmatrix} \dot{x} \\ \dot{v} \end{pmatrix} = \begin{pmatrix} f_A(x,u) \\ f_B(v, h_A(x,u)) \end{pmatrix} \quad \begin{pmatrix} x(0) \\ v(0) \end{pmatrix} = \begin{pmatrix} x_0 \\ v_0 \end{pmatrix} \tag{4.2.7}$$

$$z = h_B(v, h_A(x,u)) \tag{4.2.8}$$

Note that in general the dimension of the state of $C(x_0,v_0)$ is equal to the sum of the dimensions of the states of $A(x_0)$ and $B(v_0)$. In some special cases, however, it may be possible to reduce the state, as shown in the following lemma.

Lemma 4.1 [State Reduction] *Consider operators $A(x_0)$, $B(v_0)$ as given by (4.2.5), (4.2.6). Then if*

$$f_B(x, h_A(x,u)) = f_A(x,u) \quad \forall x,\ u \text{ and } \quad x_0 = v_0 \tag{4.2.9}$$

then $C(x_0, v_0) \ = \ C(x_0, x_0) \ = \ C(x_0)$ is reduced to the form

$$C(x_0) : \begin{array}{rcl} \dot{x} & = & f_A(x,u) \qquad\qquad x(0) = x_0 \\ z & = & h_B(x, h_A(x,u)) \end{array} \tag{4.2.10}$$

□

Proof. Suppose that (4.2.9) holds, and consider the evolution of the state equations (4.2.5), (4.2.6). Substituting $v(t) = x(t)$ into (4.2.7) gives

$$\begin{array}{rcl} \dot{v}(t) & = & f_B(v(t), h_A(x(t), u(t))) \\ & = & f_B(x(t), h_A(x(t), u(t))) \\ & = & f_A(x(t), u(t)) \\ & = & \dot{x}(t) \end{array}$$

Hence a solution of (4.2.5), (4.2.6) is $v(t) = x(t)$, $\forall t$. Note that f_B, h_A, f_A are $\mathcal{C}^1$functions. Hence by Theorem 4.1 this is the unique solution, and the lemma is established. ■

In deriving later results it will be necessary to be able to invert a nonlinear operator of the form of (4.2.1). The following lemma shows that given the existence of a map $h^{\#}: \mathcal{X} \times \mathcal{Y} \mapsto \mathcal{U}$, associated with the map h of (4.2.1), it is possible to invert the operator. The map $h^{\#}(\cdot,\cdot)$ is called the *pseudo-inverse* of $h(\cdot,\cdot)$, and the map $h(\cdot,\cdot)$ is called *pseudo-invertible*, this is an analogue of the *reversible feedback function* of [18]. Note that if $h(\cdot,\cdot)$ is $\mathcal{C}^1$, then $h^{\#}(\cdot,\cdot)$ is $\mathcal{C}^1$.

Lemma 4.2 [Inversion] *Consider an operator $G(x_0)$ as in (4.2.1), construct the operator $S(x_0): C(\mathcal{Y}) \mapsto C(\mathcal{U})$ as follows.*

$$S(v_0) : \begin{array}{rcl} \dot{v} & = & f(v, h^{\#}(v,y)) \quad v(0) = v_0 \\ u & = & h^{\#}(v,y) \end{array} \tag{4.2.11}$$

Consider also that $h^{\#}$ satisfies

$$h^{\#}(x, h(x,u)) \;=\; u \qquad \forall x,\; u \tag{4.2.12}$$

then $S(x_0)$ is a left inverse for $G(x_0)$, ie $S(x_0)G(x_0) \;=\; I$, and if $h^{\#}$ satisfies

$$h(x, h^{\#}(x,u)) \;=\; u \qquad \forall x,\; u \tag{4.2.13}$$

then $S(x_0)$ is a right inverse for $G(x_0)$, ie $G(x_0)S(x_0) \;=\; I$. Moreover, when h and $h^{\#}$ satisfy both (4.2.12), then (4.2.13), $S(x_0)$ is an inverse for $G(x_0)$, ie

$$G^{-1}(x_0) \;=\; S(x_0) \tag{4.2.14}$$

□

Proof. Consider the state equation of the composition $S(x_0)G(x_0)$

$$\dot{x} \;=\; f(x,u) \qquad\qquad x(0) = x_0 \tag{4.2.15}$$

$$\dot{v} \;=\; f(v, h^{\#}(v, h(x,u))) \qquad\qquad v(0) = x_0 \tag{4.2.16}$$

Note that by (4.2.12) we have for $x \;=\; v$, $f(x,u) \;=\; f(v, h^{\#}(v, h(x,u)))$. Applying Lemma 4.1 gives $x(t) = v(t)$, $\forall t$. The output equation for $S(x_0)G(x_0)$ thus becomes

$$z \;=\; h^{\#}(v, h(x,u)) \;=\; u \tag{4.2.17}$$

This proves the first part of the lemma. A similar argument shows that when (4.2.13) is satisfied $G(x_0)S(x_0) \;=\; I$. This completes the proof. ∎

Remark 4.3 Note the dependence on initial conditions for ensuring that the states remain equal for all time. This may be difficult to guarantee, additionally the error dynamics may be such that any small error is magnified. The question of when it is possible to stably invert a function, *i.e.* invert it and have the error dynamics such that any error will decrease with time, is a difficult one, and is dependent on the particular functions f, h, $h^{\#}$.

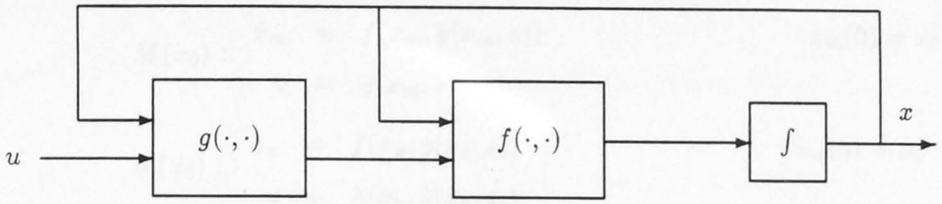

Figure 4-1: State feedback $g(\cdot,\cdot)$.

4.3 Right Factorizations

In this section a right factorization for the plant $G(x_0)$ is derived. Based on the ideas thus presented a candidate stabilizing controller is given. It is shown that this also has a right factorization.

4.3.1 Right Factorization for $G(x_0)$

The development of a stable right factorization for the plant $G(x_0)$ is critically dependent on the solution of the smooth state feedback stabilization problem for the state equation of $G(x_0)$. This is in itself an open problem, and a treatment of such is beyond the bounds of the thesis. Here, the assumption is made that for plants of interest the stabilization problem has been solved and that the solution is available.

Consider the state feedback map $g(\cdot,\cdot): \mathcal{X} \times \mathcal{U} \mapsto \mathcal{U}$, applied as in Figure 4-1, so that the state equation for $G(x_0)$ becomes

$$\dot{x} = f(x, g(x,u)), \qquad x(0) = x_0 \tag{4.3.1}$$

Assumption 4.1 *For the plant $G(x_0)$ of (4.2.1) there exists a pseudo-invertible $\mathcal{C}^1$map $g(\cdot,\cdot): \mathcal{X} \times \mathcal{U} \mapsto \mathcal{U}$ such that the state equation (4.3.1) is stable, and there exists a map $g^{\#}(\cdot,\cdot)$ which satisfies both (4.2.12) and (4.2.13).*

It is now possible to construct a stable right factorization for $G(x_0)$ as is explored in the following lemma. This lemma is equivalent to Theorem 3 of Verma, [56], and parallels the discrete time results found in Hammer, [17]

Lemma 4.3 *Consider a plant $G(x_0)$ such that there exists a map $g(\cdot,\cdot)$ satisfying Assumption 4.3.1. Then it is possible to construct a stable right factorization for $G(x_0)$ as follows.*

$$G(x_0) = N(x_0)M^{-1}(x_0) \tag{4.3.2}$$

$$M(x_0):\quad \begin{aligned} \dot{x}_m &= f(x_m, g(x_m, s)) \qquad x_m(0) = x_0 \\ z &= g(x_m, s) \end{aligned} \tag{4.3.3}$$

$$N(x_0):\quad \begin{aligned} \dot{x}_n &= f(x_n, g(x_n, s)) \qquad x_n(0) = x_0 \\ y &= h(x_n, g(x_n, s)) \end{aligned} \tag{4.3.4}$$

□

Remark 4.4 By introducing the notion of detectability Verma [56] is able to prove that this a coprime factorization. The notion of detectability used is that if u and y are stable, then x is stable, and furthermore, that for some $\beta > 0$, $||x|| \leq \beta|| \begin{bmatrix} y \\ u \end{bmatrix} ||$.

Remark 4.5 Note that the equation $G(x_0) = N(x_0)M^{-1}(x_0)$ depends on the initial conditions being identical, so that Remark 4.3 is appropriate.

Remark 4.6 The requirement that $g(\cdot,\cdot)$ be pseudo-invertible is necessary for invertibility of $M(x_0)$. It does not appear overly restrictive, as in the linear case we have $g(x, u) = Fx + u$, so that $g^{\#}(x, y) = y - Fx$, where F is some matrix chosen such that $A + BF$ is stable. Furthermore, note that Sontag [45] proves an input to state stability result which gives a stability result satisfying Assumption 4.3.1. Specifically it is shown that for systems of the form of (4.2.1), a feedback law of the form

$$g(x, u) \;=\; K(x) + G(x)u \tag{4.3.5}$$

where $G(x)$ is invertible for all x. In this case we have

$$g^{\#}(x, v) \;=\; G(x)^{-1}(v - K(x)) \tag{4.3.6}$$

and Assumption 4.3.1 is satisfied.

4.3.2 A Stabilizing Controller

Following the linear theory a controller is designed based on the idea of a state estimator. Consider that there exists a map $l(\cdot,\cdot): \mathcal{X} \times \mathcal{Y} \mapsto \mathcal{X}$, such that the state equation

$$\dot{v} \;=\; f(v, u) - l(v, h(x, u) - h(v, u)) \quad v(0) = v_0 \tag{4.3.7}$$

acts as a state estimator for (4.2.1). *i.e.*

Assumption 4.2 *For the plant $G(x_0)$ of (4.2.1) there exists a $\mathcal{C}^1$map $l(\cdot,\cdot)\colon \mathcal{X}\times\mathcal{Y}\mapsto\mathcal{X}$ such that (4.3.7) acts as a state estimator for $G(x_0)$, in that for all x_0, v_0, as $t\to\infty$, $v(t)\to x(t)$.*

It is evident that $l(x,0)=0$, $\forall x$, otherwise when $v(t)=x(t)$, $\dot v(t)\neq \dot x(t)$. As in the previous case of the design of a stabilizing state feedback map, the derivation of an $l(\cdot,\cdot)$ for a particular realization (4.2.1) is an open problem, and as such is beyond the scope of the thesis. The controller $K(x_0)\colon C(\mathcal{Y})\mapsto C(\mathcal{U})$ is then constructed as follows.

$$K(x_0):\begin{array}{rcl}\dot x_k &=& f(x_k,g(x_k,0))-l(x_k,y-h(x_k,g(x_k,0))),\qquad x_k(0)=x_0\\ u &=& g(x_k,0)\end{array}\tag{4.3.8}$$

The stable right factorization $K=UV^{-1}$ is realizable with state space realizations

$$V(v_0):\begin{array}{rcl}\dot x_v &=& f(x_v,g(x_v,0))-l(x_v,s)\qquad\qquad x_v(0)=v_0\\ z &=& h(x_v,g(x_v,0))+s\end{array}\tag{4.3.9}$$

$$U(u_0):\begin{array}{rcl}\dot x_u &=& f(x_u,g(x_u,0))-l(x_u,s)\qquad\qquad x_u(0)=u_0\\ u &=& g(x_u,0)\end{array}\tag{4.3.10}$$

Coprimeness of these factorizations is shown via Lemma 2.3, p. 23. Consider the inverse of the operator $\begin{bmatrix} M & -U\\ -N & V\end{bmatrix}$. First note that $M(x_0)$ and $N(x_0)$ have the same initial conditions, and the same state when driven from the same input. Let us denote the identical states for these operators as x_m. Similarly $V(x_0)$ and $U(x_0)$ have identical states denoted x_v.

$$\begin{bmatrix} M & -U\\ -N & V\end{bmatrix}\begin{pmatrix} x_0\\ x_0\end{pmatrix}:$$

$$\begin{pmatrix}\dot x_m\\ \dot x_v\end{pmatrix}=\begin{pmatrix} f(x_m,g(x_m,s_1))\\ f(x_v,g(x_v,0))-l(x_v,s_2)\end{pmatrix},\quad \begin{pmatrix} x_m(0)\\ x_v(0)\end{pmatrix}=\begin{pmatrix} x_0\\ x_0\end{pmatrix}\tag{4.3.11}$$

$$\begin{pmatrix} u\\ y\end{pmatrix}=\begin{pmatrix} g(x_m,s_1)-g(x_v,0)\\ s_2+h(x_v,g(x_v,0))-h(x_m,g(x_m,s_1))\end{pmatrix}\tag{4.3.12}$$

From Lemma 4.2, invertibility of this system follows if it is possible to rearrange (4.3.12) to give s_1, s_2 in terms of u, y. Note that

$$s_1 \;=\; g^{\#}(x_m,u+g(x_v,0))\tag{4.3.13}$$

$$s_2 \;=\; y + h(x_m, u + g(x_v, 0)) - h(x_v, g(x_v, 0)) \tag{4.3.14}$$

Hence the operator is invertible and has state space realization given by:

$$\begin{bmatrix} M & -U \\ -N & V \end{bmatrix}^{-1} \begin{pmatrix} x_0 \\ x_0 \end{pmatrix} :$$

$$\begin{pmatrix} \dot{v}_m \\ \dot{v}_v \end{pmatrix} = \begin{pmatrix} f(v_m, u + g(v_v, 0)) \\ f(v_v, g(v_v, 0)) - l(v_v, y + h(v_m, u + g(v_v, 0)) - h(v_v, g(v_v, 0))) \end{pmatrix},$$

$$\begin{pmatrix} v_m(0) \\ v_v(0) \end{pmatrix} = \begin{pmatrix} x_0 \\ x_0 \end{pmatrix} \tag{4.3.15}$$

$$\begin{pmatrix} s_1 \\ s_2 \end{pmatrix} = \begin{pmatrix} g^{\#}(v_m, u + g(v_v, 0)) \\ y + h(v_m, u + g(v_v, 0)) - h(v_v, g(v_v, 0)) \end{pmatrix} \tag{4.3.16}$$

Note that for $u = y = 0$ and $v_m(0) = v_v(0)$, $\dot{v}_m = \dot{v}_v = f(v_v, g(v_v, 0))$, which is stable. In the case $u \neq 0$, $y \neq 0$ it is not as clear that (4.3.15) will remain stable, although the assumption Assumption 4.3.2 implies that the state v_v will mimic v_m, giving stability. In [44] and [45] it is proven that if a system may be stabilized by state feedback so that for the zero input case the system is stable, then the system may also be input to state stabilized. Hence in this case it would seem that (4.3.15) will remain stable in the case $u \neq 0$, $y \neq 0$, at least for bounded u, y, however precise results are elusive at this time.

If this inverse operator (4.3.15), (4.3.16) is stable, Lemma 2.3 and Theorem 2.1 can be applied to give coprimeness of the factorizations and stability of the system $\{G(x_0), K(x_0)\}$.

These results are summarized in the following lemma.

Lemma 4.4 *Consider a plant $G(x_0)$, with state space description (4.2.1), such that there exist mappings $g(\cdot,\cdot)$, and $l(\cdot,\cdot)$ satisfying Assumption 4.3.1 and Assumption 4.3.2 respectively. Then there exists a controller $K(x_0)$, given by (4.3.8), and such that the system $\{G(x_0), K(x_0)\}$ is stable. Furthermore, the right factorizations of $G(x_0)$ and $K(x_0)$ given by (4.3.3), (4.3.4), (4.3.9), (4.3.10) are coprime.* □

Remark 4.7 Note that this is only one possible approach to the stabilization of nonlinear systems, albeit the one which is the most fruitful if we are to take advantage of Assumption 4.3.1.

4.4 Left Coprime Factorizations

With the formulation of the plant $G(x_0)$ as in (4.2.1) it does not appear possible to generate stable left factorizations in the form of (2.3.2). In the linear theory the construction of the left factorizations is critically dependent on being able to additively decompose the state of $G(x_0)$ into it's stable and unstable parts. The state of $\tilde{N}$, x_n is the stable part of the state of G, x. The state of $\tilde{M}^{-1}$, x_m models the difference between these, giving $x = x_n + x_m$ as is shown in the following equations.

$$\begin{aligned}
\tilde{N} \ &: \ \dot{x}_n = Ax_n + Bu + H(Cx_n + Du) \\
& s = Cx_n + Du \\
\tilde{M}^{-1} \ &: \ \dot{x}_m = Ax_m - Hs = Ax_m - H(Cx_n + Du) \\
G \ &: \ \dot{x} = \dot{x}_n + \dot{x}_m = A(x_n + x_m) + Bu = Ax + Bu
\end{aligned}$$

A nonlinear analogue of this process is not possible in the framework developed in this chapter. Other attempts have been more successful. In Moore and Irlicht [36] *lcfs* were obtained by using a specialized version of (4.2.1), as is now explored.

4.4.1 Augmented Systems

In this section a restricted form of (4.2.1) is given, and it is shown how this leads to a more complete factorization theory than that developed in the previous sections. These results are presented without proof, further details may be found in the paper by Moore and Irlicht [36].

Consider that the operator $G(x_0)$ has state space description given by

$$G(x_0) \ : \ \begin{aligned} \dot{x} &= A(x)x + B(x)u \\ y &= C(x)x + D(x)u \end{aligned} \qquad x(0) = x_0 \tag{4.4.1}$$

Then the existence of a map $g(\cdot,\cdot) = F(x)x + u$ satisfying Assumption 4.3.1 gives a stable right factorization as developed in Section 4. The exact forms of $M(x_0)$ and $N(x_0)$ follow from the definitions given, and may be found in Section 2 of [36]. Further if there exists a function $l(\cdot,\cdot) = H(x)y$ satisfying Assumption 4.3.2, it is possible to construct a controller $K(x_0)$, with stable right factorization $U(x_0)V(x_0)^{-1}$.

Working with this form of (4.2.1) is instructive as it illustrates more clearly how the linear theory generalizes to the results presented in this chapter. Although further results do not appear attainable in this framework, it does present a natural setting for the development of left coprime factorizations, and thus a more complete factorization theory.

Consider the generalization of (4.4.1) where there is an external signal y_w which is injected into each of the matrices $A(\cdot)$, $B(\cdot)$, $C(\cdot)$, $D(\cdot)$ as follows,

$$G_w(x_0) : \begin{array}{rcl} \dot{x} & = & A(y_w)x + B(y_w)u \\ y & = & C(y_w)x + D(y_w)u \end{array} \qquad x(0) = x_0 \tag{4.4.2}$$

Here y_w is a signal which is generated by a strictly causal filter $W(w_0)$ acting on y, or u. Then by constructing the matrices $F(y_w)$, $H(y_w)$ such that $A(y_w) + B(y_w)F(y_w)$ and $A(y_w) + H(y_w)C(y_w)$ are stable for all y_w, it is possible to construct stable right and left factorizations for $G_w(x_0)$ as follows.

$$G_w(x_0) = N(x_0)M(x_0)^{-1}$$

$$M(x_0) : \begin{array}{rcl} \dot{x}_m & = & (A(y_w) + B(y_w)F(y_w))x_m + B(y_w)s_r \\ u & = & F(y_w)x_m + s_r \end{array} \qquad x_m(0) = x_0$$

$$N(x_0) : \begin{array}{rcl} \dot{x}_n & = & (A(y_w) + B(y_w)F(y_w))x_n + B(y_w)s_r \\ y & = & (C(y_w) + D(y_w)F(y_w))x_n + D(y_w)s_r \end{array} \qquad x_n(0) = x_0$$

$$G_w(x_0) = \tilde{M}(x_0)^{-1}\tilde{N}(x_0)$$

$$\tilde{N}(x_0) : \begin{array}{rcl} \dot{x}_{\tilde{N}} & = & A(y_w)x_{\tilde{N}} + B(y_w)u + H(y_w)(C(y_w)x_{\tilde{N}} + D(y_w)u) \\ s_1 & = & C(y_w)x_{\tilde{N}} + D(y_w)u \end{array} \qquad x_{\tilde{N}}(0) = x_0/2$$

$$\tilde{M}(x_0) : \begin{array}{rcl} \dot{x}_{\tilde{M}} & = & (A(y_w) + H(y_w)C(y_w))x_{\tilde{M}} - H(y_w)y \\ s_2 & = & -C(y_w)x_{\tilde{M}} + y \end{array} \qquad x_{\tilde{M}}(0) = x_0/2$$

Note that as $A(\cdot)$, $B(\cdot)$, $C(\cdot)$, $D(\cdot)$ are matrices it is possible to add the states of $\tilde{M}^{-1}$ and $\tilde{N}$ to get the state of G, ie $x = v_{\tilde{M}} + x_{\tilde{N}}$, where $v_{\tilde{M}}$ is the state of $\tilde{M}^{-1}$. The choice $x_{\tilde{M}}(x_0) = x_0/2$ is somewhat arbitrary since $G(x_0) = \tilde{M}(m_0)^{-1}\tilde{N}(n_0)$ for all m_0, n_0 such that $m_0/2 + n_0/2 = x_0$.

A controller $K(x_0)$ may be constructed, having left and right factorizations as follows.

$$K_w(x_0) : \begin{array}{rcl} \dot{x}_k & = & (A(y_w) + B(y_w)F(y_w))x_k - H(y_w)(y - (C(y_w) + D(y_w)F(y_w))x_k) \\ u & = & F(y_w)x_k \end{array} \qquad x_k(0) = x_0 \tag{4.4.3}$$

$$K(x_0) = U(x_0)V(x_0)^{-1}$$

$$V(x_0): \quad \begin{aligned} \dot{x}_v &= (A(y_w) + B(y_w)F(y_w))x_v + B(y_w)s_l \\ u &= (C(y_w) + D(y_w)F(y_w))x_v + s_l \end{aligned} \qquad x_v(0) = x_0$$

$$U(x_0): \quad \begin{aligned} \dot{x}_u &= (A(y_w) + B(y_w)F(y_w))x_u + B(y_w)s_l \\ y &= F(y_w)x_u \end{aligned} \qquad x_u(0) = x_0$$

$$K(x_0) = \tilde{V}(x_0)^{-1}\tilde{U}(x_0)$$

$$\tilde{U}(x_0): \quad \begin{aligned} \dot{x}_{\tilde{U}} &= (A(y_w) + H(y_w)C(y_w)x_{\tilde{U}} - H(y_w)y \\ s_1 &= F(y_w)x_{\tilde{U}} \end{aligned} \qquad x_{\tilde{U}}(0) = x_0/2$$

$$\tilde{V}(x_0): \quad \begin{aligned} \dot{x}_{\tilde{V}} &= (A(y_w) + H(y_w)C(y_w))x_{\tilde{V}} + (B(y_w) + H(y_w)D(y_w))u \\ s_2 &= -F(y_w)x_{\tilde{V}} + u \end{aligned} \qquad x_{\tilde{V}}(0) = x_0/2$$

As in Section 4.3 when there are no external inputs to the system, and when the initial conditions of the plant and controller are the same, the state of $K_w(x_0)$ will track that of $G_w(x_0)$ giving stability of the system $\{G_w(x_0), K_w(x_0)\}$.

The problem then addressed is how to construct the signal y_w such that it may be included in the fractional descriptions of the plant and controller in a natural way. By making y_w a function of the output y of $G(x_0)$, it is possible to construct right and left factorizations for $G(x_0)$ and $K(x_0)$, however difficulties are encountered in trying to derive left factorizations of the controller which satisfy the Bezout identity $\tilde{V}M - \tilde{U}N = I$. In fact it is shown that with this particular formulation it is not possible to construct a left factorization which satisfies this Bezout identity. To overcome this problem the notion of an augmented plant $\mathcal{G}(x_0) = [G(x_0)' \;\; I]'$ is introduced. The state of $G(x_0)$ is used to construct the signal y_w as follows.

$$W(x_0): \quad \begin{aligned} \dot{x}_w &= A(x_w)x_w + B(x_w)u \\ y_w &= x_w \end{aligned} \qquad x_w(0) = x_0 \qquad (4.4.4)$$

Hence $G(x_0) = G_w(x_0)$ and the right factorizations derived for $G(x_0)$ are equal to those of $G_w(x_0)$. The unity feedthrough term of the augmented plant ensures that the input to $G(x_0)$ is available to the controller $\mathcal{K}(x_0)$ which is constructed as follows.

$$\mathcal{K}(x_0) \;:\; C(\mathcal{Y}) \times C(\mathcal{U}) \mapsto C(\mathcal{U}) \qquad \begin{pmatrix} y \\ u \end{pmatrix} \mapsto u_k$$

$$\begin{array}{rcll} \dot{x}_w & = & A(x_w)x_w + B(x_w)u & x_w(0) = x_0 \\ \dot{x}_k & = & (A(x_w) + B(x_w)F(x_w))x_k - H(x_w)(y - (C(x_w) + D(x_w)F(x_w))x_k) & x_k(0) = x_0 \\ u_k & = & F(x_w)x_k & \end{array} \tag{4.4.5}$$

Note that the state x_w is the same as that of $W(x_0)$, so that the signal fed into the matrices $A(\cdot), \ldots,\ D(\cdot)$ in (4.4.3) and (4.4.2) are the same. It is shown that if the system $\{\mathcal{G}(x_0),\ \mathcal{K}(x_0)\}$ is bounded-input stable, then there exists a controller $\bar{\mathcal{K}}(x_0)$ such that the system $\{G(x_0),\ \bar{\mathcal{K}}(x_0)\}$ is bounded-input stable.

Factorizations for the augmented plant and controller may now be constructed. The plant factorizations may be given in terms of the previous factorizations, recalling that the signal y_w is given by (4.4.4), as follows.

$$\mathcal{G}(x_0) \quad = \quad \begin{bmatrix} G(x_0) \\ I \end{bmatrix} \quad = \quad \mathcal{N}(x_0)\mathcal{M}(x_0)^{-1} \quad = \quad \tilde{\mathcal{M}}(x_0)^{-1}\tilde{\mathcal{N}}(x_0)$$

$$\mathcal{M}(x_0) \quad = \quad M(x_0) \qquad\qquad \mathcal{N}(x_0) \quad = \quad \begin{bmatrix} N(x_0) \\ M(x_0) \end{bmatrix}$$

$$\tilde{\mathcal{N}}(x_0) \quad = \quad \begin{bmatrix} \tilde{N}(x_0) \\ I \end{bmatrix} \qquad\qquad \tilde{\mathcal{M}}(x_0) \quad = \quad \begin{bmatrix} \tilde{M}(x_0) & 0 \\ 0 & I \end{bmatrix}$$

A left factorization for the controller may be written

$$\mathcal{K}(x_0) \; = \; \tilde{\mathcal{V}}(x_0)^{-1}\tilde{\mathcal{U}}(x_0)$$

$$\tilde{\mathcal{V}}(x_0): \quad \begin{array}{rcl} \dot{x}_w & = & A(x_w)x_w + B(x_w)u \\ \dot{x}_{\tilde{V}} & = & (A(x_w) + H(x_w)C(x_w))x_{\tilde{V}} + (B(x_w) + H(x_w)D(x_w))u \\ s_2 & = & -F(x_w)x_{\tilde{V}} + u \end{array} \qquad \begin{array}{rcl} x_w & = & x_0 \\ x_{\tilde{V}}(0) & = & x_0/2 \end{array}$$

$$\tilde{U}(x_0): \quad \begin{array}{rcl} \dot{x}_w & = & A(x_w)x_w + B(x_w)u \\ \dot{x}_{\tilde{U}} & = & (A(x_w) + H(x_w)C(x_w))x_{\tilde{U}} - H(x_w)y \\ s_1 & = & F(x_w)x_{\tilde{U}} \end{array} \qquad \begin{array}{rcl} x_w & = & x_0 \\ x_{\tilde{U}}(0) & = & x_0/2 \end{array}$$

The main result of Chapter 3, Theorem 3.4, p. 50, may now be applied giving the class of all stabilizing plants and controllers as follows.

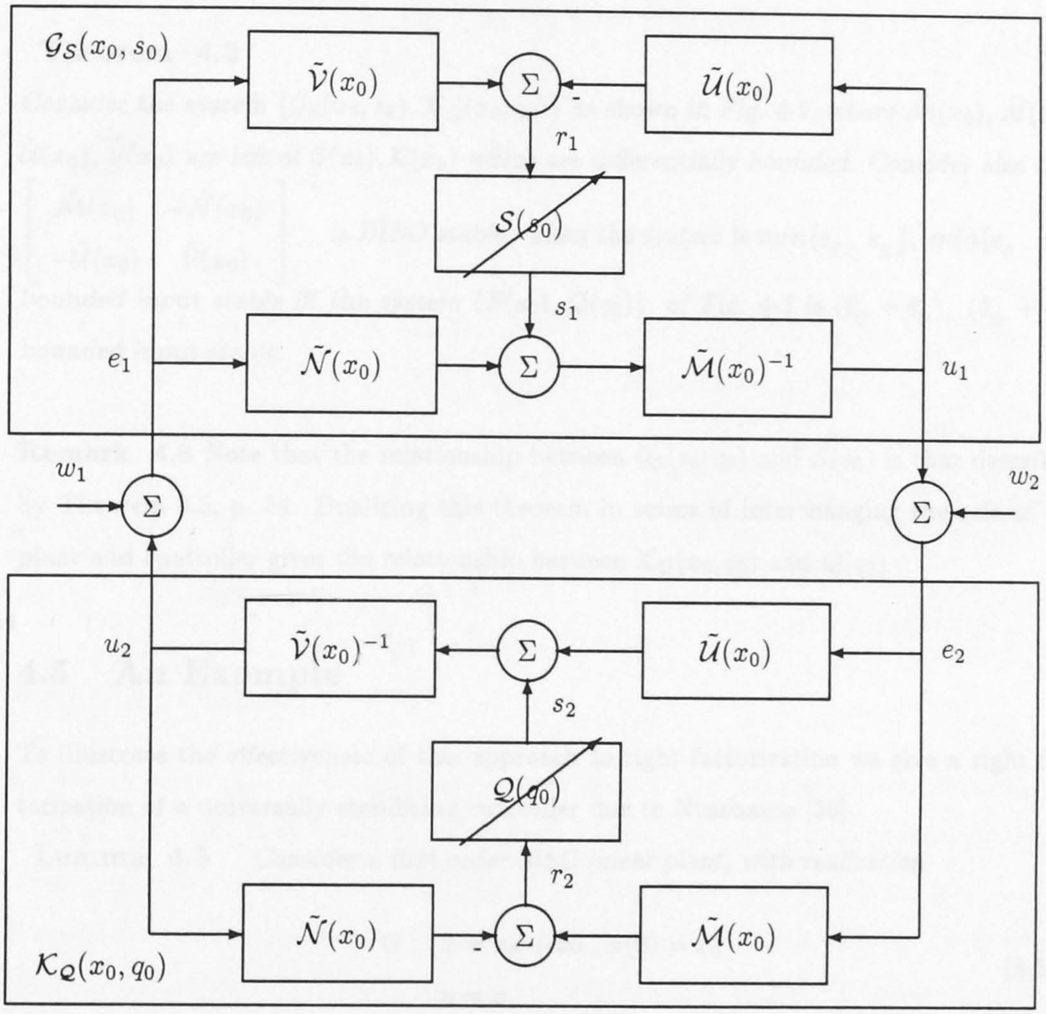
$\mathcal{G}_{\mathcal{S}}(x_0, s_0)$
$\tilde{\mathcal{V}}(x_0)$
Σ
$\tilde{\mathcal{U}}(x_0)$
r_1
$\mathcal{S}(s_0)$
s_1
e_1
$\tilde{\mathcal{N}}(x_0)$
Σ
$\tilde{\mathcal{M}}(x_0)^{-1}$
u_1
w_1
Σ
w_2
Σ
u_2
$\tilde{\mathcal{V}}(x_0)^{-1}$
Σ
$\tilde{\mathcal{U}}(x_0)$
e_2
s_2
$\mathcal{Q}(q_0)$
r_2
$\tilde{\mathcal{N}}(x_0)$
-
Σ
$\tilde{\mathcal{M}}(x_0)$
$\mathcal{K}_{\mathcal{Q}}(x_0, q_0)$

Figure 4-2: The feedback system $\{\mathcal{G}_{\mathcal{S}}(s_0, q_0), \mathcal{K}_{\mathcal{Q}}(x_0, q_0)\}$.

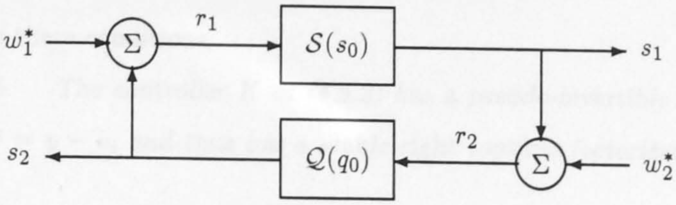

Figure 4-3: The feedback system $\{\mathcal{S}(s_0),\, \mathcal{Q}(q_0)\}$.

Theorem 4.2

Consider the system $\{\mathcal{G}_{\mathcal{S}}(x_0, s_0),\, \mathcal{K}_{\mathcal{Q}}(x_0, q_0)\}$ *as shown in Fig. 4-2, where* $\tilde{\mathcal{M}}(x_0)$, $\tilde{\mathcal{N}}(x_0)$, $\tilde{\mathcal{U}}(x_0)$, $\tilde{\mathcal{V}}(x_0)$ *are lcfs of* $\mathcal{G}(x_0), \mathcal{K}(x_0)$ *which are differentially bounded. Consider also that* $\begin{bmatrix} \tilde{\mathcal{M}}(x_0) & -\tilde{\mathcal{N}}(x_0) \\ -\tilde{\mathcal{U}}(x_0) & \tilde{\mathcal{V}}(x_0) \end{bmatrix}^{-1}$ *is BIBO stable. Then the system is* $min\{\varepsilon_{\tilde{\mathcal{V}}},\, \varepsilon_{\tilde{\mathcal{N}}}\}$, $min\{\varepsilon_{\tilde{\mathcal{U}}},\, \varepsilon_{\tilde{\mathcal{M}}}\}$ *bounded input stable iff the system* $\{\mathcal{S}(s_0),\, \mathcal{Q}(q_0)\}$, *of Fig. 4-3 is* $(\theta_{\tilde{\mathcal{U}}} + \theta_{\tilde{\mathcal{V}}})$, $(\theta_{\tilde{\mathcal{M}}} + \theta_{\tilde{\mathcal{N}}})$ *bounded input stable.* □

Remark 4.8 Note that the relationship between $\mathcal{G}_{\mathcal{S}}(s_0, q_0)$ and $\mathcal{S}(s_0)$ is that described by Theorem 3.5, p. 54. Dualizing this theorem in terms of interchanging the role of the plant and controller gives the relationship between $\mathcal{K}_{\mathcal{Q}}(x_0, q_0)$ and $\mathcal{Q}(q_0)$

4.5 An Example

To illustrate the effectiveness of this approach to right factorization we give a right factorization of a universally stabilizing controller due to Nussbaum [39].

Lemma 4.5 *Consider a first order SISO linear plant, with realization*

$$\begin{aligned} G: \quad \dot{x} &= ax + bu \quad x(0) = x_0 \\ y &= x \end{aligned} \tag{4.5.1}$$

Where $b \neq 0$. *Then there exists a nonlinear controller which will stabilize this plant for all values of* a *and* b. *The state equations are:*

$$\begin{aligned} K: \quad \dot{v} &= y(v^2 + 1) \qquad v(0) = 0 \\ u &= y(v^2 + 1)h(v) \end{aligned} \tag{4.5.2}$$

□

The proof of this lemma may be found in [39]. The function $h(\cdot)$ must satisfy certain conditions to ensure convergence, see [39] for details. Note that the function $h(x) =$

e^x *cos* x satisfies these conditions.

Lemma 4.6 *The controller K of (4.5.2) has a pseudo-invertible stabilizing state feedback, $g(v, y) = y - v$, and thus has a stable right coprime factorization $K = UV^{-1}$ given by*

$$U: \begin{array}{ll} \dot{x}_u = (s - x_u)(x_u^2 + 1) & x_u(0) = 0 \\ u = s(x_u^2 + 1)h(x_u) & \end{array} \tag{4.5.3}$$

$$V: \begin{array}{ll} \dot{x}_v = (s - x_v)(x_v^2 + 1) & x_v(0) = 0 \\ y = s - x_v & \end{array} \tag{4.5.4}$$

□

Proof. It is straightforward to see that $g(\cdot, \cdot)$ is pseudo-invertible. It is now shown that $\dot{v} = (y - v)(v^2 + 1)$ is stable. First note that $\dot{v} = 0$ iff $y = v$. If $v < y$, then $\dot{v} > 0$, so v will grow to converge to y. If $v > y$, then $\dot{v} < 0$, so v will converge down to y. Hence v will track y, and so if y is stable v will be stable. Now apply Lemma 4.3 to show that UV^{-1} is a stable right factorization of K. Following Remark 4.4 we note that this is a rcf. ■

4.6 Conclusion

In this chapter we have described a state space approach to the factorization of nonlinear systems. This has been based on attempting to find right and left factorizations which allow application of the theory of Chapters 2 and 3 to give a factorization theory. Additionally, by developing a general framework for the algebra of nonlinear operators we allow for the possibility of extending the results from the input-output approach.

It has been shown that a continuous time nonlinear system of the form of (4.2.1) will have a right factorization if there exists a solution to the smooth stabilization problem, Assumption 4.3.1. Taking advantage of this, a stabilizing controller is given, based on finding a state estimator for the plant, as in Assumption 4.3.2. This controller also has a right factorization. Left factorizations do not appear to follow from such a simple assumption. However results from Moore and Irlicht [36] are presented as one successful approach to the problem of finding a left factorization for a nonlinear plant $G(x_0)$.

As support for our approach to factorization a right factorization of the universally stabilizing controller of Nussbaum [39] is presented.

Chapter 5

Adaptive Nonlinear Estimation with Artificial Neural Networks

5.1 Introduction

The questions addressed in this chapter concern the applicability and limitations of Recursive Prediction Error (RPE) methods to the adaptive identification, estimation, prediction and control of uncertain nonlinear dynamic signal models formulated in terms of Artificial Neural Networks (ANNs). As seen in the previous chapter, if factorizations for a given nonlinear plant are to be derived, a stabilizing static state feedback map, $g(\cdot,\cdot)$, satisfying Assumption 4.3.1, p. 68, must be found. In order to find such a map, an accurate estimate of the state equation, $f(\cdot,\cdot)$, of (4.2.1) is required. Hence it is necessary to consider some form of nonlinear system identification.

Functional representations in artificial neural networks (ANNs) are now quite common in the literature. The question arises whether such representations, with their attractive training capabilities, could be useful in the representation and identification of uncertain nonlinear dynamical systems. In this chapter we formulate quite a general class of nonlinear dynamical systems in terms of neural networks with weights which may be adaptively adjusted by standard, well studied recursive prediction error (RPE) methods.

Identification theory and methodology has most to say about linear, stable systems with input-output descriptions. A key property is that the measurements are linear in the input-output model parameters, so that least squares methods apply, as well as the more general RPE methods and related extended Kalman filter (EKF) based methods. Where some *a priori* knowledge is available state space matrices characterized in terms of some vector θ, perhaps in a nonlinear fashion, may be used to represent the system. In this case RPE or EKF based methods [33, 32] are still applicable. When the model class is nonlinear in states and/or inputs, as well as in the uncertain parameters, there must be great caution in applying such identification methods. The scheme must satisfy the convergence requirements of the theory, and additionally the general approximation scheme which is used to represent the nonlinearities must be decided upon. With the growing application of ANNs, there is a building up of confidence in the role of ANNs as general purpose nonlinear function representations. Hence, it makes sense to investigate the possible role that they might play in adaptive nonlinear filtering via the existing RPE methods.

In Section 5.2 we give some background on the RPE approach to system identification, giving a thorough treatment of the conditions which must be satisfied to give convergence of the algorithm. The theorems which specify the convergence results possible are also stated. The ANN structure that is to be used is detailed in Section 5.3.1. In Section 5.4, the algorithm which is used to estimate the system is given, and convergence issues are discussed. Restrictions on the range of parameters which will guarantee convergence of the algorithm are given. Conclusions are drawn in Section 5.6.

5.2 The RPE problem formulation

5.2.1 Model structure.

Let us first recall formulations for linear stable systems.

When there is some *a priori* knowledge of the stable system dynamics, it is common to work with state space models of the following form:

$$x_{t+1} = A(\theta)x_t + B(\theta)u_t + K(\theta)w_t \qquad (5.2.1)$$

$$y_t = C(\theta)x_t + w_t \qquad (5.2.2)$$

Here the system matrices are expressed as known functions of an unknown parameter vector θ. Notice that since there is uncertainty in the model, we have chosen to work with an innovations representation in which the state process noise is identical to the measurement noise. Such representations are uniquely parametrized when the parametrizations $A(\cdot)$, $B(\cdot)$, $K(\cdot)$ and $C(\cdot)$ are unique.

We seek a parameter estimate, $\hat{\theta}$, which minimises a prediction error index. Recursive prediction error methods seek to achieve recursive estimates $\hat{\theta}_t$ so that in the limit as $t \to \infty$, $\hat{\theta}_t$ converges to the true θ, that of (5.2.1), (5.2.2). A prediction error index such as $V_t(\hat{\theta}) = \frac{1}{t}\sum_{i=1}^{t} \hat{w}_i^2(\hat{\theta})$, where $\hat{w}_t = y_t - \hat{y}_t$, the difference between the actual and estimated output of the plant, is appropriate and RPE methods can be applied to achieve asymptotically optimal estimates in terms of this index.

For nonlinear stable systems, rather than work with the most general formulations, we work here with a natural generalization of the class of models (5.2.1), (5.2.2).

$$x_{t+1} = A(\theta, x_t) + B(\theta, u_t) + K(\theta, \omega_t) \tag{5.2.3}$$

$$y_t = C(\theta, x_t) + \omega_t \tag{5.2.4}$$

Such a model class may also be called an innovations representation. In the linear case these parameterizations are uniquely parametrized. This property may carry over to the nonlinear case, however in general it may not be true that (5.2.4) gives a unique representation when the parametrizations $A(\cdot,\cdot)$, $B(\cdot,\cdot)$, $K(\cdot,\cdot)$, $C(\cdot,\cdot)$ are unique. Of course, starting with such model classes avoids questions concerning actual signal generating systems as to whether or not there is an associated innovations representation.

The specific class of nonlinear systems of the form of (5.2.3), (5.2.4) which are studied in this chapter, are those for which the nonlinear functions $A(\theta, x_t)$, $B(\theta, u_t)$, $K(\theta, w_t)$, $C(\theta, x_t)$ are ANNs, parametrized by the vector θ, and driven by inputs x_t, u_t, w_t, respectively. The system model may then be naturally generated by substituting an estimated parameter value $\hat{\theta}$ for the actual parameter value, as follows:

$$\hat{x}_{t+1} = A(\hat{\theta}_t, \hat{x}_t) + B(\hat{\theta}_t, u_t) + K(\hat{\theta}_t, \hat{\omega}_t) \tag{5.2.5}$$

$$\hat{y}_t = C(\hat{\theta}_t, \hat{x}_t) \tag{5.2.6}$$

$$\hat{\omega}_t = y_t - \hat{y}_t \tag{5.2.7}$$

Since ANNs are typically functional representations with a large number of weights, it is expected that the dimension of θ is large. The key question of interest is whether or not such representations are useful general nonlinear systems representations for adaptive identification or estimation of uncertain nonlinear systems with no *a priori* constraints or knowledge of system nonlinearities.

5.2.2 General Algorithm structure.

Recursive prediction error approaches such as those in Weiss and Moore [60], Moore and Boel [38] or Ljung [31] have been formulated to take advantage of the convergence theory for stochastic algorithms developed by Ljung [30, 33]. This formulation assumes the basic recursive algorithm can be written as:

$$\zeta_t = \zeta_{t-1} + \gamma_t Q(t, \zeta_{t-1}, \phi_t) \tag{5.2.8}$$

The vector $\zeta_t \in R^n$ contains the unknown parameters θ_t as well as additional elements to be discussed below. The vector $\phi_t \in R^m$, which includes the dynamic system state, is generated by the equation:

$$\phi_t = A(\zeta_{t-1})\phi_{t-1} + B(\zeta_{t-1})e_t \tag{5.2.9}$$

or more generally:

$$\phi_t = g(\phi_{t-1}, \zeta_{t-1}, e_t, t) \tag{5.2.10}$$

where $e_t \in R^r$ contains the system inputs which may be stochastic. This equation is known as the observer equation, and the ϕ_t are referred to as the observations.

Ljung [30] provides three sets of assumptions under which the system (5.2.8), (5.2.9) converges. The first two are for e_t a stochastic process and one for when it is deterministic sequence. In [29] Ljung gives another set of assumptions which give convergence for the more general form (5.2.10). As the ANN representation requires the observer equation to have the form (5.2.10), this last set of assumptions is the set which will be used.

The assumptions require exponential stability of (5.2.10) in some domain D_S, which may be the entire space, and that the sequence ϕ_t is bounded for all $\zeta \in D_S$. It may then be shown that convergence of (5.2.8) is governed by that of an ordinary differential equation of the form:

$$\frac{d}{d\tau}\zeta_\tau^D = f_D(\zeta_\tau^D). \tag{5.2.11}$$

The sequence ζ_t is shown to be close to the solution $\zeta(\tau)$ of (5.2.11) in some sense, and it is shown that the possible convergence points of (5.2.8) are exactly those stationary points of (5.2.11).

As these assumptions form the basis for the convergence theory for our RPE scheme, they are now presented from Ljung [29].

5.2.3 Assumptions for the convergence of the Algorithm (5.2.8) (5.2.10)

In this section the assumptions on (5.2.8) and (5.2.10) which allow the convergence theory developed by Ljung to be applied are detailed. These assumptions are taken from Ljung [29]. Theorems relating the convergence of the algorithm and the stationary points of (5.2.11) are also presented.

Firstly for some domain of ζ, D_R which is to be determined later, assume:

$$|g(\phi, \zeta, e, t)| < C, \quad \forall \phi, \ e, \ \forall \zeta \in D_R \tag{5.2.12}$$

The constant C may be dependent on D_R. Further, assume that

$Q(t, \zeta, \phi)$ is continuously differentiable with respect to ζ and ϕ, and the derivatives are bounded in t, $\forall \zeta \in D_R$. (5.2.13)

$g(\phi, \zeta, e, t)$ is continuously differentiable with respect to ζ, $\forall \zeta \in D_R$. (5.2.14)

The sequence $\bar{\phi}_t(\bar{\zeta})$ is now defined. This equation gives the dynamics of (5.2.10) when the parameter ζ is held constant, or "frozen" at the value $\bar{\zeta}$.

$$\bar{\phi}_t(\bar{\zeta}) = g(\bar{\phi}_{t-1}(\bar{\zeta}), \bar{\zeta}, e_t, t) \qquad \bar{\phi}_0(\bar{\zeta}) = 0 \tag{5.2.15}$$

This equation will be referred to as the *frozen observer equation.*

It is also assumed that $g(\phi, \zeta, e, t)$ has the property that given $\bar{\phi}_n(\bar{\zeta}) = \phi_n$:

$$|\bar{\phi}_t(\bar{\zeta}) - \phi_t| < C. \max_{n \leq k < t} |\bar{\zeta} - \zeta_k| \tag{5.2.16}$$

This implies that small variations in ζ will not be amplified by the action of the observer ϕ.

The domain of exponential stability of (5.2.10) is now defined. Let $\bar{\phi}_t^i(\bar{\zeta})$, $i = 1,\ 2$, be the solutions of (5.2.15), where $\bar{\phi}_s^i(\bar{\zeta}) = \phi_0^i$. Then,

$$D_S = \left\{ \bar{\zeta} : |\bar{\phi}_t^1(\bar{\zeta}) - \bar{\phi}_t^2(\bar{\zeta})| < M(\phi_0^1, \phi_0^2)\lambda^{t-s}(\bar{\zeta}), \ \ \forall t > s \right\} \tag{5.2.17}$$

where $0 < \lambda(\bar{\zeta}) < 1$. In the sequel D_R is taken to be an open, connected subset of D_S. The averaged form of (5.2.8) is now defined.

$$f_D(\bar{\zeta}) = \lim_{t \to \infty} E\left\{ Q(t, \bar{\zeta}, \bar{\phi}_t(\bar{\zeta})) \right\} \tag{5.2.18}$$

The expectation $E\{\cdot\}$ is taken over e_t. We now make the following assumptions on the random variable e_t, and the sequence γ_t.

$$e_t \text{ is a sequence of independent random variables.} \tag{5.2.19}$$

$$\sum_{t=1}^{\infty} \gamma_t = \infty \tag{5.2.20}$$

$$\sum_{t=1}^{\infty} \gamma_t^p = \infty, \text{ for some integer } p > 1 \tag{5.2.21}$$

$$\gamma_t \text{ is decreasing} \tag{5.2.22}$$

$$\lim_{t\to\infty} \sup \left[\frac{1}{\gamma_t} - \frac{1}{\gamma_{t-1}}\right] < \infty \tag{5.2.23}$$

Under these assumptions Lemma 1 and Theorems 1-6 of Ljung [29] will hold. Some of these results are re-stated here, thus formally relating the convergence of the differential equation (5.2.11) to the evolution of the algorithm (5.2.8), (5.2.10). Note that these results may be also proved under any of the assumption sets of Ljung [30].

Theorem 5.1 Theorem 1, [29]

Consider the algorithm (5.2.8), (5.2.10), subject to the assumptions (5.2.12)-(5.2.23). Let $\bar{D}$ be a compact subset of D_R such that the trajectories of (5.2.11) that start in $\bar{D}$ remain in a closed subset of D_R for all $\tau > 0$. Assume that there is a random variable C such that:

$$\zeta(t) \in \bar{D} \text{ and } |\phi(t)| < C \text{ infinitely often, with probability 1} \tag{5.2.24}$$

and that:

$$\text{the differential equation (5.2.11) has an invariant set } D_c \text{ with domain of attraction } D_A \supset \bar{D}. \tag{5.2.25}$$

Then $\zeta(t) \to D_c$ with probability one as $t \to \infty$. □

Remark 5.1 An interesting case is when the set D_c is a stationary point, $\zeta^\star$ of (5.2.11). In this case the theorem gives convergence of $\zeta(t)$ to $\zeta^\star$.

Remark 5.2 The assumption (5.2.24) is known as a *boundedness condition.* It forces the algorithm to remain within the exponentially stable region D_R, so that the differential equation (5.2.11) is a valid representation of the algorithms behaviour. However, although analytically tractable, this condition may be difficult to guarantee in practice. One alternative is to project ζ back into the exponential stability domain D_R whenever the update would force it out. This is formalized in the following theorem.

Theorem 5.2 Theorem 3, [29]

Consider the algorithm (5.2.8), (5.2.10), subject to the assumptions (5.2.12)-(5.2.23), where (5.2.8) has been modified to

$$\zeta_t = [\zeta_{t-1} + \gamma_t Q(t, \zeta_{t-1}, \phi_t)]_{D_1,\, D_2} \tag{5.2.26}$$

$$[f]_{D_1,\, D_2} = \begin{cases} f & \text{if } f \in D_1 \\ x \in D_2 & \text{if } f \notin D_1 \end{cases}$$

where $D_1 \subset D_R$ is an open, bounded set containing the compact set D_2. Define $\tilde{D} = D_1 \backslash D_2$. Further assume that $D_2 \subset D_A \subset D_S$, with D_A as defined in Theorem 5.1. Suppose that there exists a twice differentiable function $U(x) \geq 0$ defined in a neighbourhood of $\tilde{D}$ with the properties:

$$\sup_{x \in \tilde{D}} U'(x) f(x) < 0 \tag{5.2.27}$$

$$U(x) \geq C_1 \qquad \text{for } x \notin D_1 \tag{5.2.28}$$

$$U(x) \leq C_2 < C_1 \qquad \text{for } x \in D_2 \tag{5.2.29}$$

Then Theorem 5.1 holds without (5.2.24). □

Remark 5.3 The function $U(x)$ is made a Lyapunov function in $\tilde{D}$ by (5.2.27), so the trajectories of (5.2.11) will converge to D_2. The intuitive notion of the trajectories from D_2 never leaving D_1 is formalized by (5.2.28) and (5.2.29). These equations will hold if the trajectories of (5.2.11) only pass through the boundary of D_1 when entering D_1, and D_2 is sufficiently close to D_1.

Conditions which guarantee the convergence of the algorithm have now been stated. The relationship between the stationary points of the differential equation (5.2.11) and the limit points of (5.2.8) is now investigated. The following theorem proves that the possible convergence points of the algorithm are exactly those of the differential equation.

Theorem 5.3 Theorem 4, [29]

Consider the algorithm (5.2.8), (5.2.10), subject to the assumptions (5.2.12)-(5.2.23). Suppose that $\zeta^\star \in D_R$ has the property that:

$$P(\zeta(t) \to \mathcal{B}(\zeta^\star, \rho)) > 0 \qquad \forall \rho > 0 \tag{5.2.30}$$

where $\mathcal{B}(\zeta, \rho)$ is the ball of radius ρ about ζ. Further suppose that

$Q(t, \zeta^\star, \bar{\phi}(t, \zeta^\star))$ has a covariance matrix bounded from below $Q_0 > 0$. (5.2.31)

and that

$E\{Q(t, \zeta, \bar{\phi}(t, \zeta))\}$ is continuously differentiable with respect to ζ in a neighbourhood of $\zeta^\star$, and the derivatives converge uniformly in this neighbourhood as $t \to \infty$ (5.2.32)

Then $f(\zeta^\star) = 0$, and $H(\zeta^\star) = \frac{d}{d\zeta} f(\zeta)\Big|_{\zeta=\zeta^\star}$ has all eigenvalues in the left half plane. □

Remark 5.4 The matrix $H(\zeta^\star)$ is the matrix obtained for the linearization of $f_D(\cdot)$ about the point $\zeta^\star$. Hence the theorem states that the algorithm can only converge to stable points of the differential equation (5.2.11).

The following theorem relates the trajectories of the differential equation (5.2.11) to the paths of the algorithm, (5.2.8). This is done by comparing the values of ζ_t with the values of the solution to (5.2.11) at the times τ_t, given by:

$$\tau_i = \sum_{k=1}^{t} \gamma_k \qquad (5.2.33)$$

Let ζ_t be the solution to (5.2.8), (5.2.10), and let $\zeta^D(\tau)$ be the solution of (5.2.11) with initial value $\zeta(t_0)$ at time τ_{t_0}. Let I be a set of integers, then the probability that all points ζ_t, $t \in I$ are within a certain distance ϵ from the trajectory is given by the following theorem.

Theorem 5.4 Theorem 6, [29]

Consider the algorithm (5.2.8), (5.2.10), subject to the assumptions (5.2.12)-(5.2.23). Assume that $f(\zeta)$ is continuously differentiable, and that (5.2.24) holds. Assume that the solutions to (5.2.11) with initial conditions in $\bar{D}$ are exponentially stable, and let I be a set of integers, such that $\inf |\tau_i - \tau_j| = \delta_0$ where $i \neq j$, i, $j \in I$. Then for any $p \geq 1$ there exist constants K, ϵ_0 and T_0 that depend on p, D and δ_0, such that for $\epsilon < \epsilon_0$ and $t_0 > T_0$:

$$P\left\{\sup_{t \in I,\ t \geq t_o} |\zeta_t - \zeta^D(t)| > \epsilon\right\} \leq \frac{K}{\epsilon^{4p}} \sum_{j=t_0}^{N} \gamma_j^p \qquad (5.2.34)$$

where $N = \sup_{i \in I} i$, which may be infinite. □

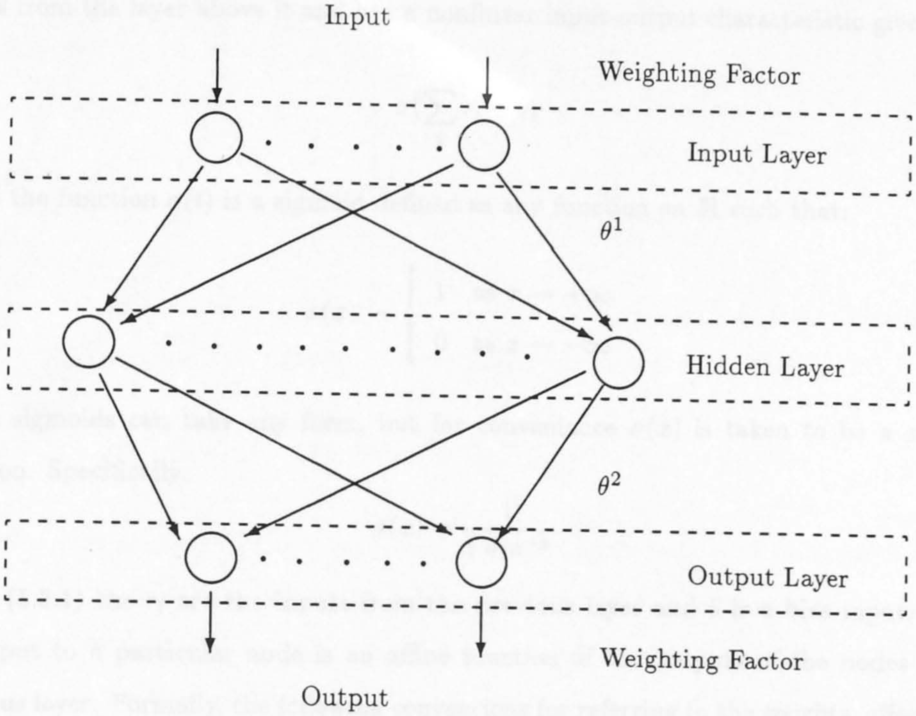

Figure 5-1: The structure of a three layer ANN.

As has been seen the convergence properties for the system of (5.2.8), (5.2.10) are dependent on the convergence of (5.2.11). Further the existence of local minima may be predicted by considering the fixed points of $f_D(\cdot)$. It is not difficult to construct RPE schemes so that (5.2.11) is guaranteed to converge. The other conditions may be less straightforward to guarantee.

Once the RPE scheme has been set up, the results of this section may be applied to give conditions for the convergence of the RPE algorithm when used with ANNs for nonlinear system identification.

5.3 Artificial Neural Networks

5.3.1 Network Architecture

The term "Artificial Neural Network" (ANN) refers to a highly connected array of elementary processors referred to as neurons, or nodes. The standard configuration for a three layer network having is to have an input layer, one hidden layer and an output layer, as shown in Figure 5-1. Note, however that an ANN may have any number of hidden layers, and commonly the nodes in the input layer are the identity. Each node only receives inputs from the layer above it and has a nonlinear input-output characteristic given by:

$$\sigma(\sum_i r_i + b) \tag{5.3.1}$$

where the function $\sigma(t)$ is a sigmoid defined as any function on $\mathbb{R}$ such that:

$$\sigma(x) \rightarrow \begin{cases} 1 & \text{as } x \rightarrow +\infty \\ 0 & \text{as } x \rightarrow -\infty \end{cases} \tag{5.3.2}$$

These sigmoids can take any form, but for convenience $\sigma(x)$ is taken to be a smooth function. Specifically:

$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{5.3.3}$$

In (5.3.1) the r_i are the inputs from the previous layer and b is a bias input. Thus the input to a particular node is an affine function of the outputs of the nodes of the previous layer. Formally, the following conventions for referring to the weights, offsets and the inputs and outputs of the layers for an N layer network are adopted.

Denote the output from the i^{th} layer by s^i, and the output of the j^{th} node in the i^{th} layer by s^i_j, the input to the j^{th} node in the i^{th} layer is denoted r^i_j. The input to the net is considered to be the output of the 0^{th} layer, s^0. Denote the weights matrix from the $(i-1)^{th}$ layer to the i^{th} layer by θ_i, and the j^{th} row of this matrix as θ^i_j. The offset to each node is accounted for by setting the last node in each layer always equal to 1, thus the offset to the j^{th} node of the i^{th} layer is given by the last component of θ^i_j. Hence, if n_i is the number of nodes in the i^{th} layer of the network, $dim(s^i) = n_i + 1$, $dim(r^i) = n_i$, and θ^i is a $(n_{i-1}+1) \times n_i$ matrix. The exception to this is the final layer, where the extra node with output 1 is not added, as this is the output of the ANN.

Hence the inputs and outputs of the layers of the network will be given by:

$$s^0 = \begin{pmatrix} x \\ 1 \end{pmatrix} \tag{5.3.4}$$

$$y = s^N \tag{5.3.5}$$

$$r^i = \theta^i s^{i-1} \tag{5.3.6}$$

$$s^i = F_i(r^i) \tag{5.3.7}$$

$$F_i \qquad \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_{n_i} \end{pmatrix} \mapsto \begin{pmatrix} \sigma(x_1) \\ \sigma(x_2) \\ \vdots \\ \sigma(x_{n_i}) \\ 1 \end{pmatrix} \tag{5.3.8}$$

The exception to the above is that the "1" in the output vector of $F_N(\cdot)$ is suppressed.

Thus the output y, of a general N layer feedforward ANN is given by:

$$y \;=\; F_N(\theta^N \cdot F_{N-1}(\theta^{N-1} \cdot F_{N-2}(\ldots(\theta^2 \cdot F_1(\theta^1, x))\ldots))) \tag{5.3.9}$$

In the special case where the vector x represents the n inputs to a 3-layer ANN, we can represent the total output of a network with r nodes in the hidden layer and m outputs by $G(\theta, x)$ where:

$$G(\theta, x) \;=\; F_2(\theta^2 \cdot F_1(\theta^1 \cdot x)) \tag{5.3.10}$$

5.3.2 Functional representation and training using ANNs

We are interested in using an ANN to approximate a function of many variables, $f(x)$. The weights and offsets contained in the ANN provide a convenient mode of parameterization, which we may use in our RPE scheme. Hornik, Stinchcombe and White [22] (see also [46]) have proved a general representation theorem for a class of Neural Networks called $\Sigma\Pi$ networks. The class of neural networks described in the previous section is a special case of this class of ANNs, called $\Sigma^r(G)$ networks. In [22] it is shown that our class of ANNs are able to approximate any continuous function arbitrarily well provided that there are enough nodes in the hidden layer. Similar results were also obtained by Cybenko [4] and Funahashi [11] using different methods of proof. For our purposes, it is not vital to consider the details of these proofs, apart from noting that the proofs were non-constructive. That is, they are purely existence proofs: they say nothing about the number of nodes needed in order to achieve a certain approximation accuracy.

To train our ANNs we will use an approach introduced by Werbos [61], known as Back-Propagation. This is essentially a gradient descent technique based on using the chain rule to find the change of the output of the net with respect to the change of the parameters. This is a useful approach for our purposes as the algorithm uses gradient information to calculate the parameter updates. The details of how to differentiate an ANN of the structure (5.3.9) with respect to the parameter and input vectors are presented in Appendix B.1.

5.4 Nonlinear RPE Problem Using ANNs

In the previous sections the properties of ANNs and the RPE approach to estimation were outlined. Specifically, RPE problems have been provided with a general theory for convergence of unknown parameters, while ANNs have shown to have strong function representation properties. Both schemes are compatible in terms of the RPE algorithm requiring differentiability with respect to the parameters, and the ANNs being differentiable with respect to the parameters and inputs. In this section we show that these two schemes may be combined, leading to an effective approach to nonlinear system identification.

5.4.1 Formulation of the Algorithm

We assume an unknown nonlinear system driven by an input u_t and a noise process ω_t and having output y_t as in (5.2.3), (5.2.4). It is assumed that it is possible to use ANNs to parameterize each of the nonlinearities. The estimating system is as given in (5.2.5)-(5.2.7), where the $A(\cdot,\cdot)$, $B(\cdot,\cdot)$, $K(\cdot,\cdot)$ and $C(\cdot,\cdot)$ are ANNs. Within the algorithm, this becomes (5.4.1) and (5.4.2). Thus the problem is to find an algorithm of the form of (5.2.8), (5.2.10), which can incorporate this model, and estimate the parameter θ of (5.2.3), (5.2.4).

Following the customary description of RPE problems, Ljung [31, 32, 33], Weiss and Moore [60], Moore and Boel [38], the RPE scheme of (5.4.1)-(5.4.9) is constructed. Before stating the algorithm, a short description is given so as to indicate the action of each of the equations.

The parameter vector estimate at time t, $\hat{\theta}_t$, is updated by the system given in (5.4.3)-(5.4.4). Here $\hat{\omega}_{t/t-1}$ is the predicted error, the error obtained using the current state estimate but the lagged parameter estimate, as shown in (5.4.5). The matrix R_t is positive definite and bounded from below by δI where δ is some small positive constant and $R_0 = \delta I$. The sequence γ_t must adhere to the convergence requirements of Ljung [30], (5.2.20)-(5.2.23) The prediction error sensitivity function $\psi_t^T \triangleq -\frac{\partial}{\partial\alpha}\hat{\omega}_{t/t-1}(\alpha) \mid_{\alpha=\hat{\theta}_{t-1}}$ and related state sensitivity function $W_t \triangleq \frac{\partial}{\partial\alpha}\hat{x}_t(\alpha) \mid_{\alpha=\hat{\theta}_{t-1}}$ are updated as in equations (5.4.6) and (5.4.7), where F_t and G_t are intermediate variables defined in (5.4.8) and (5.4.9).

The algorithm is stated as follows:

$$\hat{x}_t = A(\hat{\theta}_{t-1}, \hat{x}_{t-1}) + B(\hat{\theta}_{t-1}, u_{t-1}) + K(\hat{\theta}_{t-1}, \hat{\omega}_{t-1}) \tag{5.4.1}$$

$$\hat{\omega}_{t-1} = y_{t-1} - C(\hat{\theta}_{t-1}, \hat{x}_{t-1}) \tag{5.4.2}$$

$$\hat{\theta}_t = \hat{\theta}_{t-1} + \gamma_t R_{t-1}^{-1} \psi_t \hat{\omega}_{t/t-1} \tag{5.4.3}$$

$$R_t = R_{t-1} + \gamma_t (\psi_t \psi_t^T + \delta I - R_{t-1}) \tag{5.4.4}$$

$$\hat{\omega}_{t/t-1} = y_t - C(\hat{\theta}_{t-1}, \hat{x}_t) \tag{5.4.5}$$

$$\psi_t = C_2(\hat{\theta}_{t-1}, \hat{x}_t) W_t + C_1(\hat{\theta}_{t-1}, \hat{x}_t) \tag{5.4.6}$$

$$W_t = F_t W_{t-1} + G_t \tag{5.4.7}$$

$$F_t = A_2(\hat{\theta}_{t-1}, \hat{x}_{t-1}) - K_2(\hat{\theta}_{t-1}, \hat{\omega}_{t-1}) C_2(\hat{\theta}_{t-1}, \hat{x}_{t-1}) \tag{5.4.8}$$

$$G_t = A_1(\hat{\theta}_{t-1}, \hat{x}_{t-1}) + B_1(\hat{\theta}_{t-1}, u_{t-1}) + K_1(\hat{\theta}_{t-1}, \hat{\omega}_{t-1}) - K_2(\hat{\theta}_{t-1}, \hat{\omega}_{t-1}) C_1(\hat{\theta}_{t-1}, \hat{x}_{t-1}) \tag{5.4.9}$$

Here a subscript j is used to denote differentiation with respect to the j-th variable, so that $C_2(\hat{\theta}_{t-1}, \hat{x}_t)$ is $\frac{\partial}{\partial x} C(\hat{\theta}_{t-1}, x)_{x=\hat{x}_t}$.

5.4.2 Requirements for the convergence of the algorithm.

By properly identifying the variables, this system of equations can be put into a form similar to (5.2.8), (5.2.10), and the results of Section 5.2.3 may be applied giving the conditions on (5.4.1)-(5.4.9) which guarantee convergence of the algorithm. Note that although it may be necessary to later modify the algorithm to project onto a stability domain, as in Theorem 5.2, the algorithm (5.2.8), (5.2.10) will be considered for the moment.

If ζ is defined as,

$$\zeta_t^T = (\hat{\theta}_t, colR_t)^T \tag{5.4.10}$$

where $colR_t$ denotes the vector formed by concatenating the columns of R_t, then (5.4.3) and (5.4.4) correspond to (5.2.8) by writing:

$$\zeta_t = \zeta_{t-1} + \gamma_t \left[\begin{array}{c} R_{t-1}^{-1} \psi_t \hat{\omega}_{t/t-1} \\ col(\psi_t \psi_t^T + \delta I - R_{t-1}) \end{array} \right] \tag{5.4.11}$$

From (5.4.5) and (5.4.6), note that $\hat{\omega}_{t/t-1}$ and ψ_t are functions of $\hat{\theta}_{t-1}$, $\hat{x}_t$, W_t and y_t. Thus (5.4.11) is of the form of (5.2.8), giving $Q(\zeta_{t-1}, \phi_t)$ provided that ϕ_t and $e(t)$ are appropriately defined. This may be done by defining ϕ and $e(t)$ as follows.

$$\phi_t^T = (\hat{x}_t, colW_t, y_t)^T \tag{5.4.12}$$

$$e_t^T = (u_{t-1}, y_t)^T \tag{5.4.13}$$

Note that under these definitions $\hat{w}_{t/t-1}$ and ψ_t are functions of ζ_{t-1} and ϕ_t, and can thus be considered to be part of the function $Q(\zeta_{t-1}, \phi_t)$. Similarly, $\hat{w}_{t-1}$, F_t and G_t are functions of u_{t-1}, y_{t-1}, ζ_{t-1} and ϕ_{t-1} and may thus be included in $g(\phi_{t-1}, \zeta_{t-1}, e_t)$. Equations (5.4.1), (5.4.7) thus correspond to (5.2.10).

$$\phi_t = \begin{bmatrix} A(\hat{\theta}_{t-1}, \hat{x}_{t-1}) + B(\hat{\theta}_{t-1}, u_{t-1}) + K(\hat{\theta}_{t-1}, y_{t-1} - C(\hat{\theta}_{t-1}, \hat{x}_{t-1})) \\ col(F_t W_{t-1} + G_t) \\ y_t \end{bmatrix} \tag{5.4.14}$$

The equations (5.4.11), (5.4.14) are thus of the form of (5.2.8), (5.2.10) when $Q(\zeta_{t-1}, \phi_t)$ and $g(\phi_{t-1}, \zeta_{t-1}, e_t)$ are defined by:

$$Q(\zeta_{t-1}, \phi_t) = \begin{bmatrix} R_{t-1}^{-1} \psi_t \hat{\omega}_{t/t-1} \\ col(\psi_t \psi_t^T + \delta I - R_{t-1}) \end{bmatrix} \tag{5.4.15}$$

$$g(\phi_{t-1}, \zeta_{t-1}, e_t) = \begin{bmatrix} A(\hat{\theta}_{t-1}, \hat{x}_{t-1}) + B(\hat{\theta}_{t-1}, u_{t-1}) + K(\hat{\theta}_{t-1}, y_{t-1} - C(\hat{\theta}_{t-1}, \hat{x}_{t-1})) \\ col(F_t W_{t-1} + G_t) \\ y_t \end{bmatrix} \tag{5.4.16}$$

where the auxiliary variables $\hat{\omega}_{t/t-1}$, ψ_t, F_t and G_t are given by (5.4.5), (5.4.6), (5.4.8) and (5.4.9) respectively.

As (5.4.11), (5.4.14) are of the form (5.2.8), (5.2.10) the results of Section 5.2.3 may now be applied to derive conditions on the algorithm which guarantee convergence of the algorithm. It is shown that under some restrictions the assumptions (5.2.12)-(5.2.23) may be satisfied. In the sequel the time dependency of the variables will be suppressed, unless it is necessary for the particular property being considered.

Consider (5.2.12), which requires that $g(\phi, \zeta, e)$ be bounded, $\forall \phi$, e $\forall \zeta \in D_R$. Note that ζ affects $g(\phi, \zeta, e)$ only through the action of θ within the ANNs. As the ANNs are bounded for all θ, ζ will not affect the boundedness of $g(\phi, \zeta, e)$. Similarly $\hat{x}$ only has an effect through the ANNs, and thus will not lead to unboundedness of $g(\phi, \zeta, e)$. However, note that $g(\phi, \zeta, e)$ is linear in W_t and y_t, thus it is not possible to guarantee (5.2.12) for all ϕ, e. Hence it is necessary to assume that W_t and y_t are bounded, so that (5.2.12) is satisfied.

As shall be seen later, it is necessary that the dynamics of (5.4.14) are stable, so it does not appear too restrictive to assume that W_t is bounded. Similarly, due to the assumptions on e_t, we must assume that the system being estimated is stable, giving boundedness of y_t.

Provided that R is bounded below by a constant, positive definite matrix it is straightforward to see that $Q(\zeta, \phi)$ is continuously differentiable with respect to ζ. As noted previously, with the update equation (5.4.4), R_t is bounded below by δI. Since $Q(\zeta, \phi)$ is independent of t, the derivatives are bounded with respect to t. Hence (5.2.13) is satisfied.

If $g(\phi, \zeta, e)$ is differentiable with respect to θ, (5.2.14) will be satisfied. Thus the satisfaction of this requirement is dependent on the differentiability of the ANNs with respect to the parameter vector θ. Note that F_t and G_t include first derivatives of the ANNs with respect to θ, hence if the ANNs used are twice differentiable with respect to θ (5.2.14) will be satisfied. Examining (5.3.10) shows that this is dependent on the smoothness of the sigmoid function, $\sigma(\cdot)$. As we are using a smooth function for $\sigma(\cdot)$, this assumption is trivially satisfied. See Appendix B.1 for details on the derivatives of an ANN with respect to it's parameter and input vectors.

The exact form of $\bar{\phi}$ is now stated, giving the dynamics of (5.4.14) when ζ is held fixed.

$$\left(\begin{array}{c} \bar{x}_t \\ col\bar{W}_t \\ y_t \end{array} \right) = \left[\begin{array}{l} A(\bar{\theta}, \bar{x}_{t-1}) + B(\bar{\theta}, u_{t-1}) + K(\bar{\theta}, y_{t-1} - C(\bar{\theta}, \bar{x}_{t-1})) \\ col(\bar{F}_t \bar{W}_{t-1} + \bar{G}_t) \\ y_t \end{array} \right] \tag{5.4.17}$$

The intermediate variables $\bar{F}_t$, $\bar{G}_t$ and $\bar{\omega}_{t-1}$ are defined as follows:

$$\bar{F}_t = A_2(\bar{\theta}, \bar{x}_{t-1}) - K_2(\bar{\theta}, \bar{\omega}_{t-1}) C_2(\bar{\theta}, \bar{x}_{t-1}) \tag{5.4.18}$$

$$\bar{G}_t = A_1(\bar{\theta}, \bar{x}_{t-1}) + B_1(\bar{\theta}, u_{t-1}) + K_1(\bar{\theta}, \bar{\omega}_{t-1}) - K_2(\bar{\theta}, \bar{\omega}_{t-1}) C_1(\bar{\theta}, \bar{x}_{t-1}) \tag{5.4.19}$$

$$\bar{\omega}_{t-1} = y_t - C(\bar{\theta}, \bar{x}_{t-1}) \tag{5.4.20}$$

The various properties of the frozen observer equation are now investigated. That

$g(\phi, \zeta, e)$ leads to the property (5.2.16) is not obvious. However if we assume that $g(\phi, \zeta, e)$ is Lipschitz continuous in each of the first two variables, then it is possible to show that (5.2.16) holds. This is more precisely stated in the following lemma.

Lemma 5.1 *Suppose that the function $g(\phi, \zeta, e, t)$ of (5.2.10) is uniformly Lipschitz continuous with respect to ϕ and ζ with Lipschitz constants L_1, L_2 respectively, and the sequence $\bar{\phi}_t$ is defined by (5.2.15). Then if $L_1 < 1$ and $C \geq L_2/(1 - L_1)$, equation (5.2.16) will hold.* □

Proof. If $g(\phi, \zeta, e, t)$ is uniformly Lipschitz continuous with respect to ϕ and ζ with Lipschitz constants L_1, L_2 respectively, then independently of t the following equations will hold.

$$|g(\phi_1, \zeta, e, t) - g(\phi_2, \zeta, e, t)| = L_1|\phi_1 - \phi_2| \tag{5.4.21}$$

$$|g(\phi, \zeta_1, e, t) - g(\phi, \zeta_2, e, t)| = L_2|\zeta_1 - \zeta_2| \tag{5.4.22}$$

That property (5.2.16) holds will now be proved by induction. Suppose at some time n, it is true that $\bar{\phi}_n = \phi_n$. Then applying (5.4.21) and (5.4.22) to (5.2.10) and (5.2.15) gives the following expression.

$$\begin{aligned} |\bar{\phi}_{n+1} - \phi_{n+1}| &= |g(\bar{\phi}_n, \bar{\zeta}, e_n, n) - g(\phi_n, \zeta_n, e_n, n)| && (5.4.23) \\ &< L_2|\bar{\zeta} - \zeta_n| && (5.4.24) \\ &= L_2 \max_{n \leq k < n+1} |\bar{\zeta} - \zeta_k| && (5.4.25) \end{aligned}$$

Now for some time $m > n$ consider that $\bar{\phi}_m \neq \phi_m$, and that:

$$|\bar{\phi}_m - \phi_m| < C \max_{n \leq k < m} |\bar{\zeta} - \zeta_k| \tag{5.4.26}$$

for some $C > 0$. We now calculate a bound on $|\bar{\phi}_{m+1} - \phi_{m+1}|$.

$$\begin{aligned} |\bar{\phi}_{m+1} - \phi_{m+1}| &= |g(\bar{\phi}_m, \bar{\zeta}, e_m) - g(\phi_m, \zeta_m, e_m)| \\ &= |g(\bar{\phi}_m, \bar{\zeta}, e_m) - g(\phi_m, \bar{\zeta}, e_m) + g(\phi_m, \bar{\zeta}, e_m) - g(\phi_m, \zeta_m, e_m)| \\ &\leq |g(\bar{\phi}_m, \bar{\zeta}, e_m) - g(\phi_m, \bar{\zeta}, e_m)| + |g(\phi_m, \bar{\zeta}, e_m) - g(\phi_m, \zeta_m, e_m)| \\ &< L_1|\bar{\phi} - \phi_m| + L_2|\bar{\zeta} - \zeta_m| \\ &< L_1 C \max_{n \leq k < m} |\bar{\zeta} - \zeta_k| + L_2|\bar{\zeta} - \zeta_m| \end{aligned}$$

$$\begin{aligned} &\leq L_1 C \max_{n \leq k < m+1} |\bar{\zeta} - \zeta_k| + L_2 \max_{n \leq k < m+1} |\bar{\zeta} - \zeta_k| \\ &= \ (L_1 C + L_2) \max_{n \leq k < m+1} |\bar{\zeta} - \zeta_k| \end{aligned}$$

Hence if $L_1 C + L_2 \leq C$, we have proved that:

$$|\bar{\phi}_m - \phi_m| < C \max_{n \leq k < m} |\bar{\zeta} - \zeta_k| \implies |\bar{\phi}_{m+1} - \phi_{m+1}| < C \max_{n \leq k < m+1} |\bar{\zeta} - \zeta_k| \qquad (5.4.27)$$

As the premise is true for $m = n + 1$, (5.4.26) will hold for all $m > n$, and (5.2.16) will be true.

Note that if $L_1 < 1$ and $C \geq L_2/1 - L_1$, then $L_1 C + L_2 \leq C$ and the proof is complete. ■

The assumptions of the lemma are not as restrictive as they may first appear. In our case $g(\phi, \zeta, e, t)$ is independent of t, so that if it is Lipschitz continuous, it will be uniformly Lipschitz continuous. Since ANNs with smooth activation functions are used to generate $g(\phi, \zeta, e)$, it will be smooth, and so it will be possible to define Lipschitz constants with respect to ζ and ϕ. Recall the definition of $g(\phi, \zeta, e)$ (5.4.16). For the following discussion denote the components of this equation g_1, g_2, g_3. These correspond to the update equations for $\hat{x}_t$, W_t and y_t respectively.

Note that g_3 is independent of $\hat{\theta}_t$, W_t and $\hat{x}_t$, and so is automatically Lipschitz continuous with constants $L_1 = 0$, $L_2 = 0$. The function g_1 consists entirely of the outputs of ANNs. It is shown in Appendix B.2 that ANNs are Lipschitz continuous in the inputs and parameters, and so it is possible to define Lipschitz constants for this component. These constants are Dependant on ζ and ϕ, so the previous lemma naturally introduces some restrictions on the domain of interest of ζ and ϕ.

The second component, g_2, does not admit such simple characterizations. This function is linear in W_t, but all the other parameters contribute through the first derivative of the output of an ANN. Specifically, $g_2 = F_t W_{t-1} + G_t$. The function F_t corresponds to the derivative of $\hat{x}_t$ with respect to $\hat{x}_{t-1}$, and G_t corresponds to the derivative of $\hat{x}_t$ with respect to $\hat{\theta}_{t-1}$. Thus, although it is straightforward to bound F_t and G_t thus obtaining the Lipschitz constants for g_1, it is very difficult to obtain bounds on the Lipschitz constants for g_2. Even in the case where the signals y_t, x_t, u_t, w_t are real valued sequences, the expressions are very complicated. Although it is possible to find Lipschitz constants in such cases, there does not appear to be any readily accessible underlying structure. Thus, although we note that it is possible to obtain Lipschitz constants for g_2, a simple expression is not currently obtainable.

Note that it is possible to absolutely bound the difference between W_t and $\bar{W}_t$ by $\max\{W_s,\ L_2/(1-L_1)\}$.

Thus the only restriction that the lemma imposes on our scheme is that the domain of ζ is such that $L_1 < 1$. Even this assumption is not restrictive as this restriction on $\bar{\zeta}$ is exactly that which guarantees that the evolution of $\bar{\phi}_t$ is stable, which is what is required to form the set D_S.

The following lemma leads to a sufficient condition on $\bar{\zeta}$ for $\bar{\zeta}$ to be in the set D_S. It proves that there is an exponential bound on the distance between trajectories of (5.2.15).

Lemma 5.2 *Suppose that $g(\phi, \zeta, e, t)$ is Lipschitz continuous in ϕ, with Lipschitz constant dependent on ζ.* i.e. *$\forall\zeta$ there exists a value $L(\zeta)$ such that*

$$|g(\phi_1, \zeta, e, t) - g(\phi_2, \zeta, e, t)| < L(\zeta)|\phi_1 - \phi_2| \tag{5.4.28}$$

Then for $\bar{\phi}^i_s(\bar{\zeta}) = \phi^i_0$, $i = 1,\ 2$ and $t > s$:

$$|\bar{\phi}^1_t(\bar{\zeta}) - \bar{\phi}^2_t(\bar{\zeta})| < M(\phi^1_0, \phi^2_0)\lambda^{t-s}(\bar{\zeta}) \tag{5.4.29}$$

Furthermore, $M(\phi^1_0, \phi^2_0) = |\phi^1_0 - \phi^2_0| + \varepsilon$, for some $\epsilon > 0$, and $\lambda(\bar{\zeta}) = L(\bar{\theta}) < 1$. □

Proof. This result is proved by a simple inductive argument. Suppose that at time t, it is true that:

$$|\bar{\phi}^1_t(\bar{\zeta}) - \bar{\phi}^2_t(\bar{\zeta})| < M(\phi^1_0, \phi^2_0)L^{t-s}(\bar{\zeta}) \tag{5.4.30}$$

Then at time $t+1$:

$$\begin{aligned}
|\bar{\phi}^1_{t+1}(\bar{\zeta}) - \bar{\phi}^2_{t+1}(\bar{\zeta})| &= |g(\bar{\phi}^1_t, \bar{\zeta}, e_{t+1}, t+1) - g(\bar{\phi}^2_t, \bar{\zeta}, e_{t+1}, t+1)| && (5.4.31)\\
&< L(\bar{\zeta})|\bar{\phi}^1_t(\bar{\zeta}) - \bar{\phi}^2_t(\bar{\zeta})| && (5.4.32)\\
&< L(\bar{\zeta})M(\phi^1_0, \phi^2_0)L^{t-s}(\bar{\zeta}) && (5.4.33)\\
&= M(\phi^1_0, \phi^2_0)L^{t+1-s}(\bar{\zeta}) && (5.4.34)
\end{aligned}$$

Thus if (5.4.30) holds for time t, then it will hold at time $t+1$. Considering $t = s$, and defining $M(\cdot, \cdot)$ as in the lemma gives an initial time at which (5.4.30) holds and completes the proof of the lemma. ∎

Remark 5.5 When $L(\bar\zeta) < 1$, $\bar\zeta$ satisfies the condition in (5.2.17) and so is an element of D_S. Note that for the particular form of (5.2.8), (5.2.10) being considered (5.4.11), (5.4.14), it has already been necessary to restrict $\bar\zeta$ such that $L(\bar\zeta) < 1$, in order to satisfy (5.2.16).

Remark 5.6 Although it is not possible to obtain a general expression giving the Lipschitz constants, in this case when $\bar\zeta$ is restricted to force L_1 of g_1 to be less than 1, the form of g_2 guarantees exponential stability of $\bar W_t$. Thus when the induced norm of F_t for a given $\bar\zeta$ is less than 1, $\bar\zeta \in D_S$.

The set D_S is thus defined.

$$D_S \;=\; \{\bar\zeta \;:\; L_1(\bar\zeta) < 1\} \tag{5.4.35}$$

It is now possible to state the differential equation (5.2.11) which governs the behaviour of the system given by equations (5.2.12)-(5.4.9). Theorems 5.1-5.4 may then be applied to give the convergence points of the algorithm.

Combining (5.2.18) and (5.4.15) gives:

$$f_D(\bar\zeta) = \lim_{t\to\infty} E\left\{\left[\begin{array}{c} \bar R_{t-1}^{-1}\bar\psi_t(\bar\zeta)\bar\omega_{t/t-1}(\bar\zeta) \\ col(\bar\psi_t(\bar\zeta)\bar\psi_t^T(\bar\zeta) + \delta I - \bar R_{t-1}) \end{array}\right]\right\} \tag{5.4.36}$$

where the expectation is taken over the signal $e(t)$.

In the limit as $t \to \infty$, the algorithm will converge to the fixed points of (5.4.36), in the sense detailed by the theorems of Section 5.2.3. It is straightforward to see that:

$$\bar R = \delta I + E\left\{\bar\psi(\bar\zeta)\bar\psi^T(\bar\zeta)\right\} \tag{5.4.37}$$

is the fixed point for the second component of the equation. Where it is assumed that $\bar\phi_t(\bar\zeta)$ has reached a stationary point $\bar\phi(\bar\zeta)$, leading to the expression $\bar\psi(\bar\zeta)$. As $\bar R > 0$, $\bar\theta$ will converge only if:

$$E\left\{\bar\psi_t(\bar\zeta)\bar\omega_{t/t-1}(\bar\zeta)\right\} \;=\; 0 \tag{5.4.38}$$

That is, the expected value of the prediction error is perpendicular to the sensitivity function. So that the algorithm is acting as a whitening filter. Thus it is concluded that the parameter has converged to an estimate that gives an accurate model for the system.

The preceding discussion is summarized in the following theorem.

Theorem 5.5

Consider the Recursive Prediction Error scheme given by (5.4.1)-(5.4.9), where the functions $A(\theta,\cdot)$, $B(\theta,\cdot)$, $C(\theta,\cdot)$, $K(\theta,\cdot)$, are Artificial Neural Networks parameterized by the weights vector θ. Then this system is given the form of (5.2.8), (5.2.10) by the identifications of (5.4.11), (5.4.14). Assume that the system of (5.2.3), (5.2.4), and the signals u_t and w_t are such that $e_t = (u_{t-1}^T, y_t^T)^T$ is a sequence of independent random variables, and y_t is bounded above by some fixed constant. For a compact domain of ϕ, D_ϕ, denote $L_2 = \max_{\phi\in D_\phi} L_2(\phi)$. Then if ζ is restricted to lie in the domain D_S, and ϕ is restricted to D_ϕ, the assumptions (5.2.12)-(5.2.18) will hold, and if (5.2.19)-(5.2.23) hold, Theorems 5.1-5.4 may be applied giving convergence of the algorithm to a fixed point of (5.4.36). □

5.5 An Example

In this section we consider a specific example of the general scheme presented here in order to demonstrate some of the properties of this scheme. Specifically, a single-input, single-output system with one dimensional state is considered. A system of the form of (5.2.3), (5.2.4) where the A, B, K and C are ANNs, is used as the system model. These networks have a single hidden layer of 5 nodes. The problem now becomes that of correctly identifying the weights of the system model. There are a total of 64 weights to be estimated. For convenience, the parameter vector θ is subdivided into 4 sub-vectors, θ_A, θ_B, θ_C and θ_K, each vector corresponding to the ANN it represents.

Following Theorem 5.5, if the existence of the constants $L_1 < 1$ and L_2 may be proven, the asymptotic behaviour of the algorithm will follow the behaviour of the associated differential equation, giving convergence of $\hat{\theta}$ to a fixed point of (5.4.36). Note that L_2 may be regarded as the maximum value of a smooth function. As we are considering the algorithm over a compact domain, the existence of this constant is guaranteed. Hence we will not attempt to find an expression for it. The constant L_1 must be bounded above, so it is necessary to derive conditions on θ in order to find the exponential stability domain of (5.2.17). An expression for the upper bound of L_1 is now derived.

The following lemmas regarding the calculation of Lipschitz constants are useful.

Lemma 5.3

(i). If $g(x)$ has the form $[g_1(x)^T,\ g_2(x)^T,\ g_3(x)^T]^T$, then its Lipschitz constant is bounded by:

$$L_x(g) \leq \left[L_x(g_1)^2 + L_x(g_2)^2 + L_x(g_3)^2\right]^{1/2} \tag{5.5.1}$$

(ii). Consider the function $f(x_1, x_2, x_3)$ which has Lipschitz constants with respect to each parameter, $L_1(f)$, $L_1(f)$, $L_1(f)$, then the Lipschitz constant with respect to the vector (x_1, x_2, x_3) is bounded as follows:

$$L(f) \leq \left[L_1(f)^2 + L_2(f)^2 + L_3(f)^2\right]^{1/2} \tag{5.5.2}$$

(iii). The following bounds may be obtained for the product and composition of two functions with known Lipschitz constants.

$$\begin{aligned} L_1(f(x)g(x)) &\leq L_1(f)\max_x|g(x)| + L_1(g)\max_x|f(x)| && (5.5.3)\\ L_1(f(g(x))) &\leq L_1(f)L_1(g) && (5.5.4)\end{aligned}$$

□

Lemma 5.4 *Consider a 3 layer feedforward ANN, as described by (5.3.10), with one input x, and one output y and parameter vector θ.*

$$y = N(\theta, x)$$

Denote the derivatives of $N(\theta, x)$ with respect to θ and x by $N_1(\theta, x)$ and $N_1(\theta, x)$ respectively. Then the Lipschitz constants of these functions will be bounded by the following expressions.

$$\begin{aligned} L_x(N) &\leq \frac{1}{16}\|\theta^1\|\ \|\theta^2\| && (5.5.5)\\ L_\theta(N) &\leq \frac{1}{2}\left[\frac{1}{16}\left(\|x\|^2+1\right)\|\theta^2\|^2 + (n+1)\right]^{\frac{1}{2}} && (5.5.6)\\ L_x(N_1) &\leq \frac{1}{4}\|\theta^1\|^2\|\theta^2\|\left(\frac{1}{4}\|\theta^2\|+1\right) && (5.5.7)\\ L_x(N_2) &\leq 4\ L_x(N)L_\theta(N) + \frac{1}{4}\|\theta^2\| + \left[\left(\frac{1}{10}\|\theta^2\|\ \|\theta^1\|\right)^2 + \frac{1}{16}\|\theta^1\|^2\right]^{\frac{1}{2}} && (5.5.8)\end{aligned}$$

□

Remark 5.7 The proofs of these lemmas are based on the idea that the Lipschitz constant may be determined by deriving the maximum value of the derivative of the function. As the proofs are straightforward and tedious, they are not presented here.

These lemmas may now be applied to give an expression for an upper bound on L_1 as follows.

Note that $g(\zeta, \phi, e)$ is composed of the components g_1, g_2, g_3, so that (5.5.1) so that:

$$L_1 \;=\; \left(L_1(g_1)^2 + L_1(g_2)^2 + L_1(g_3)^2\right)^{\frac{1}{2}} \tag{5.5.9}$$

The Lipschitz constants of the g_i may be determined through the use of (5.5.2):

$$L_1(g_i) \;=\; \left(L_x(g_i)^2 + L_W(g_i)^2 + L_y(g_i)^2\right)^{\frac{1}{2}}, \qquad i = 1, 2, 3 \tag{5.5.10}$$

Note that $g_3(\phi_{t-1}, \theta_{t-1}, e_t) = y_t$, so $L_1(g_3) = 0$.

The expression for g_1 is relatively straightforward, so that the Lipschitz constants may be stated as follows:

$$L_x(g_1) \;\le\; \frac{1}{16}\|\theta_A^1\|\;\|\theta_A^2\| + \frac{1}{256}\|\theta_K^1\|\;\|\theta_K^2\|\;\|\theta_C^1\|\|\theta_C^2\| \tag{5.5.11}$$

$$L_W(g_1) \;=\; 0 \tag{5.5.12}$$

$$L_y(g_1) \;\le\; \frac{1}{16}\|\theta_K^1\|\;\|\theta_K^2\| \tag{5.5.13}$$

The expressions for the components of $L_1(g_2)$ are not as tractable, as F_t and G_t are the first derivatives of g_1 with respect to x and θ respectively.

$$L_x(g_2) \;\le\; L_x(F) \max_{\phi \in D_\phi} \|W\| + L_x(G) \tag{5.5.14}$$

$$L_W(g_2) \;\le\; \max_{\phi \in D_\phi} \|F\| \quad = L_1(g_1) \tag{5.5.15}$$

$$L_y(g_2) \;\le\; L_y(F) \max_{\phi \in D_\phi} \|W\| + L_y(G) \tag{5.5.16}$$

From (5.4.8) and (5.4.9), and applying Lemma 5.3 and Lemma 5.4, the following expressions may be derived.

$$L_x(F) \;\le\; L_2(A_2) + L_2(K_2)L_2(C)^2 + L_2(K)L_2(C_2) \tag{5.5.17}$$

$$L_x(G) \;\le\; L_2(A_1) + L_2(K_1)L_2(C) + L_2(K_2)L_2(C)L_1(C) + L_2(K)L_2(C_1) \tag{5.5.18}$$

$$L_y(F) \leq L_2(K_2) \tag{5.5.19}$$

$$L_y(G) \leq L_2(K_1) + L_1(C)L_2(K_2) \tag{5.5.20}$$

Substituting equations (5.5.10)-(5.5.20) into (5.5.9) gives an upper bound on L_1. By ensuring that this is less than 1, the conditions of the theorem will be satisfied.

By choosing a θ from within the region thus specified to represent the system which is to be identified, and then choosing an initial estimate from within the same region, the discussion of the previous section indicates that convergence will occur when (5.4.38) holds.

The algorithm (5.4.1)-(5.4.9) was encoded using the XMATH simulation package. A number of estimation problems were then simulated. The system parameter vector θ, and the initial parameter estimate $\hat{\theta}_1$ were chosen randomly so as to be guaranteed to satisfy the requirement $L_1 < 1$. Three choices for γ_t were used, $1/10$, $1/t$, $t^{-1/5}$. The sequences u_t and w_t were chosen to be sequences of random numbers in the range $[0, 1]$, generated by XMATH.

In all cases the parameters converged to some limiting value $\hat{\theta}^\star$, however at widely varying rates. This seemed to be dependent on a combination of the choice of γ_t, u_t and w_t as well as θ and $\hat{\theta}_1$. An important point to note is that the limiting value $\hat{\theta}^\star$ seemed to bear no discernible relationship to the actual system parameter θ. However, in all cases as the parameter converged the difference between the noiseless system output and the estimated system output approached 0, or some small positive constant. This appeared to be dependent on the choice of the sequence γ_t. The slower that $\gamma_t \to 0$, the better the approximation to the system output.

The parameter vector and initial conditions were chosen randomly. In presenting this simulation results, we only intend giving an idea of how the algorithm behaves, in terms of the convergence of the parameter estimate to the models parameter value, and in terms of how well the identified system models the model. From this point of view the exact parameters are not important, and so are not presented here.

The evolution of $\hat{\theta}_t$ is shown in Figure 5.5, and the evolution of the difference between the system model's output and the estimated system's output is shown in Figure 5.5. Note that this is $y_t - w_t - \hat{y}_t$, and not $y_t - \hat{y}_t$. The noise input w_t was sufficiently large that the signal $y_t - \hat{y}_t$ was meaningless.

As may be seen from the figures, the ouput of the estimated plant was approaching that of the model. It is interesting to note, however that $\hat{\theta}_t$ was not approaching θ. This

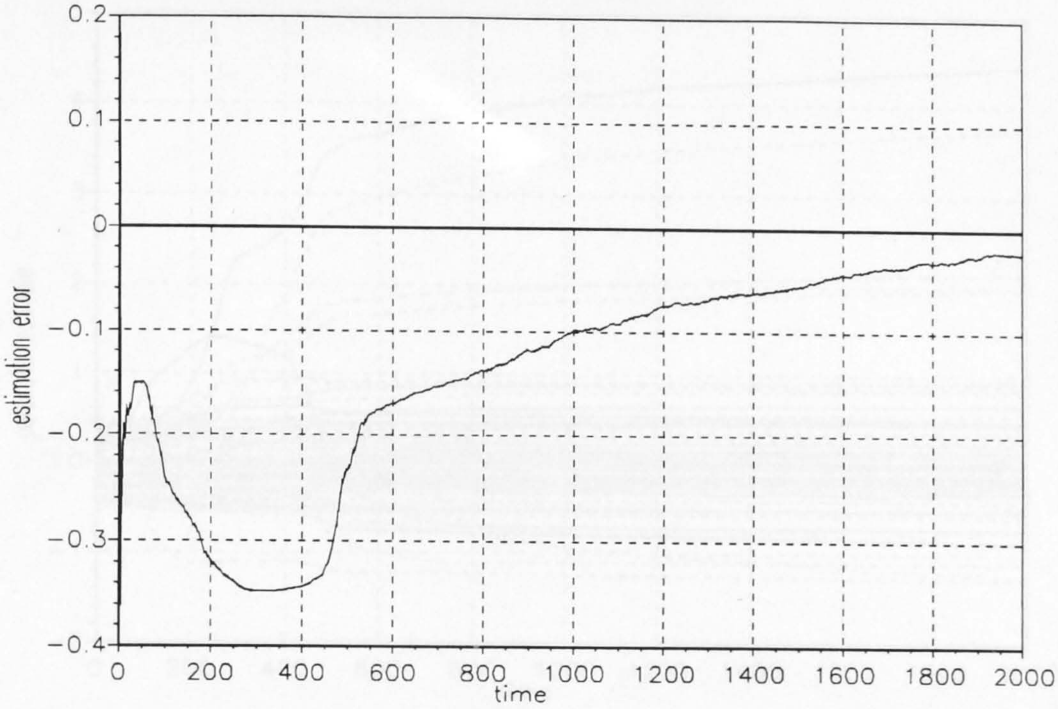

Figure 5-2: Evolution of the estimation error, $y_t - \hat{y}_t - w$.

indicates that there are many potential convergence points for the algorithm.

It is recognised that the simulation presented offers a very limited viewpoint of the properties of the algorithm, however it does demonstrate the promise of this approach to nonlinear system identification.

5.6 Conclusion

In this chapter we have presented an algorithm for the identification of an unknown stable nonlinear system, as given by (5.4.1)-(5.4.9), which is based on an RPE approach. The nonlinearities in the system are modelled by feedforward ANNs with smooth sigmoidal nodes, as described in Section 5.3.1. Convergence of the algorithm is guaranteed when the algorithm satisfies the conditions of Section 5.4.2. Furthermore an ordinary differential equation is presented which has as it's stationary points exactly those points to which the parameter estimate will converge.

It is shown in Theorem 5.5 that under some restrictions on the range of parameters allowed, the algorithm presented will satisfy these assumptions, and so it is possible to predict the behaviour of the system in terms of the solutions to the associated ordinary

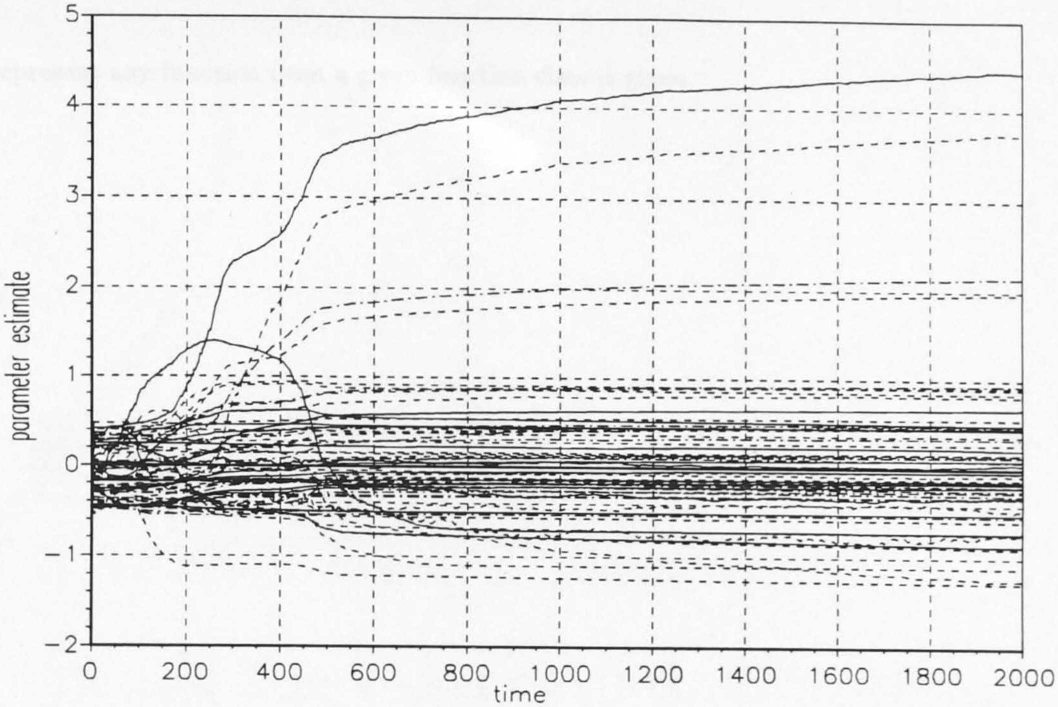

Figure 5-3: Evolution of the parameter estimate, $\hat{\theta}_t$.

differential equation. As was stated in Section 5.5, the parameter $\hat{\theta}$ will converge, such that the system behaviour is modelled, however no relationship between the actual system parameter θ and $\hat{\theta}$ is guaranteed.

Although much further work is required, the RPE-ANN algorithm presented is interesting. A simple example confirms the appropriateness of this approach, and indicates the potential of this approach to nonlinear system identification.

Artificial Neural Networks have been used for the function estimators as it has been shown that they can act as universal approximators. For a given approximation error an adequate number of nodes will be required, however this has not been adequately determined in the past. As the parameter vector, θ, is the vector of all the weights in the ANNs, its dimension will be determined by the accuracy required. As this is not currently determined, it seems appropriate to use as many nodes as possible. However, note that the dimension of R_t is equal to the square of the dimension of θ, and a matrix inverse must be calculated for each iteration. Thus it is desirable for the number of nodes used to be as small as possible. At this point a result giving the some relationship between the accuracy of approximation and the number of nodes required is required. This is the subject of the following chapter. For a specific architecture the number of nodes required to represent any function from a given function class is given.

Chapter 6

The Number of Nodes Required in a Feed-Forward Neural Network for Functional Representation

6.1 Introduction

Feedforward Artificial Neural Networks (ANNs) have been shown to be "universal approximators" in the sense that a wide class of functions can be approximately represented in terms of a neural network. These results say nothing of the number of nodes needed to attain a given level of approximation. As indicated in the previous chapter, in some circumstances it is useful to be able to specify the number of nodes required to represent a function to a given level of accuracy. Specifically, for a particular architecture presented, a bound on the number of nodes required to represent any function from a given function class is calculated. Further, the number of bits required to represent the neural network is compared to the ε-entropy of the class of functions being represented as a means of determining the efficiency of the representation.

Consider the use of feedforward neural networks as functional representations, approximating some function $f\colon \mathbb{R}^n \to R$, where R is typically a closed and compact interval. While the problem of determining whether feedforward neural nets are in some sense "universal approximators" can be considered solved, the solutions to date are still unsatisfactory. Most of the currently available results either appeal to Kolmogorov's famous theorem, such as Funahashi [11], Hecht-Nielsen [20], Kolmogorov [25], or are non-constructive existence statements, Cybenko [4], Funahashi [11], Hornik, Stinchcombe and White [22]. The only exceptions we know of are Carroll and Dickinson [2] and Pati and Krishnaprasad [43]. The problem of determining the rate at which the number of hidden units needed to attain a given accuracy of approximation must grow as the dimension of the input space increases is yet to be solved.

In this chapter we will consider the difficulty of representing an arbitrary function from a given function class which is specified *a priori*. The representability problem is studied using Kolmogorov's ε-entropy. A new neural network structure for uniform functional approximation is presented, and it is shown that it is close to optimal in terms of the bit efficiency of storage. An explicit upper bound on the number of nodes needed in this neural network architecture is also obtained.

This new scheme represents a function via convex polytopal approximations to the level sets of the function. All but one of the neurons in the network have threshold logic output functions. Note that in the previous chapter we used smooth activation functions, and not threshold functions as used in this chapter. We have chosen to work with step functions as the analysis of an ANN with smooth activation functions is considerably more difficult. However, note that a step function may be regarded as the limiting case of the sigmoid of the previous chapter as the gradient at the origin approaches infinity. Hence it is expected that the results of this chapter also provide a bound on the number of smooth nodes required to represent any function from the given function class.

This chapter is organised as follows. Section 6.2 introduces the idea of ε-entropy, and some useful results are presented. In Section 6.3 the method of representing a function in terms of its level sets is described. The new neural network architecture is presented in Section 6.4. This architecture is based on the level set representation of a function from Section 6.3. The number of nodes required for this network to achieve an ε-approximation is calculated, giving the main result of the chapter, Theorem 6.8, Section 6.4. These results are then compared with the ε-entropy for the function class for which this architecture gives an ε-approximation, giving an idea of the efficiency of our representation. Conclusions are presented in Section 6.5.

6.2 ε-Entropy of Functional Classes

The concept of ε-entropy was introduced by Kolmogorov [23, 24], Vitushkin [58, 59] and Tikhomirov [52]. The ε-entropy is sometimes called the metric entropy [9, 10, 35]

6.2.1 Basic Concepts and Results of ε-Entropy

Let F be a non-empty set in a metric space Φ. By a metric space we shall mean a function space which has a metric $d(\cdot,\cdot)$ associated with it. Note that all logarithms are to base 2.

Definition 6.1 [ε-covering] *A system γ of subsets of Φ is called an ε-covering of F if the diameter of any set $U \in \gamma$ is less than 2ε and if $F \subset \bigcup_{U\in\gamma} U$.*

Definition 6.2 [ε-net] *A set $U \subset \Phi$ is called an ε-net of F if for all $x \in F$ there exists $y \in U$ such that $d(x, y) < \varepsilon$.*

Definition 6.3 [ε-separation] *A set $U \subset \Phi$ is called ε-separated if every pair of distinct points in U are at a distance greater than ε apart.*

Note that the set F is more important to our discussion than the space in which it is embedded, Φ, which is not unique. For example, if F is a collection of sequences in l^2, Φ could be taken to be l^p for any $p = 2, 3, \dots, \infty$.

We will work with totally bounded sets as they have three equivalent useful properties, as stated in the following theorem.

Theorem 6.1 [52]

The following three properties of the set F are equivalent and depend on the metric of F. That is, they hold independently of the space Φ within which F is embedded.

(i). For all ε, there exists a finite ε-covering of F.

(ii). For all ε, there exists a finite ε-net of F.

(iii). For all ε, every ε-separated set is finite.

□

For a totally bounded set F it is natural to introduce three functions which are measures of the size of the set F.

Definition 6.4 [minimal ε-covering] *$\mathcal{N}_\varepsilon(F)$ is the minimal number of sets in an ε-covering of F.*

Definition 6.5 [minimal ε-net] *$\mathcal{N}_\varepsilon^\Phi(F)$ is the minimal number of sets in an ε-net of F.*

Definition 6.6 [maximal ε-separated set] *$\mathcal{M}_\varepsilon(F)$ is the maximal number of points in an ε-separated subset of F.*

Note that for a given metric, $\mathcal{N}_\varepsilon(F)$ and $\mathcal{M}_\varepsilon(F)$ are independent of the set Φ in which F is embedded, while $\mathcal{N}^\Phi_\varepsilon(F)$ in general depends on Φ.

Definition 6.7 [absolute ε-entropy] *The absolute ε-entropy for a set F is given by $\mathcal{H}_\varepsilon(F) = log\mathcal{N}_\varepsilon(F)$.*

Definition 6.8 [relative ε-entropy] *The relative ε-entropy for a set F with respect to Φ is given by $\mathcal{H}^\Phi_\varepsilon(F) = log\mathcal{N}^\Phi_\varepsilon(F)$.*

Definition 6.9 [ε-capacity] *The ε-capacity of a set F is given by $\mathcal{C}_\varepsilon(F) = log\mathcal{M}_\varepsilon(F)$.*

Theorem 6.2 [59]

The absolute ε-entropy of a compact metric space F is equal to the lower bound on the relative ε-entropies of F for all possible metric expansions Φ of F:

$$\mathcal{H}_\varepsilon(F) = \inf_{\Phi\supset F} \mathcal{H}^\Phi_\varepsilon(F). \tag{6.2.1}$$

□

Theorem 6.3 [59]

For any compact metric space F, any metric expansion Φ of F and any $\varepsilon > 0$,

$$\mathcal{H}_{2\varepsilon}(F) \leq \mathcal{H}^\Phi_{2\varepsilon}(F) \leq \mathcal{C}_{2\varepsilon}(F) \leq \mathcal{H}_\varepsilon(F) \leq \mathcal{H}^\Phi_\varepsilon(F) \tag{6.2.2}$$

□

Vitushkin [59] introduces the idea of a table $T^\Phi_{F,\varepsilon}$ of functions of a metric space F. This may be thought of as an encoding of an ε-net, or ε-representation of F. In this way the idea of representing an arbitrary function from the class F is introduced. Further, the relative ε-entropy with respect to this ε-net gives an idea of the efficiency of the representation. The most efficient ε-representation has an entropy equal to the absolute ε-entropy for the class of functions F.

Let $C = C(I)$ be the metric space of continuous functions on $I = [0,1]^n$ with the sup (or uniform) metric d_∞:

$$d_\infty(f,g) = \sup_{x\in I} |f(x) - g(x)| \qquad \forall f,g \in C(I) \tag{6.2.3}$$

Theorem 6.4 [59]

C is the best metric expansion for a compact metric space F in the sense that

$$\mathcal{H}_\varepsilon^C(F) = \mathcal{H}_\varepsilon(F) \leq \mathcal{H}_\varepsilon^\Phi(F) \tag{6.2.4}$$

for any metric expansion Φ of F. □

Remark 6.1 This theorem states that if the L_∞ norm is used for calculating the ε-entropy of a function class F, it will be less than if a different norm, for instance the L_2 norm, where used. *i.e.* $\mathcal{H}_\varepsilon^{L_\infty}(F) \leq \mathcal{H}_\varepsilon^{L_2}(F)$. Nevertheless there may be circumstances in which we wish to use a different metric (thus meaning that the approximations will not be uniform: for example using the Euclidean distance in the definitions above, we would end up with mean-square approximations). Many of the results on ε-entropy in terms of the uniform metric carry over to other metrics. Details can be found in the papers by Lorentz [34, 35].

6.2.2 ε-Entropy for some Function Classes

The ε-entropy is now calculated for some specific function classes. Proofs may be found in [26].

Definition 6.10 [Lipschitz Function] *A function f which is a member of some metric space F with metric d_F satisfies* a Lipschitz condition with constant L and index α *if for all $x, y \in \operatorname{dom} f$,*

$$|f(x) - f(y)| \leq L\,[d_F(x,y)]^\alpha. \tag{6.2.5}$$

Theorem 6.5

Let F_L^ρ be the space of all functions $f(x)$ defined on the interval $\rho = [a,b]$, satisfying a Lipschitz condition with constant L under the metric d_∞ and satisfying $f(a) = 0$. Then

$$\begin{aligned} \mathcal{H}_\varepsilon(F_L^\rho) &= \frac{|\rho| L}{\varepsilon} + O(1) && (6.2.6) \\ \mathcal{C}_\varepsilon(F_L^\rho) &= \frac{2|\rho| L}{\varepsilon} + O(1), && (6.2.7) \end{aligned}$$

where $|\rho| = b - a$. □

Definition 6.11 [The Class $F^{\rho,n}_{s,L,C}$] *For a given non-negative integer p and $\alpha \in (0,1]$, set $s = p + \alpha$. Then $F^{\rho,n}_{s,L,C}$ denotes the space of real functions f defined on $[0,\rho]^n$ all of whose partial derivatives of order p satisfy a Lipschitz condition with constant L and index α (6.2.5), and are such that*

$$\left| \frac{\partial^{k_1+k_2+\cdots+k_n} f(0)}{\partial x_1^{k_1} \partial x_2^{k_2} \cdots \partial x_n^{k_n}} \right| \le C \quad \text{for} \quad \sum_{i=1}^{n} k_i \le p. \tag{6.2.8}$$

Theorem 6.6 [Kolmogorov, Vitushkin [59, p.86]]
For sufficiently small ε

$$A(s,n)\rho^n \left(\frac{L}{\varepsilon}\right)^{\frac{n}{s}} \le \mathcal{H}_\varepsilon \left(F^{\rho,n}_{s,L,C}\right) \le B(s,n)\rho^n \left(\frac{L}{\varepsilon}\right)^{\frac{n}{s}}, \tag{6.2.9}$$

where $A(s,n)$ and $B(s,n)$ are positive constants depending only on s and n □

This theorem is proved in [59, pp.86ff] and [26, pp.308ff]. The constants $A(s,n)$ and $B(s,n)$ can not be determined exactly.

Theorem 6.6 says that one can trade off dimensionality n against smoothness s, thus indicating a way of avoiding complexities exponential in the dimension of the input space.

The class of Lipschitz continuous functions from $\mathbb{R}^n$ to $\mathbb{R}$, ie $F^{\rho,n}_{1,L,C}$, shall be referred to as $F^{\rho,n}_{L,C}$.

Corollary 6.1 *For sufficiently small ε*

$$A(n)\left(\frac{\rho L}{\varepsilon}\right)^n \le \mathcal{H}_\varepsilon \left(F^{\rho,n}_{L,C}\right) \le B(n)\left(\frac{\rho L}{\varepsilon}\right)^n. \tag{6.2.10}$$

□

6.3 Representation of Functions in Terms of Their Level Sets

The main idea in this chapter is a scheme whereby a function of n variables with a domain which is a compact subset of $\mathbb{R}$ can be represented in terms of its level sets. It is first shown how this representation works mathematically, and then a simple discretization of it is described. Our motivation for this representation is from Arnol'd [1].

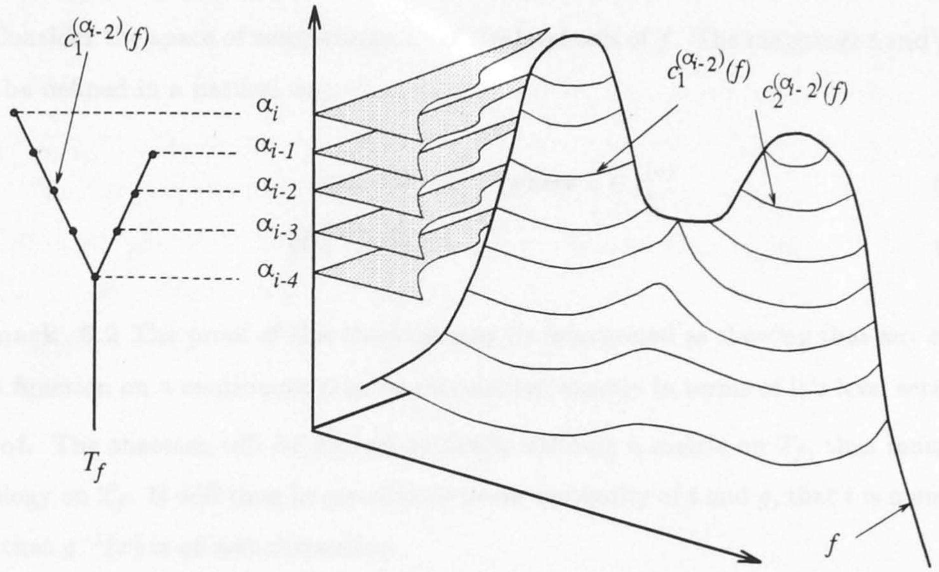

Figure 6-1: Illustration of T_f and $l_\alpha(f)$ for f defined on $\mathbb{R}^2$. For this function all levels below α_{i-4} have only one component. Also note that the levels are shown equally spaced and finite in number for the purpose of illustration.

6.3.1 Level Sets and the Space of Components

Definition 6.12 [α-level set] *The* α-level set *of a function* $f\colon D \to R$ *is defined by*

$$l_\alpha(f) \overset{\triangle}{=} \{x\colon f(x) = \alpha\}. \tag{6.3.1}$$

If $\dim D = n$, $\dim l_\alpha(f) \leq n-1$. Each set $l_\alpha(f)$ of a different level consists of *components,* [27] (see [1, p.128]) continua that do not intersect each other. That is $l_\alpha(f) = \bigcup_j c_j^{(\alpha)}$ and $c_j^{(\alpha)} \cap c_k^{(\alpha)} = \emptyset$ for $j \neq k$. Denote by T_f the space of components of all the level sets of f. These concepts are illustrated for a function in two dimensions in figure 6-1.

Theorem 6.7 [1, p. 129]
The real continuous mapping $f\colon A \to \mathbb{R}$, *such that* A *is a continuum, is the product of two continuous mappings, a monotone mapping* $t\colon A \to T_f$, *where* T_f *is the space of components of* f, *and* $g\colon T_f \to \mathbb{R}$. *under which the counter image of every point is of zero dimension.* □

A topological space A is said to be a continuum if it is compact, complete and path-connected. A continuous mapping is said to be monotone if the counterimage of every point is connected. Before continuing, note that since f is continuous and A is compact, $\operatorname{ran} f$ is compact. Furthermore if A is connected then $\operatorname{ran} f$ is connected (i.e. a closed interval in $\mathbb{R}$).

Consider the space of components T_f of the level sets of f. The mappings t and g may now be defined in a natural way:

$$t(x) \overset{\triangle}{=} c_i^{(\alpha)} \quad \text{where } x \in c_i^{(\alpha)} \tag{6.3.2}$$

$$g(c_i^{(\alpha)}) \overset{\triangle}{=} \alpha \tag{6.3.3}$$

Remark 6.2 The proof of this theorem may be interpreted as showing that any continuous function on a continuum may be represented exactly in terms of it's level sets.

Proof. The theorem will be proved by firstly defining a metric on T_f, thus inducing a topology on T_f. It will then be possible to prove continuity of t and g, that t is monotone, and that $g^{-1}(x)$ is of zero dimension.

Note that c, an element of T_f will be considered as a point in T_f and a subset of A with no change in notation. The particular sense of c being referred to will be clear from the context. Also, note that whereas $c_i^{(\alpha)}$ denotes the i^{th} component of l_α, c_i (without the (α) superscript) denotes the i^{th} (in some other indexing) component out of all the possible components of all the level sets. Two given components c_i and c_j need not be of the same level set.

Let $c_1, c_2 \in T_f$ and define the operator $d(\cdot,\cdot)$ on T_f by

$$d(c_1,c_2) \overset{\triangle}{=} \inf_{\substack{c_1 \cup c_2 \subseteq F \subseteq A \\ F \text{ connected}}} \left[\max_{x\in F} f(x) - \min_{x \in F} f(x)\right], \tag{6.3.4}$$

where the term in brackets is known as the *oscillation* of f on F. Note that the infinum in (6.3.4) is taken over F, a connected superset of $c_1 \cup c_2$. It is obvious that

$$0 \leq d(c_1,c_2) \leq d(c_1,c_3) + d(c_3,c_2) \tag{6.3.5}$$

and that $d(c_1,c_1) = 0$. In order to prove that $d(\cdot,\cdot)$ is in fact a metric we need to show $d(c_1,c_2) = 0 \Rightarrow c_1 = c_2$.

If $g(c_1) \neq g(c_2)$ it is obvious that $d(c_1,c_2) \neq 0$, so the only case we need to deal with is when c_1 and c_2 are disjoint components of the same level set, $c^{(\alpha)}$. As c_1 and c_2 are disjoint, there must exist points $x \in F$ such that $f(x) \neq \alpha$, hence the oscillation over F must be non-zero. Thus the only case in which $d(c_1,c_2) = 0$ is when c_1 and c_2 are connected components of the same level set. By the definition of a component of a level set this means that $c_1 = c_2$. Thus $d(\cdot,\cdot)$ is a metric on T_f.

Since f is continuous, for all $\varepsilon > 0$ there exists a $\delta > 0$ such that, if $y \in B(x,\delta)$, then $f(y) \in B(f(x), \varepsilon/2)$. This implies that f has oscillation less than ε on $B(x,\delta)$. Now consider c_1 and c_2 passing through $B(x,\delta)$. Then $d(c_1,c_2) < \varepsilon$. Let c_1 be $t(x)$ and c_2 be $t(y)$. Then for all $y \in B(x,\delta)$, $t(y) \in B(f(x),\varepsilon)$, which establishes the continuity of t.

Consider $c_1, c_2 \in T_f$, such that $d(c_1,c_2) < \varepsilon$. Then for $x_1 \in c_1$ and $x_2 \in c_2$, $|f(x_1) - f(x_2)| < \varepsilon$ as otherwise the oscillation on $F \supseteq c_1 \cup c_2$ would be greater than ε. This establishes continuity of g, as $g(c_1) = f(x_1)$ and $g(c_2) = f(x_2)$.

We now consider the counterimage of f, thus showing monotonicity of t. The counterimage of any element of T_f under the mapping t is c, a component of a level set of f. By the definition of a component of a level set of f, c will be connected and hence t is monotone. This establishes the theorem. ∎

6.3.2 ε-Approximations Based on Theorem 6.7

Consider now ε-approximations based on the representation of theorem 6.7 Since dom f is assumed to be a continuum, ran f will be a closed interval which can be mapped isometrically to $[0,1]$. Thus the following development is quite general.

Definition 6.13 [α-above set] *The* α-above set *of a function f is defined by*

$$\begin{aligned} \bar{l}_\alpha(f) &\triangleq \{x: f(x) \geq \alpha\} && (6.3.6) \\ &= \bigcup_{\beta \geq \alpha} l_\beta(f). && (6.3.7) \end{aligned}$$

Obviously the above-sets will also consist of components $\bar{c}_j^\alpha(f)$, i.e. $\bar{l}_\alpha(f) = \bigcup_j \bar{c}_j^\alpha(f)$.

Definition 6.14 [N-uas representation] *Let $\alpha^{(N)} \triangleq \{\alpha_1, \ldots, \alpha_N\}$ where $\alpha_i = \frac{i-1}{N}$ be a set of levels over $[0,1]$ (a* uniform quantization *over $[0,1]$) and let D be a compact interval in $\mathbb{R}^n$. Then the* N-uniform above set representation *of $f: D \to [0,1]$ is given by*

$$\tilde{f}_N(x) \triangleq \frac{1}{2N} + \frac{1}{N}\sum_{i=1}^{N} \mathbf{1}_{\bar{l}_{\alpha_i}(f)}(x). \qquad (6.3.8)$$

where $\mathbf{1}_S$ is the indicator function *of a set S.*

The $\frac{1}{2N}$ term arises in (6.3.8) because the error is in terms of absolute values: the approximation is allowed to be greater than or less than the function f. The inter-level spacing, or step-size, $\frac{1}{N}$ shall be referred to as s_α.

Lemma 6.1 *$\tilde{f}_N(x)$ of (6.3.8) is an ε-approximation of $f\colon D \to [0,1]$ in the sup-metric if $s_\alpha \le \varepsilon$.* □

Proof. By Theorem 6.7 and the construction of $\tilde{f}_N(x)$, it is apparent that $\left|\tilde{f}_N(x) - f(x)\right| \le s_\alpha$ for all $x \in D$. Setting $s_\alpha \le \varepsilon$ proves the lemma. ∎

6.4 A Neural Network ε-Approximation Based on Lemma 6.1

In this section a new neural network architecture is proposed, based on representing a function by N equally spaced level sets. Such an ANN will be denoted a $\mathcal{L}_{2\varepsilon_l}$-$N$-ANN. The main result of this chapter is:

Theorem 6.8

The number of nodes needed in a $\mathcal{L}_{2\varepsilon_l}$-$N$-ANN in order to represent any $f \in F^{\rho,n}_{L,C}$ to within ε_r in the sup-metric is bounded above by

$$\nu^A \approx \frac{n\rho L}{2\varepsilon_r} + \frac{1}{\sqrt{2}\varepsilon_r} + \left(1 + \frac{n}{\sqrt{2}}\right)\left(\frac{\rho L}{4\varepsilon_r}\right)^n \qquad (6.4.1)$$

$$= O\left(\frac{n}{\sqrt{2}}\left(\frac{\rho L}{4\varepsilon_r}\right)^n\right). \qquad (6.4.2)$$

□

6.4.1 The Architecture of the Networks we will Consider

We now develop an ANN architecture based on N uniform above set representations which can be used to approximately represent functions $f \in F^{\rho,n}_{L,C}$. The type of architecture we adopt is shown in figure 6-2. The idea is that the neural net NN_i approximately represents $\bar{l}_{\alpha_i}(f)$ by $\tilde{l}_{\alpha_i}(f)$ where α_i is as in definition 6.3.2. The function f is then approximated by (6.3.8). Each subnet NN_i is a two layer net with Heavyside step functions as activation nodes. The first layer implements a number of hyperplane decision boundaries; the second layer forms convex polytopes by and-ing the outputs of the first layer; and the third layer forms general regions by or-ing the outputs of the second layer.

The output of the Neural Network may be precisely stated as

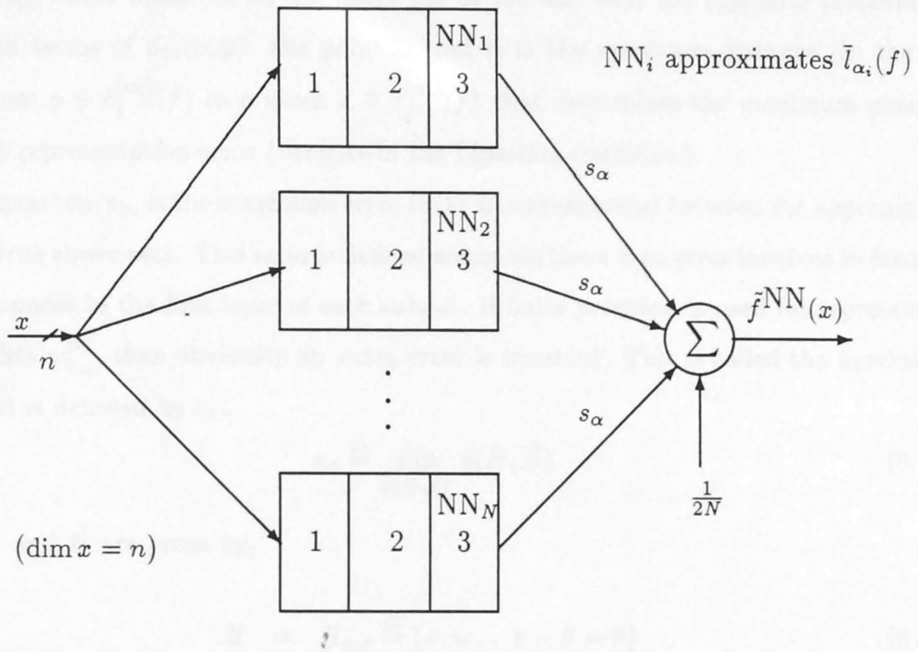

Figure 6-2: The Neural Network architecture we adopt.

$$\tilde{f}^{\ \mathrm{NN}}(x) = \frac{1}{2N} + \underbrace{\sum_{i=1}^{N} s_\alpha}_{\text{last}} \underbrace{\bigvee_{j=1}^{\nu_2^{(i)}}}_{\text{third}} \underbrace{\bigwedge_{k=1}^{\nu_1^{(i)}} v_{j,k}^{(i)}}_{\text{second}} \underbrace{\mathrm{sgn}\left(\sum_{q=1}^{n} w_{k,q}^{(i)} x_q - \theta_k^{(i)}\right)}_{\text{first}} \qquad x \in [0,\rho]^n, \qquad (6.4.3)$$

where $x = (x_1, \ldots, x_n)^T$, $w_{k,q}^{(i)} \in \mathbb{R}$ and $v_{j,k}^{(i)} \in \{1, 0, -1\}$. We denote ANNs described by (6.4.3) as N-uniform above-set ANNs (N-ANNs).

6.4.2 The Errors Incurred in the $\tilde{f}^{\mathrm{NN}}$ Approximation

Consider the errors incurred by the approximation (6.4.3). Denote by ε_{α_i} the error incurred by approximating the i^{th} above-set by the net NN_i.

$$\varepsilon_{\alpha_i} \overset{\triangle}{=} \max_{j=1,\ldots,\Lambda^{(i)}} \ \sup_{x \in \overline{c}_j^{(\alpha_i)}(f)} \ \inf_{y \in \hat{c}_j^{(\alpha_i)}(f)} \ d_\infty(x,y) \qquad (6.4.4)$$

where $\Lambda^{(i)}$ is the number of components in the above-level-set $\overline{l}_{(\alpha_i)}(f)$, $\overline{c}_j^{(\alpha_i)}(f)$ is a component of the above-level-set, and $\hat{c}_j^{(\alpha_i)}(f)$ is the approximation to this (actually the boundary of $\tilde{c}_j^{(\alpha_i)}(f)$, the approximation to the j^{th} component of the α_i-above-set). Note that the distance used to define ε_{α_i} is not a metric (it does not satisfy the triangle inequality). However it is the distance we want because symmetrizing it (as in the Hausdorff metric

[19, p.166]) would mean we do not make use of the fact that the Lipschitz condition is defined in terms of $d_\infty(x,y)$: the point is that it is the *minimum* distance (in the d_∞ sense) from $y \in \tilde{c}_j^{(\alpha_i)}(f)$ to a given $x \in c_j^{(\alpha_i)}(f)$ that determines the maximum possible allowable representation error (because of the Lipschitz condition).

The quantity ε_{α_i} is the maximum error (over all components) between the approximate and the true above-sets. This error is defined assuming there is no error involved in forming the half-spaces in the first layer of each subnet. If finite precision is used for representing the weights $w_{k,q}^{(i)}$, then obviously an extra error is incurred. This is called the *hyperplane error* and is denoted by ε_h.

$$\varepsilon_h \overset{\triangle}{=} \sup_{x \in [0,\rho]^n} d(H, \tilde{H}) \tag{6.4.5}$$

where H and $\tilde{H}$ are given by,

$$\begin{aligned} H &= H_{w,\theta} \overset{\triangle}{=} \{x : w \ldots x - \theta = 0\} && (6.4.6) \\ \tilde{H} &= H_{\tilde{w},\tilde{\theta}} = \{x : \tilde{w} \ldots x - \tilde{\theta} = 0\}, && (6.4.7) \end{aligned}$$

and the distance between the hyperplanes is

$$d\,(H, \tilde{H}) \overset{\triangle}{=} \sup_{y \in \tilde{H}} \inf_{x \in H} \parallel x - y \parallel_\infty . \tag{6.4.8}$$

Assume the same number of bits are used to represent all of the weights in the first layer. The quantity $d(H,\tilde{H})$ may be interpreted as the maximum distance between the two closest points on the hyperplanes.

Hyperplane Error Resulting From Inaccuracy of Weights

Consider the problem of approximating a given hyperplane, H, by $\tilde{H}$ on the n-dimensional cube $J = [0,\rho]^n$. Assume that $\parallel w \parallel = 1$, and that $\parallel \tilde{w} \parallel$ is close to 1. Define $\varepsilon_w \in \mathbb{R}^n$, the error in the weights, and $\varepsilon_\theta \in \mathbb{R}$ the offset error by

$$\begin{aligned} \varepsilon_w &\overset{\triangle}{=} w - \tilde{w} && (6.4.9) \\ \varepsilon_\theta &\overset{\triangle}{=} \theta - \tilde{\theta}. && (6.4.10) \end{aligned}$$

Lemma 6.2 *For an approximation $\tilde{H}$ to the hyperplane H to be within ε_h of H over all $[0,\rho]^n$, it is sufficient that the errors ε_{w_i} $(i = 1,\ldots,n)$ and ε_θ satisfy*

$$\sqrt{n}\left(|\varepsilon_\theta| + \rho\sum_{i=1}^{n}|\varepsilon_{w_i}|\right) \;\leq\; \varepsilon_h \tag{6.4.11}$$

□

Proof. Consider first the problem of finding the distance to the closest point on H from a given point $y \in \tilde{H}$. If we were to use the Euclidean norm, $\|\cdot\|_2$, it may be seen that this distance will be given by the radius of the sphere centered on y, and tangent to the hyperplane H, intersecting it at the point x_0. Furthermore the vector $x_0 - y$ will be normal to H and thus parallel to w. Note that for all x, $\| x \|_\infty \leq \sqrt{n} \| x \|_2$. Hence, taking account of this factor, we can use the Euclidean norm to bound $d(H,\tilde{H})$. We have for any $y \in \tilde{H}$,

$$\inf_{x\in H} \| x - y \|_2 \;=\; |\, t \,| \tag{6.4.12}$$

where $t \in \mathbb{R}$ satisfies

$$tw = x_0 - y \tag{6.4.13}$$

and hence

$$t(w^T \cdot w) = w^T \cdot x_0 - w^T \cdot y. \tag{6.4.14}$$

Recalling that $\| w \| = 1 = \sqrt{w^T \cdot w}$ we have

$$\begin{aligned} t &= \theta - y^T \cdot w \\ &= \varepsilon_\theta + \tilde{\theta} - y^T \cdot \tilde{w} - y \cdot \varepsilon_w \\ &= \varepsilon_\theta - y \cdot \varepsilon_w. \end{aligned} \tag{6.4.15}$$

Therefore we find that

$$\begin{aligned} d(H,\tilde{H}) &\leq \sqrt{n}\sup_{y\in\tilde{H}} |\, \varepsilon_\theta - y\cdot\varepsilon_w \,| \\ &\leq \sqrt{n}\sup_{y\in J} |\, \varepsilon_\theta - y\cdot\varepsilon_w \,|\,. \end{aligned} \tag{6.4.16}$$

Note that the supremum in (6.4.16) will be attained when $y = (\rho,\ldots,\rho)^T$, and the weight errors are of similar sign, and of opposite sign to the offset error. Hence the following inequality is obtained:

$$d\,(H, \tilde{H}) \;\leq\; \sqrt{n}\left(|\varepsilon_\theta| + \rho \sum_{i=1}^{n} |\varepsilon_{w_i}|\right) \tag{6.4.17}$$

It is evident that if equation (6.4.11) is satisfied, then $d\,(H, \tilde{H}) \leq \varepsilon_h$, and we have established the lemma.

∎

Corollary 6.2 *Consider the special case where the elements of the weight error and of the offset error have similar magnitude, so that $\varepsilon_\omega = \varepsilon_\theta = \varepsilon_{w_i}$ for $i = 1, \ldots, n$ (where $\varepsilon_w^T = (\varepsilon_{w_1}, \ldots, \varepsilon_{w_n})$), then the hyperplane error will be less than ε_h if*

$$\varepsilon_\omega \leq \frac{\varepsilon_h}{\sqrt{n}(1 + n\rho)} \tag{6.4.18}$$

□

Effects of Errors on the Step Size s_α

Let

$$\varepsilon_l \overset{\triangle}{=} \varepsilon_\alpha + \varepsilon_h \tag{6.4.19}$$

denote the total error in the $\tilde{l}_\alpha(f)$ approximations to $\bar{l}_\alpha(f)$ and use ε_r to denote the representation error:

$$\varepsilon_r \overset{\triangle}{=} \sup_{x \in R} \left| f(x) - \tilde{f}^{\mathrm{NN}}(x) \right|. \tag{6.4.20}$$

Lemma 6.3 *Consider $f \in F_{L,C}^{\rho,n}$. To achieve a given representation error, ε_r (6.4.20), s_α must satisfy the relationship*

$$s_\alpha \leq 2(\varepsilon_r - L\varepsilon_l). \tag{6.4.21}$$

□

Proof. By the Lipschitz condition on f the maximum error in $\tilde{f}(x)$ caused by an error ε_l in x is $L\varepsilon_l$ (i.e. $\| \tilde{x} - x \|_\infty = \varepsilon_l$). The error caused by the quantization process is at most

$\frac{s_\alpha}{2}$ (and not s_α because the quantization error can be of either sign). Thus the total error in $\tilde{f}$ is $\varepsilon_r = \frac{s_\alpha}{2} + L\varepsilon_l$. If s_α is set less than or equal to $2(\varepsilon_r - L\varepsilon_l)$, then (6.4.20) will be met. ∎

Note that errors in the hyperplane positions could cause the approximations $\tilde{c}_j^{(\alpha_i)}$ of $c_m^{(\alpha_i)}$ to have "gaps" in them because of misalignment of the hyperplanes. However we can ignore any such errors as long as (6.4.20) is satisfied. In any case, if we share the hyperplanes as much as possible, "gaps" will rarely occur.

6.4.3 Construction of the "Worst Case" $f \in F_{L,C}^{\rho,n}$ for an N-ANN

Consider now the number of nodes necessary in a N-ANN in order to achieve an ε_r-approximation of *any* $f \in F_{L,C}^{\rho,n}$. We do this by considering the worst function to represent in the given class.

Arrangement of the Hyperplanes

Consider the choice and arrangement of the hyperplanes in the first layer. Observe that if it is possible to *share* hyperplanes between the subnets NN_i, then a reduced number of nodes will be needed in total. The structure which allows the greatest degree of sharing has an architecture in which the first layer is common to all of the subnets NN_i. The hyperplanes in this first layer are arranged to form a regular $2\varepsilon_l$-lattice, $\mathcal{L}_{2\varepsilon_l}$ on $[0,\rho]^n$. There are $\left\lfloor \frac{\rho}{2\varepsilon_l} \right\rfloor$ hyperplanes $H_{U_j,\theta}$ parallel to each of the $(n-1)$-dimensional hyperplanes $H_{U_j,0}$. This leads to a total of

$$\nu_1 = \left\lfloor \frac{n\rho}{2\varepsilon_l} \right\rfloor \tag{6.4.22}$$

hyperplanes and hence ν_1 nodes in the first layer. The approximations $\tilde{l}_{\alpha_i}(f)$ to $\bar{l}_{\alpha_i}(f)$ will be finite unions of n-rectangular regions formed by the $2n$-fold intersections of the half spaces formed by the hyperplanes.

Note that if arbitrary orientations are allowed it is easy to show (using [3, p.27, eq.57]) a *lower bound* for the number of $2\varepsilon_l$-distinguishable hyperplanes is $\left(\frac{\rho}{4\varepsilon_l}\right)^n$, so there is far less opportunity for sharing hyperplanes in that case. Since, as shall be seen below, the number of nodes in the first layer is dominated by the number of nodes in the the second layer, The lattice structure is preferred and is assumed from now on. We will denote such ANNs as $\mathcal{L}_{2\varepsilon_l}$-$N$-ANNs.

Two plausible worst case functions are now constructed and the costs associated with each determined.

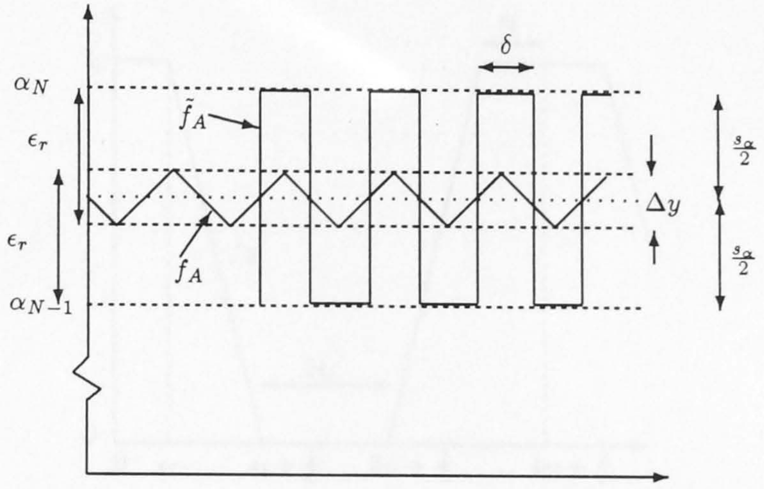

Figure 6-3: The function f_A. Note that $\Delta y = \varepsilon_r - \frac{s_\alpha}{2}$ and thus using (6.4.21) with equality, $\Delta y = L\varepsilon_l$ and thus $\delta = 2\varepsilon_l$.

Case A

The idea here is to construct a function f_A which swings above and below a certain level α_i "as fast as possible" in the sense that the maximum number of ε_l-distinguishable components of $\bar{l}_{\alpha_i}(f)$ are created. The idea can be easily seen in one dimension in figure 6-3.

The number of ε_l-distinguishable components of $\bar{l}_{\alpha_N}(f_A)$ is $\left\lfloor \frac{\rho}{4\varepsilon_l} \right\rfloor^n$, and since each component is an n-rectangle (and thus the (convex) intersection of half spaces defined by hyperplanes in $\mathcal{L}_{2\varepsilon_l}$), the number of nodes in the second layer is

$$\nu_2^A = (N-1) + \left\lfloor \frac{\rho}{4\varepsilon_l} \right\rfloor^n, \tag{6.4.23}$$

where the $(N-1)$ term accounts for the single component (comprising the whole domain $[0,\rho]^n$) of $\bar{l}_{\alpha_i}(f_A)$, for $i = 1, \ldots, N-1$. Note that since $\delta = 2\varepsilon_l$ (see figure 6-3), there does not exist a function of this general form with more components.

Case B

The idea behind case B is to construct a function f_B which swings up and down as far as possible as rapidly as the Lipschitz condition allows, thus crossing as many levels α_i as possible as often as possible. In one dimension f_B is as shown in figure 6-4

Obviously each "cycle" (in each dimension) is of length $4\varepsilon_l + \frac{2}{L} = \Delta$. There are N components of the α_i-above-sets $\bar{l}_{\alpha_i}(f_B)$ formed in each cycle. Noting that partial cycles

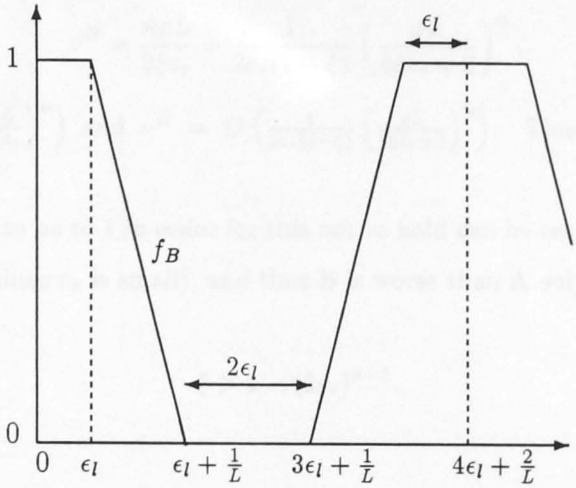

Figure 6-4: The function f_B when $n = 1$.

are also possible, the total number of components for all levels α_i, $i = 1, \ldots, N$ formed by f_B is $\left\lfloor N\left(\frac{\rho}{4\varepsilon_l + 2/L}\right)^n\right\rfloor$, and since all the components are n-rectangles, this is the number of nodes needed in the second layer for case B:

$$\nu_2^B = \left\lfloor N\left(\frac{\rho}{4\varepsilon_l + \frac{2}{L}}\right)^n\right\rfloor. \tag{6.4.24}$$

Lemma 6.4 *Case A is worse than case B for all reasonable values of s_α and ε_l. That is case A requires more nodes than case B.* □

Proof. Let ν^A and ν^B denote the total number of nodes for cases A and B respectively. From (6.4.22), (6.4.23) and (6.4.24)

$$\nu^A = \frac{n\rho}{2\varepsilon_l} + (N-1) + \left(\frac{\rho}{4\varepsilon_l}\right)^n \tag{6.4.25}$$

and

$$\nu^B = \frac{n\rho}{2\varepsilon_l} + N\left(\frac{\rho}{4\varepsilon_l + \frac{2}{L}}\right)^n \tag{6.4.26}$$

and the constraint that $s_\alpha = 2(\varepsilon_r - \varepsilon_l L)$. Setting $\varepsilon_l = \xi\frac{\varepsilon_r}{L}$ with $\xi \in (0,1)$ implies $s_\alpha = 2\varepsilon_r(1-\xi)$. This means that

$$\nu^A = \frac{n\rho}{2\xi\varepsilon_r} + \frac{1}{2\varepsilon_r(1-\xi)} - 1 + \left(\frac{\rho L}{4\xi\varepsilon_r}\right)^n \tag{6.4.27}$$

and

$$\nu^B = \frac{n\rho L}{2\xi\varepsilon_r} + \frac{1}{2\varepsilon_r(1-\xi)}\left(\frac{\rho L}{4\xi\varepsilon_r + 2}\right)^n . \tag{6.4.28}$$

Thus $\nu^A = O\left(\left(\frac{\rho L}{4\xi\varepsilon_r}\right)^n\right)$ and $\nu^B = O\left(\frac{1}{2\varepsilon_r(1-\xi)}\left(\frac{\rho L}{4\xi\varepsilon_r+2}\right)^n\right)$. Thus unless $1-\xi \ll 1$, $\nu^A \gg \nu^B$.

How close ξ has to be to 1 in order for this not to hold can be seen as follows. If $\xi \approx 1$, then $2 + 4\xi\varepsilon_r \approx 2$ (since ε_r is small), and thus B is worse than A only if $\frac{1}{2\varepsilon_r(1-\xi)} > (\frac{1}{2\varepsilon_r})^n$. That is if

$$\xi > 1 - (2\varepsilon_r)^{n-1}. \tag{6.4.29}$$

■

Lemma 6.5 *The number of nodes, $\nu^A(m)$, is minimised by setting $\varepsilon_l = (1-2^{-m})\varepsilon_r/L$, where $m = \frac{1}{2}\log\left(n\varepsilon_r\left(\frac{\rho L}{4\varepsilon_r}\right)^n\right)$. In this case*

$$\nu^A = O\left(\frac{n}{\sqrt{2}}\left(\frac{\rho L}{4\varepsilon_r}\right)^n\right). \tag{6.4.30}$$

□

Remark 6.3 Theorem 6.8 is an immediate consequence of this lemma.

Proof. Note that while it is possible to differentiate (6.4.25) with respect to ε_l and set to zero, the resulting equation is not solvable algebraically. Instead note that the two effects due to terms (1 and 3) and 2 in (6.4.25), respectively are monotonic. Set

$$\varepsilon_l = (1-2^{-m})\varepsilon_r/L \quad m = 1, 2, \ldots \tag{6.4.31}$$

and thus

$$s_\alpha = 2^{-m+1}\varepsilon_r. \tag{6.4.32}$$

Now determine the relative decrease (from m to $m+1$) of the first and last terms, and the relative increase of the second term of $\nu^A(m)$. When these are roughly the same in magnitude, we must be near the minimum of ν^A as a function of ε_l. Substituting for ε_l, s_α in (6.4.25) gives

$$\nu^A(m) \;=\; \frac{n\rho L}{2(1-2^{-m})\varepsilon_r} + \frac{2^{m-1}}{\varepsilon_r} - 1 + \left(\frac{\rho L}{4\varepsilon_r(1-2^{-m})}\right)^n$$

$$= \frac{2^m}{(2^m-1)} \frac{n\rho L}{2\varepsilon_r} + \frac{2^{m-1}}{\varepsilon_r} + \left(\frac{\rho L}{4\varepsilon_r}\right)^n \left(\frac{2^m}{2^m-1}\right)^n - 1 \qquad (6.4.33)$$

Note that the first term is changing slowly with m in comparison with the second and third terms. Thus we find the value of m for which the decrease in the third term is approximately the same as the increase in the third term.

Note that

$$\left(\tfrac{2^m}{2^m-1}\right)^n \approx \left(\tfrac{2^m+1}{2^m}\right)^n = (1+\tfrac{1}{2^m})^n \approx (1+\tfrac{n}{2^m}) \overset{\triangle}{=} \chi(m).$$

Now $\chi(m+1)-\chi(m) = (1+\frac{n}{2^{m+1}}) - (1+\frac{n}{2^m}) = \frac{-n}{2^{m+1}}$. Thus the third term decreases by $\frac{n}{2^{m+1}}\left(\frac{\rho L}{4\varepsilon_r}\right)^n$ as m goes to $m+1$.

The second term is $\frac{2^{m-1}}{\varepsilon_r} - 1$, and we will ignore the -1. Obviously as m goes to $m+1$ this *increases* by $\frac{2^m - 2^{m-1}}{\varepsilon_r} = \frac{2^{m-1}}{\varepsilon_r}$.

To determine m such that these relative increases are the same, solve

$$\frac{2^{m-1}}{\varepsilon_r} = \frac{n}{2^{m+1}} \left(\frac{\rho L}{4\varepsilon_r}\right)^n \qquad (6.4.34)$$

for m. Rearrangement of (6.4.34), followed by taking logarithms gives

$$m = \tfrac{1}{2}\log\left(n\varepsilon_r \left(\tfrac{\rho L}{4\varepsilon_r}\right)^n\right). \qquad (6.4.35)$$

We can now substitute the value of ε_l and s_α given by (6.4.35), (6.4.31) and (6.4.32). First we rewrite (6.4.33) as

$$\nu^A \approx (1+2^{-m})\frac{n\rho L}{2\varepsilon_r} + (1+2^{-m})^n \left(\frac{\rho L}{4\varepsilon_r}\right)^n + \frac{2^{m-1}}{\varepsilon_r}. \qquad (6.4.36)$$

Noting that $(1+2^{-m})^n \approx (1+n2^{-m})$, we can thus substitute to obtain

$$\begin{aligned} \nu^A &= \left(1 + \frac{1}{\sqrt{2}\,n\varepsilon_r}\left(\frac{4\varepsilon_r}{\rho L}\right)^n\right)\left(\frac{n\rho L}{2\varepsilon_r}\right) \\ &\quad + \left(1 + \frac{n}{\sqrt{2}}\frac{1}{n\varepsilon_r}\left(\frac{4\varepsilon_r}{\rho L}\right)^n\right)\left(\frac{\rho L}{4\varepsilon_r}\right)^n + \frac{\sqrt{2}}{2}\frac{n\varepsilon_r\left(\frac{\rho L}{4\varepsilon_r}\right)^n}{\varepsilon_r} \\ &= \frac{n\rho L}{2\varepsilon_r} + \frac{4^n\varepsilon_r^{n-1}}{2\sqrt{2}\rho^{n-1}L^{n-1}} + \left(\frac{\rho L}{4\varepsilon_r}\right)^n + \frac{1}{\sqrt{2}\varepsilon_r} + \frac{n}{\sqrt{2}}\left(\frac{\rho L}{4\varepsilon_r}\right)^n . \qquad (6.4.37) \end{aligned}$$

Neglecting the (very small) second term we can thus write

$$\nu^A \approx \frac{n\rho L}{2\varepsilon_r} + \frac{1}{\sqrt{2}\varepsilon_r} + \left(1 + \frac{n}{\sqrt{2}}\right)\left(\frac{\rho L}{4\varepsilon_r}\right)^n \tag{6.4.38}$$

$$= O\left(\frac{n}{\sqrt{2}}\left(\frac{\rho L}{4\varepsilon_r}\right)^n\right). \tag{6.4.39}$$

■

6.4.4 The Bit Complexity of an $\mathcal{L}_{2\varepsilon_l}$-$N$-ANN ε-representation

Theorem 6.8 would appear to be one of the first explicit upper bounds on the number of nodes needed by a neural net to represent a function to within a specified accuracy. However, it is not just the number of nodes that determines the complexity of a net. A more reasonable idea is to determine the number of bits required to specify a net which represents any function from a given function class. Clearly some measure of information required to specify the network is needed, and the number of bits needed to specify any given net from a specified class seems appropriate.

Theorem 6.9

The number of bits needed to represent the weights of an $\mathcal{L}_{2\varepsilon_l}$-$N$-ANN in order to represent an arbitrary $f \in F^{\rho,n}_{L,C}$ to within ε_r in the sup-metric is

$$\beta = O\left(\frac{n^2}{4^n}\left(\frac{\rho L}{\varepsilon_r}\right)^{n+1}\right). \tag{6.4.40}$$

□

Proof. Recall that the number of nodes required in the net is given by (6.4.1), now calculate the number of weights in the net. The number of nodes in the i^{th} layer of the net is denoted ν_i, and the number of weights in the i^{th} layer is denoted μ_i.

The number of weights in the first layer is given by the number of nodes in the first layer times the number of inputs to each node. However, since the $\mathcal{L}_{2\varepsilon_l}$ lattice structure has been adopted, all but one of the weights for each node will be zero. Thus there will be one weight and the threshold for each node, giving a total of

$$\mu_1 = n\nu_1 = n^2 \left\lfloor \frac{\rho}{2\varepsilon_l} \right\rfloor \tag{6.4.41}$$

Since it is possible for any node in the first layer to be connected to any node in the second layer, μ_2 will be given by

$$\mu_2 = \nu_1 \times \nu_2 = \left(n \left\lfloor \frac{\rho}{2\varepsilon_l} \right\rfloor\right) \times \left[(N-1) + \left\lfloor \frac{\rho}{4\varepsilon_l} \right\rfloor^n\right]. \tag{6.4.42}$$

From corollary 6.2, $\varepsilon_w = \varepsilon_h/(\sqrt{n}(1+n\rho))$. Recalling that $\| w \| \leq 1$, in order to achieve a hyperplane accuracy of ε_h, it is necessary to have $\left\lceil \log \frac{\sqrt{n}(1+n\rho)}{\varepsilon_h} \right\rceil$ bits per weight. This gives a total of

$$\beta_1 = \left\lceil \log \frac{\sqrt{n}(1+n\rho)}{\varepsilon_h} \right\rceil 2n \left\lfloor \frac{\rho}{2\varepsilon_l} \right\rfloor \tag{6.4.43}$$

bits for the first layer.

To calculate the number of bits per weight in the second layer, observe that the weights on all the possible $\nu_1\nu_2$ interconnections between the first and second layer need to be able to take one of the three possible values $\{1, 0, -1\}$. There is a 1 representing the output from $S(h)$, a 0 if there is no connection, and a -1 if the hyperplane is actually $-S(h)$. This last case is necessary if there is to be the maximum possible sharing of hyperplanes that the lattice $\mathcal{L}_{2\varepsilon_l}$ allows. Thus β_2, the number of bits required for the second layer is given by

$$\beta_2 = (\lceil \log 3 \rceil \nu_1 + \text{threshold term})\,\nu_2. \tag{6.4.44}$$

The threshold term is the number of bits needed to represent the threshold $(M - \frac{1}{2})$ required to implement the logical "or" in (6.4.3)). While in general M *could* reach ν_1, for the function f_A, M is clearly bounded above by $2n$ for every node. In order to represent $(2n - \frac{1}{2}) \pm \frac{1}{2}$, $\lceil \log(4n-1) \rceil$ bits are required.

Thus the total number of bits required for the second layer is

$$\beta_2 = \left(2n \left\lfloor \frac{\rho}{2\varepsilon_l} \right\rfloor + 2 + \lceil \log n \rceil\right) \left((N+1) + \left\lfloor \frac{\rho}{4\varepsilon_l} \right\rfloor^n\right). \tag{6.4.45}$$

The total number of bits required β is now given by $\beta_1 + \beta_2$. In order to calculate the asymptotic upper bound, drop the floor and ceiling operators and write

$$\beta \leq \left[\log\left(\tfrac{\sqrt{n}(1+n\rho)}{\varepsilon_h}\right) + 1\right] \frac{n^2\rho}{2\varepsilon_l} + \left(\frac{n\rho}{\varepsilon_l} + 3 + \log n\right) \left((N-1) + \left(\frac{\rho}{4\varepsilon_l}\right)^n\right). \tag{6.4.46}$$

Let

$$t_1 = \left[\log\left(\tfrac{\sqrt{n}(1+n\rho)}{\varepsilon_h}\right) + 1\right] \frac{n^2\rho}{2\varepsilon_l} = O\left(\log\left(\tfrac{\sqrt{n}\,n\rho}{\varepsilon_h}\right) \frac{n^2\rho}{2\varepsilon_l}\right) \tag{6.4.47}$$

and

$$t_2 = \left(\frac{n\rho}{\varepsilon_l} + 3 + \log n\right)\left((N-1) + \left(\frac{\rho}{4\varepsilon_l}\right)^n\right) = O\left(\frac{n\rho}{\varepsilon_l}\left(\frac{\rho}{4\varepsilon_l}\right)^n\right) \tag{6.4.48}$$

and let $N = \frac{1}{s_\alpha} = \frac{1}{2(\varepsilon_r - L\varepsilon_l)}$. Consider what value of ε_h makes $t_1 \approx t_2$. Solving

$$\log\left(\tfrac{\sqrt{n}\,n\rho}{\varepsilon_h}\right)\frac{n^2\rho}{2\varepsilon_l} = \frac{n\rho}{\varepsilon_l}\left(\frac{\rho}{4\varepsilon_l}\right)^n \tag{6.4.49}$$

for ε_h gives

$$\varepsilon_h = \sqrt{n}\,n\rho 2^{-\left[\frac{2}{n}\left(\frac{\rho}{4\varepsilon_l}\right)^n\right]} \tag{6.4.50}$$

which is a *very* small number.

Hence as long as ε_h is small enough, but not as small as (6.4.50), then $t_2 > t_1$, and hence

$$\beta = O\left(\frac{n\rho}{\varepsilon_l}\left[\frac{1}{2(\varepsilon_r - L\varepsilon_l)} + \left(\frac{\rho}{4\varepsilon_l}\right)^n\right]\right). \tag{6.4.51}$$

It is now possible to effectively ignore the ε_h error. Recalling the argument used at the end of Section 4, (6.4.51) can be roughly minimised by setting $\varepsilon_l = (1 - 2^{-m})\frac{\varepsilon_r}{L}$ and $s_\alpha = 2^{-m+1}\varepsilon_r$, where $m = \frac{1}{2}\log\left(n\varepsilon_r\left(\frac{\rho L}{4\varepsilon_r}\right)^n\right)$. Upon substituting into (6.4.51)

$$\begin{aligned}
\beta &= O\left((1+2^{-m})\frac{n\rho L}{\varepsilon_r}\left[\frac{2^{m-1}}{\varepsilon_r} + (1+2^{-m})^n\left(\frac{\rho L}{4\varepsilon_r}\right)^n\right]\right) \\
&= O\left(\left(1 + \frac{1}{\sqrt{2}\,n\varepsilon_r}\left(\frac{4\varepsilon_r}{\rho L}\right)^n\right)\frac{n\rho L}{\varepsilon_r}\left[\frac{\frac{1}{2}\sqrt{2}n\left(\frac{\rho L}{4\varepsilon_r}\right)^n}{\varepsilon_r} + \left[1 + \frac{n}{\sqrt{2}}\frac{1}{n\varepsilon_r}\left(\frac{4\varepsilon_r}{\rho L}\right)^n\right]\left(\frac{\rho L}{4\varepsilon_r}\right)^n\right]\right) \\
&= O\left(\left(\frac{n\rho L}{\varepsilon_r} + \frac{4^n\varepsilon_r^{n-2}}{\sqrt{2}\rho^{n-1}L^{n-1}}\right)\left[\left(1 + \frac{n}{\sqrt{2}}\right)\left(\frac{\rho L}{4\varepsilon_r}\right)^n + \frac{1}{2\varepsilon_r}\right]\right) \\
&= O\left(\frac{n\rho L}{\varepsilon_r}\left(1 + \frac{n}{\sqrt{2}}\right)\left(\frac{\rho L}{4\varepsilon_r}\right)^n\right) \\
&= O\left(\frac{n^2}{4^n}\left(\frac{\rho L}{\varepsilon_r}\right)^{n+1}\right)
\end{aligned} \tag{6.4.52}$$

■

Remark 6.4 Note that (6.4.40) is suboptimal by a factor of $\frac{\rho L}{\varepsilon_r}$ when compared with $\mathcal{H}_\varepsilon(F^{\rho,n}_{L,C})$ (see (6.2.10)).

6.5 Conclusions

In this chapter we have calculated the number of nodes needed to represent an arbitrary function from the class of multi-dimensional Lipschitz continuous functions over a compact domain. An ANN architecture has been proposed, taking advantage of the work by Kolmogorov which proves that any function may be represented in terms of its level sets. We have pointed out the connection between the number of bits required to represent the parameters and the ε-entropy of the function class from which the function to be approximated is drawn. The scheme proposed is thus seen to be close to optimal, in the sense that the bit representation of our scheme is only a factor of $\frac{\rho L}{\varepsilon_r}$ more complicated than the best possible representation. It should be noted that all of the results are for approximation in the ∞-norm, or supremum metric. From a statistical point of view the L_2, or weighted L_2 metrics are of more interest as they lead naturally to a least squares approach to the problem, with weightings according to the distribution of the data. However such a treatment of the representation problem is beyond the scope of this work.

Chapter 7

Conclusions

7.1 Overview of the Thesis

The thesis begins with a factorization approach to the stabilization of nonlinear systems. The problem is first considered via an input-output approach, and then through a state space formulation, thus demonstrating that system identification is also relevant. An initial study of an approach using Artificial Neural Networks (ANNs) within a Recursive Prediction Error (RPE) algorithm is presented. This in turn leads to an investigation of the representation power of ANNs.

Nonlinear generalizations of the definitions of right and left coprimeness are presented. It is demonstrated that well-posedness and stability of the system is sufficient for the existence of these factorizations. The connections between matrices of coprime factors and the stability and well-posedness of the system are also studied. The stability of these matrix operators is seen to relate to the stability of the system. Contrary to the linear case, it is found that the stability of the inverses of the matrices of *lcfs* and *rcfs* may not be simply related. This is due to the requirement of superposition in deriving many of the left factorization results. Thus it is made evident that it is necessary to distinguish between the right and left factorization approaches to the general problem.

Past approaches have taken a right factorization approach using the Bezout identity. These have ultimately required the assumption of linearity in the plant, controller, or both in order to derive results similar to those of the Youla-Kucera parameterization for linear systems. This is avoided here by taking a left factorization approach, then differential boundedness assumptions on the left factors are required.

The linear results are generalized to give the class K_Q of all controllers which stabilize a given plant, and dually the class G_S of all plants stabilized by a given controller. A generalization of Linear Fractional Mappings gives a bijection between K_Q and the class of all stable operators Q. Dually, a bijection between G_S and S is given. These classes of plants and controllers are then combined to prove that the system $\{G_S, K_Q\}$ is stable if and only if the system $\{S, Q\}$ is stable. Two nonlinear robust stabilization results follow. The preceding results are derived via left factorizations, so the only regularity assumptions required are that the factors be differentially bounded.

Having conducted a detailed study of the problem from an input-output point of view, state space results are then presented. Note that once realizations for the factorizations have been given, the theory developed from the input-output point of view may be applied. Hence the problem of deriving these realizations is considered. It is shown that right coprime factorizations may be derived when a solution to the smooth stabilization problem may be found. A static stabilizing state feedback map is used to give the right factors of the nonlinear operator. This leads to an approach to the design of a stabilizing controller for the plant. By assuming that a state estimator may be designed, the stabilizing state feedback map may be used to provide a stabilizing input to the plant. The controller thus constructed is seen to have a right factorization.

Obtaining left factorizations for a given operator is seen to be less tractable. A specialization is presented which allows the construction of a left factorization when the plant is augmented by a unity feed-through term.

Completing the first part of the thesis on the factorization approach to the stabilization of nonlinear systems, a right factorization of the universally stabilizing controller due to Nussbaum is presented.

The requirement for an accurate model in deriving a state feedback map demonstrates the need for a system identification scheme. Thus the second part of the thesis is motivated, giving an approach to the nonlinear system identification problem. The power of Ljung's theory in giving the convergence of RPE schemes when the parameterization is nonlinear leads us to use such an algorithm for the nonlinear problem. ANNs have been seen to give a convenient representation of the class of all continuous maps between real vector spaces, hence they are used for function estimators within the identification scheme. It is demonstrated that the algorithm presented satisfies the convergence requirements of the theory, and thus may be used for the identification of some nonlinear systems.

The use of ANNs as function estimators in such a scheme leads us to consider exactly how large a network must be in order to adequately estimate a given function. An architecture is proposed based on the idea of representing a function in terms of its level sets. This structure is easily analysed, giving the number of nodes which are required to estimate any function from a given function class.

7.2 Further Research

There is scope for future work arising from the results presented in all the areas studied. This research may be divided roughly into the areas of factorization theory, RPE convergence when using ANN as estimators and ANN representation theory.

7.2.1 Coprime Factorizations

There are many areas for further research continuing from the work presented in chapters 2 to 4, in terms of extending the existing factorization theory and in applying the theory which has already been developed.

Most of the work developed in these chapters may be considered to apply in either discrete or continuous time as the proofs use techniques which are independent of the input and output spaces. This requires some formalization. Specifically, some of the earlier results due to Hammer, presented in Chapter 2, have no continuous time analogues. Before the nonlinear factorization theory can be considered complete, continuous time versions of these results will need to be proved.

The proofs of the results of Chapter 3 do not appeal to the definition of left coprimeness presented, but instead appeal to the use of a Bezout identity, or the stability of the inverse of the matrix formed by the left factors of the plant and controller. It is not clear whether a set-theoretic, or algebraic definition should be used. Also, it is not known if there exists a set theoretic definition for left coprimeness which is equivalent to the algebraic expressions which are necessary to the proofs.

To date it has not been possible to develop a dual result to Theorem 3.4, p. 50 using right factorizations. This may now be possible to prove using the generalization of linear fractional maps of Section 3.6. It is not clear what additional restrictions on the right factors will be necessary to prove such a result, but it is likely that a condition such as differential boundedness will be required. This issue has become more important since the results of Chapter 4 have shown the ease of constructing right factorizations.

The foremost problem in the state space approach to the factorization theory lies in developing techniques for obtaining a left factorization of a general nonlinear system.

Results quoted solve the problem for a special case. This indicates that there may be a more general approach to deriving left factorizations.

As the linear results rely on adding the states of the factors to prove that the plant G is equivalent to $\tilde{M}^{-1}\tilde{N}$, it would appear that some form of decomposition of the state equation of the plant may be appropriate. It may be possible to apply the results of Center Manifold Theory in this case. An approach which may give some intuition for the problem is to give expressions for the factors $\tilde{M}$ and $\tilde{N}$, and then investigate the properties of the operator $\tilde{M}^{-1}\tilde{N}$ thus formed.

Another approach to this problem is to consider it in terms of the design of a stable, stabilizing, post-filter for the plant. Once obtained, such an operator gives a candidate for $\tilde{M}$, and thus $\tilde{N} = \tilde{M}G$. This may be possible through the design of a stable state estimator.

Supposing that the problem for deriving left factorizations for the plant and controller is solved, it will still be necessary to ensure that these maps are differentially bounded. Thus it will be necessary to derive state space conditions which ensure differential boundedness of the operator. An alternative is to derive a Lipschitz constant for the operator, based on the realization. This appears possible for discrete time, stable systems, but a general approach is not clear.

State space techniques may offer a solution to the problem mentioned earlier of deriving a more complete theory based on right factorizations. Once the right factorizations have been expressed, algebraic techniques might give rise to more general results. Initial investigations that I have carried out into the area yield an algebraic requirement on the output maps of the plant and controller. There exists a simple example for which this requirement is not satisfied, so it is clear that some restrictions on these output maps will be required.

The greatest scope for research arising from this work lies in the application of the techniques developed to other problems. Part of the motivation for developing this theory was to provide a method for solving problems in Robust and Adaptive Control. Excluding the robust stabilization results presented in Chapter 3, the application of these results has yet to be considered.

Another area of application for this theory is in providing an approach to solving the nonlinear regulator problem.

7.2.2 Nonlinear System Identification via ANNs

The work presented on the identification problem represents only preliminary studies. There are many further cases which could be examined in order to develop an understanding of the general problem. For instance, it is possible to represent the state and output relations by fewer neural networks. These model architectures have yet to be investigated, so it is not clear what benefits such an approach may yield.

The bounds on the Lipschitz constants derived for the observer equation merit further study. The approach taken was somewhat naive, and it may be possible to further tighten these bounds. This would have the effect of increasing the domain of exponential stability, expanding the range of systems which may be estimated.

Additionally there is scope for further simulation studies. Due to the algebraic complexity of the problem, Monte Carlo simulations could be the best way to examine the properties of the algorithm presented.

It is not clear how the convergence requirements presented in Chapter 5 should be best applied, or if they are too restrictive for the case we are studying. Currently in order to guarantee convergence of the proposed scheme, an exponential stability domain must be derived. This involves taking second derivatives of the output of a number of neural networks, yielding highly complex equations. A more thorough investigation of Ljung's theory is indicated. Through such a study less restrictive conditions on the parameter vector may be obtained, potentially increasing the domain of application for this algorithm.

It is apparent from the simulation studies carried out so far that substantially different weight sets can give rise to the same input-output map. As yet it is not clear how this affects the convergence properties of the algorithm.

Another difficulty is determining the level of accuracy with which we have to estimate the state equations. For feedforward equations, the effect of taking an approximation is easily seen and understood. For a nonlinear dynamical system it is not clear that having a given level of approximation for the state and output equations guarantees a finite error in the output.

Note that the proposed scheme acts as a dual parameter and state estimator. It may be possible to use this scheme to derive a nonlinear state estimator, as is required in the design of the candidate controller of Chapter 4.

The application of the results to an adaptive scheme has not yet been considered. We have the option of combining this with the factorization results obtained so far to guarantee stability for the system. Our algorithm could be used in an adaptive Q, estimating the S which represents the difference between the actual and modelled plant.

As a method for identifying unstable plants, it may be possible to use the identification scheme presented to estimate instead the stable factors of the plant.

7.2.3 ANNs and Functional Representation

The work that has been completed leaves a number of unanswered questions and problems to be investigated. In this chapter only the use of a piecewise constant activation function has been considered, it is conceivable that there may be some benefit derived from using a piecewise linear, or quadratic activation function instead. This would generalize to show the possible benefits to be derived from considering a smooth activation function. Such results may accommodate the calculation of the number of nodes required to represent an arbitrary function from the class of functions which has a Lipschitz condition on the first or higher derivatives. A first step would be to prove that it is possible to attain the same bound for the number of nodes required to achieve a given level of approximation using smooth activation functions in the nodes of the proposed architecture.

Additionally the results given account for the worst case in the class of functions being represented. If a probability measure is defined on this functional class, it may be that the average number of nodes required to represent a member of the class is significantly less than the bound given. The effects of the probability measure used, and in fact what measure would be appropriate have not been studied.

The results presented are entirely constructive and theoretical. Examples have yet to be constructed to illustrate that this scheme will work. Additionally the problem of learning has been left untouched. As the construction is based on level sets of the function being estimated it seems that the algorithms developed for learning decision regions are appropriate.

Bibliography

[1] V.I. Arnol'd. Representation of continuous functions of three variables by the superposition of continuous functions of two variables. *Matematicheshii Sbornik (N.S.)*, 48:3–74, 1959.

[2] S.M. Carroll and B.W. Dickinson. Construction of neural nets using the radon transform. *International Conference on Neural Networks*, 1989.

[3] J. H. Conway and N. J. A. Sloane. Sphere packings, lattices and groups. *Springer-Verlag, New York*, 1987.

[4] G. Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals, and Systems*, 2:303–314, 1989.

[5] Mark J. Damborg, Robert C. Williamson, Andrew D. B. Paice, and John B. Moore. Adaptive nonlinear estimation with artificial neural networks. *Proc. ISITA 1990*, pages 743–746, 1990.

[6] C. A. Desoer. Right coprime factorizations of a class of time-varying nonlinear systems. *Technical Report, Electronics Research Laboratory, University of California.*, 1987.

[7] C. A. Desoer and A. N. Gundes. Bicoprime factorizations of the plant and their relation to right- and left-coprime factorizations. *Technical Report, Electronics Research Laboratory, University o f California.*, 1987.

[8] Doyle. ONR/Honeywell workshop, October 1984.

[9] R. M. Dudley. Metric entropy of some classes of sets with differentiable boundaries. *Journal of Approximation Theory*, 10:227–236, 1974. Corrections in, **26**, 1979, pp. 192–193.

[10] R. M. Dudley. A course on empirical processes. In R. M. Dudley, H. Kunitay, and F. Ledrappier, editors, *École d'Été de probabilités de Saint-Flour XII-1982*, volume 1097 of *Lecture Notes in Mathematics*, pages 1–142. Springer-Verlag, Berlin, 1984.

[11] K. I. Funahashi. On the approximate realization of continuous mappings by neural networks. *Neural Networks*, 2:183–192, 1989.

[12] J. Hammer. Nonlinear systems, additive feedback and rationality. *International Journal of Control*, 40(5):953–969, November 1984.

[13] J. Hammer. Nonlinear systems, stability and rationality. *International Journal of Control*, 40(1):1–35, 1984.

[14] J. Hammer. Nonlinear systems stabilization and coprimeness. *International Journal of Control*, 42(1):1–20, July 1985.

[15] J. Hammer. Stabilization of nonlinear systems. *International Journal of Control*, 44(5):1349–1381, November 1986.

[16] J. Hammer. Fraction representations of nonlinear systems: a simplified approach. *International Journal of Control*, 46:455–472, 1987.

[17] J. Hammer. Fraction representations of non-linear systems and non-additive state feedback. *International Journal of Control*, 50:1981–1990, 1989.

[18] J. Hammer. State feedback for non-linear control systems. *International Journal of Control*, 50:1981–1990, 1989.

[19] F. Hausdorff. Set theory. *Chelsea, New York*, 1957.

[20] R. Hecht-Nielsen. Kolmogorov's mapping neural network existence theorem. *IEEE First International Conference on Neural Networks*, 3:11–14, 1987.

[21] M. W. Hirsch and S. Smale. *Differential Equations, Dynamical Systems, and Linear Algebra.* Academic Press, New York, 1984.

[22] K. Hornik, S. Stinchcombe, and H. White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2:359–366, 1989.

[23] A.N. Kolmogorov. Asymptotic characteristics of some completely bounded metric spaces. *Doklady Akademy Nauk SSSR (N.S.) Seriia Matematika, Fizika*, 108:585–589, 1956.

[24] A.N. Kolmogorov. On the shannon theory of information transmission in the case of continuous signals. *IRE Transactions on Information Theory*, 2:102–108, 1956.

[25] A.N. Kolmogorov. On the representation of continuous functions of many variables by superposition of continuous functions of one variable and addition. *Doklady Akademy Nauk SSSR (N.S.)*, 114:953–956, 1957. Translation in, American Mathematical Society Translations, Series 2, **28**, 1963, pp. 55–59.

[26] A.N. Kolmogorov and V.M. Tihomirov. ε-entropy and ε-capacity of sets in functional spaces. *Uspehi Mat. (N.S.)*, 14:3–86, 1959. Translation in, American Mathemaical Society Translations, Series 2, **17**, 1961, pp. 277–364.

[27] A. S. Konrod. On functions of two variables. *Uspehi Mat. Nuak*, 5:24–134, 1950.

[28] A. J. Krener and Yi Zhu. The fractional representation of a class of nonlinear systems. *Proc. of the 28th CDC*, pages 963–968, 1989.

[29] L. Ljung. Theorems for the asymptotic analysis of recursive stochastic algoritms. Technical Report Report 7522, Department of Automatic Control, Lund Institute of Technology, Lund, Sweden.

[30] L. Ljung. Analysis of recursive stochastic algorithms. *IEEE Trans. on Automatic Control*, 22(4):551–575, August 1977.

[31] L. Ljung. On recursive prediction error identification algorithms. Technical Report Report LITH-ISY-I-0226, Department of Electrical Engineering, Linköping University, Sweden., 1978.

[32] L. Ljung. *Systems identification: theory for the user.* Prentice Hall, New Jersey, 1987.

[33] L. Ljung and T. Söderström. *Theory and practice of recursive identification.* MIT Press, Cambridge.

[34] G. G. Lorentz. Lower bounds for the degree of approximation. *Transactions of the American Mathematical Society*, 97:25–34, 1960.

[35] G. G. Lorentz. Metric entropy and approximation. *Bulletin of the American Mathematical Society*, 72:903–937, 1966.

[36] J. B. Moore and L. Irlicht. Coprime factorization over a class of nonlinear systems. *Submitted for publication.*, 1991.

[37] J. B. Moore and Lige XIA. On improving control-loop robustness of model-matching controllers. *Systems and Control letters*, 7:83–87, 1986.

[38] J.B. Moore and R.K. Boel. Asymptotically optimum recursive prediction error methods in adaptive estimation and control. *Automatica*, 22, No. 2:237–240, 1986.

[39] R. D. Nussbaum. Some remarks on a conjecture in adaptive control. *Systems and Control Letters*, 3:243–246, 1983.

[40] A. D. B. Paice and J. B. Moore. On the Youla-Kucera parameterization for nonlinear systems. *Systems and Control Letters*, 14:121–129, 1990.

[41] A. D. B. Paice and J. B. Moore. Robust stabilization of nonlinear plants via left coprime factorizations. *Systems and Control Letters*, 15:125–135, 1990.

[42] Andrew D. B. Paice, John B. Moore, and Roberto Horowitz. Nonlinear feedback stability via coprime factorization analysis. Accepted by Journal of Mathematical Systems Estimation and Control, October 1991.

[43] Y. C. Pati and P. S. Krishnaprasad. Analysis and synthesis of feedforward neural networks using discrete affine wavelet transformations. Technical Report TR90-44, Systems Research Center, University of Maryland, 1990.

[44] E. D. Sontag. Smooth stabilization implies coprime factorization. *IEEE Transactions on Automatic Control*, 34:435–443, 1989.

[45] E. D. Sontag. Further facts about input to state stabilization. *IEEE Transactions on Automatic Control*, 35:473–476, 1990.

[46] M. Stinchcombe and H. White. Universal approximation using feedforward networks with non-sigmoid hidden layer activation functions. *International Conference on Neural Networks*, pages I–613–I–617, 1989.

[47] T. T. Tay and J. B. Moore. Adaptive control within the class of stabilizing controllers for a time-varying nominal plant. *International Journal of Control*, 50:33–53, 1989.

[48] T. T. Tay and J. B. Moore. Enhancement of fixed controllers via adaptive disturbance estimate feedback. *Automatica*, To appear 1989.

[49] T. T. Tay and J. B. Moore. Left coprime factorizations and a class of stabilzing controllers for nonlinear systems. *International Journal of Control*, 49:1235–1248, 1989.

[50] T. T. Tay and J. B. Moore. Performance enhancements of two-degree-of-freedom controllers via adaptive techniques. *Adaptive Control and Signal Processing*, 1990. Yet to chase this up.

[51] T. T. Tay, J. B. Moore, and R. Horowitz. Indirect adaptive techniques for fixed controller performance enhancement. *International Journal of Control*, 5:1941–1959, 1989.

[52] V. M. Tikhomirov. Kolmogorov's work on ε-entropy of functional classes and the superposition of functions. *Russian Mathematical Surveys*, 18:51–87, 1963.

[53] J. F. Traub, G. W. Wasilkowski, and H. Waźniakowski. *Information Based Complexity*. Academic Press, New York, 1988.

[54] M. S. Verma. Coprime fractional representations and stability of nonlinear feedback systems. *International Journal of Control*, 48:897–918, 1988.

[55] M. S. Verma. Coprime fractional representations of nonlinear systems. *Proc. IEEE Int. Symp. on Circuits and Systems*, 3:2449, 1988.

[56] M. S. Verma and Tho Pham. Stabilization of nonlinear systems and coprime factorization. *Proc Int. Symp. MTNS-89 Volume 2*, pages 473–482, 1989.

[57] M. Vidyasagar. *Control system synthesis: A Factorization Approach.* MIT Press, Cambridge, 1985.

[58] A. G. Vitushkin. The absolute ε-entropy of metric spaces. *Doklady Akademia Nuak SSR (N.S.) Seriia Matematika, Fizika*, 117:745–747, 1957. Translation in, American Mathemaical Society Translations, Series 2, **17**, 1961, pp. 365–367.

[59] A. G. Vitushkin. Theory of the transmission and processing of information. *Pergamon Press, Oxford*, 1961. Originally published as: Otsenka slozhnosti zadachi tabulirovaniya (Estimation of the complexity of the tabulation problem), *Fizmatgiz*, Moscow, 1959.

[60] H. Weiss and J.B. Moore. Recursive prediction error algorithms without a stability test. *Automatica*, 16:683–688, 1980.

[61] P.J. Werbos. Backpropagation and neural control: A review and prospectus. *Proceedings of the International Joint Conference on Neural Networks*, pages I–209–I–216, 1989.

[62] R.C. Williamson and A.D.B. Paice. The number of nodes required in a feed-forward neural network for functional representation. Submitted to Neural Networks, 1991.

Appendix A

A.1 Factorizations of K_Q

It is shown that for the linear case, the parameterizations based on the left and right factorizations of the original controller give the same controller for each Q. That is, it is proved that

$$\begin{aligned} \tilde{V}_Q^{-1}\tilde{U}_Q &= U_Q V_Q^{-1} \\ (\tilde{V} + Q\tilde{N})^{-1}(\tilde{U} + Q\tilde{M}) &= (U + MQ)(V + NQ)^{-1} \end{aligned} \tag{A.1.1}$$

First consider the Bezout identity

$$\begin{aligned} \tilde{V}_Q M - \tilde{U}_Q N &= I \\ \tilde{V}_Q &= (I + \tilde{U}_Q N)M^{-1} \end{aligned}$$

Substituting into the left hand side of (A.1.1)

$$\tilde{V}_Q^{-1}\tilde{U}_Q = M\left(I + \left(\tilde{U} + Q\tilde{M}\right)^{-1}\right)\left(\tilde{U} + Q\tilde{M}\right)$$

Note that $(I + DC)^{-1}D = D(I + CD)^{-1}$

$$\begin{aligned} \tilde{V}_Q^{-1}\tilde{U}_Q &= M\left(\tilde{U} + Q\tilde{M}\right)\left(I + N\left(\tilde{U} + Q\tilde{M}\right)\right)^{-1} \\ &= \left(M\tilde{U} + MQ\tilde{M}\right)\left(I + N\tilde{U} + NQ\tilde{M}\right)^{-1} \end{aligned} \tag{A.1.2}$$

Reversing equation (1.1.18) gives

$$\begin{bmatrix} M & U \\ N & V \end{bmatrix} \begin{bmatrix} \tilde{V} & -\tilde{U} \\ -\tilde{N} & \tilde{M} \end{bmatrix} = I \tag{A.1.3}$$

Therefore

$$\begin{aligned} M\tilde{U} &= U\tilde{M} \\ V\tilde{M} - N\tilde{U} &= I \end{aligned}$$

Substituting into (A.1.2)

$$\begin{aligned} \tilde{V}_Q^{-1}\tilde{U}_Q &= \left(U\tilde{M} + MQ\tilde{M}\right)\left(V\tilde{M} + NQ\tilde{M}\right)^{-1} \\ &= (U + MQ)(V + NQ)^{-1} \\ &= U_Q V_Q^{-1} \end{aligned} \tag{A.1.4}$$

Which is as required. ■

A.2 Stability of $\{G_S, K_Q\}$ via left factorizations.

$$\begin{aligned} \begin{bmatrix} \tilde{V}_Q & -\tilde{U}_Q \\ -\tilde{N}_S & \tilde{M}_S \end{bmatrix}^{-1} &= \begin{bmatrix} \tilde{V} + Q\tilde{N} & -\tilde{U} - Q\tilde{M} \\ -\tilde{N} - S\tilde{V} & \tilde{M} + S\tilde{U} \end{bmatrix}^{-1} && \text{(A.2.1)} \\ &= \left\{ \begin{bmatrix} \tilde{V} & -\tilde{U} \\ -\tilde{N} & \tilde{M} \end{bmatrix} - \begin{bmatrix} -Q\tilde{N} & Q\tilde{M} \\ S\tilde{V} & -S\tilde{U} \end{bmatrix} \right\}^{-1} && \text{(A.2.2)} \\ &= \left\{ \begin{bmatrix} \tilde{V} & -\tilde{U} \\ -\tilde{N} & \tilde{M} \end{bmatrix} - \begin{bmatrix} 0 & Q \\ S & 0 \end{bmatrix} \begin{bmatrix} \tilde{V} & -\tilde{U} \\ -\tilde{N} & \tilde{M} \end{bmatrix} \right\}^{-1} && \text{(A.2.3)} \\ &= \begin{bmatrix} \tilde{V} & -\tilde{U} \\ -\tilde{N} & \tilde{M} \end{bmatrix}^{-1} \begin{bmatrix} I & -Q \\ -S & I \end{bmatrix}^{-1} && \text{(A.2.4)} \end{aligned}$$

Note that the only point at which linearity was used was in stating

$$\begin{bmatrix} -Q\tilde{N} & Q\tilde{M} \\ S\tilde{V} & -S\tilde{U} \end{bmatrix} = \begin{bmatrix} 0 & Q \\ S & 0 \end{bmatrix} \begin{bmatrix} \tilde{V} & -\tilde{U} \\ -\tilde{N} & \tilde{M} \end{bmatrix} \tag{A.2.5}$$

Hence it would seem that it is possible to consider that only the operators Q and S are linear, while the factorizations of the plant and controller, and thus G and K are nonlinear.

A.3 Stability of $\{G_S, K_Q\}$ via right factorizations.

$$\begin{aligned}
\begin{bmatrix} M_S & -U_Q \\ -N_S & V_Q \end{bmatrix}^{-1} &= \begin{bmatrix} M+US & -U-MQ \\ -N-VS & V+NQ \end{bmatrix}^{-1} && \text{(A.3.1)} \\
&= \left\{ \begin{bmatrix} M & -U \\ -N & V \end{bmatrix} - \begin{bmatrix} -US & MQ \\ VS & -NQ \end{bmatrix} \right\}^{-1} && \text{(A.3.2)} \\
&= \left\{ \begin{bmatrix} M & -U \\ -N & V \end{bmatrix} - \begin{bmatrix} M & -U \\ -N & V \end{bmatrix} \begin{bmatrix} 0 & Q \\ S & 0 \end{bmatrix} \right\}^{-1} && \text{(A.3.3)} \\
&= \begin{bmatrix} I & -Q \\ -S & I \end{bmatrix}^{-1} \begin{bmatrix} M & -U \\ -N & V \end{bmatrix}^{-1} && \text{(A.3.4)}
\end{aligned}$$

Note that the only point at which linearity was used was in stating

$$\begin{bmatrix} -US & MQ \\ VS & -NQ \end{bmatrix} = \begin{bmatrix} M & -U \\ -N & V \end{bmatrix} \begin{bmatrix} 0 & Q \\ S & 0 \end{bmatrix} \tag{A.3.5}$$

Hence it would seem that if the plant and controller are linear, the operators Q and S may be nonlinear, and the stability result will still hold.

Appendix B

B.1 Differentiating the output of an ANN

As was seen in Chapter 5, it is important to be able to guarantee that it is possible to differentiate the output of an ANN with respect to the input and the parameter vector. In this appendix we detail the procedure for calculating the derivative of a the feed-forward neural network of Section 5.3.1, with sigmoid given by (5.3.3), with respect to the input or parameter vector.

Consider that the ANN has the structure detailed in Section 5.3.1, and has N layers, not including the input layer. Where the map from the inputs x to the outputs y, which is dependent on the parameter, θ is denoted,

$$y = G(\theta, x)$$

Recall the definitions of the inputs and outputs to layer i, r^i and s^i, and the weights matrix θ_i, giving the map from the output of layer $i-1$ to the input to layer i, the precise definitions being given in (5.3.4)-(5.3.8).

The method for calculating the derivative of $G(\theta, x)$ with respect to θ and x is now detailed. Note that as the ANN is defined recursively, with each layer defined in terms of the outputs of the previous layer and the weights, the chain rule may be applied to give the derivative of the output with respect to inputs and weights. Firstly, the derivatives of s^i with respect to s^{i-1} and θ^i are given. Note that the sigmoid of (5.3.3) is used, giving a simple form for the derivative of F_i.

$$\frac{\partial r^i_j}{\partial s^{i-1}_k} = \theta^i_{j,\ k} \qquad \text{(B.1.1)}$$

$$\frac{\partial r^i_j}{\partial \theta^i_{l,\ k}} = \begin{cases} s^{i-1}_k & ,l = j \\ 0 & ,l \neq j \end{cases} \qquad \text{(B.1.2)}$$

$$\frac{\partial s^i_j}{\partial r^i_k} = \begin{cases} s^i_j(1 - s^i_j) & ,j = k \\ 0 & ,j \neq k \end{cases} \tag{B.1.3}$$

Hence the Jacobian of the map $s^{i-1} \mapsto s^i$, denoted J^s_i, is a $n_i \times n_{i-1}$ matrix, dependent on s^i, θ^i and is given by,

$$J^s_i = Diag\{s^i_1(1 - s^i_1),\ s^i_2(1 - s^i_2), \ldots,\ s^i_{n_i}(1 - s^i_{n_i})\}\bar{\theta}^i \tag{B.1.4}$$

where $\bar{\theta}^i$ is the matrix θ^i with the final column deleted. Strictly speaking, this matrix should be $\begin{bmatrix} Diag\{s^i_1(1-s^i_1), \ldots, s^i_{n_i}(1-s^i_{n_i})\} \\ 0 \end{bmatrix} \theta^i$, but since the final node in each layer is constant, it does not appear important to consider this case.

The Jacobian of the map $\theta^i \mapsto s^i$, denoted J^θ_i, is calculated as follows. First note that θ^i must be arranged into a vector. The vector Θ_i is thus constructed as follows,

$$\Theta_i = row(\theta^i) = (\theta^i_{1,1},\ \theta^i_{1,2}, \ldots,\ \theta^i_{1,n_{i-1}+1},\ \theta^i_{2,1}, \ldots,\ \theta^i_{n_i,n_{i-1}+1})$$

Then J^θ_i is a $n_i \times n_i(n_{i-1} + 1)$ matrix, which is constructed by concatenating the n_i, $n_i \times (n_{i-1} + 1)$ matrices, T^i_j.

$$J^\theta_i = \left[T^i_1\ T^i_2\ \ldots\ T^i_{n_i}\right] \tag{B.1.5}$$

The matrix T^i_j is zero everywhere except the j^{th} row, which is given by

$$\left[\ s^i_j(1 - s^i_j)s^{i-1}_1,\ \ s^i_j(1 - s^i_j)s^{i-1}_2,\ \ldots,\ \ s^i_j(1 - s^i_j)s^{i-1}_{n_{i-1}},\ \ s^i_j(1 - s^i_j)\ \right]$$

Thus the derivatives of y with respect to the inputs x and the entire parameter vector $\Theta = (\Theta_1, \Theta_2, \ldots, \Theta_N)$ may be stated as follows:

$$\frac{\partial y}{\partial x} = J^s_N J^s_{N-1} \cdots J^s_2 J^s_1 \tag{B.1.6}$$

$$\frac{\partial y}{\partial \theta} = \left[J^s_N J^s_{N-1} \cdots J^s_2 J^\theta_1 \mid J^s_N J^s_{N-1} \cdots J^s_3 J^\theta_2 \mid \ldots \mid J^s_N J^\theta_{N-1} \mid J^\theta_N\right] \tag{B.1.7}$$

For the particular, 3-layer, architecture which is used in Chapter 5, the function $G(\theta, x)$ is given by,

$$G(\theta, x) = F_2(\theta^2 F^1(\theta^1)) \tag{B.1.8}$$

Hence, using the notation for the derivatives of $G(\theta, x)$ from Section 5.4, the following expressions for the derivatives of an ANN with respect to it's parameter and input vectors are derived.

$$G_1(\theta, x) = \left[J_2^s J_1^\theta \mid J_2^\theta \right] \tag{B.1.9}$$

$$G_2(\theta, x) = J_2^s J_1^s \tag{B.1.10}$$

Note that $G_1(\theta, x)$ and $G_2(\theta, x)$ are functions of both x and θ.

B.2 The Lipschitz constant for an ANN

Following the development in Section 5.4.2, the need for specifying the Lipschitz constant for an ANN with respect to its parameter and input vectors becomes evident. In this section this problem is considered. Recall that a nonlinear function $F : R^n \rightarrow R^m$ is said to be Lipschitz with Lipschitz constant L_F if,

$$\forall x,\, y\, \epsilon R^n,\; \|F(x) - F(y)\| \leq L\|x - y\|. \tag{B.2.1}$$

Where the norms $\|\cdot\|$ are defined on the appropriate spaces.

First note two properties of Lipschitz functions which are important to the problem.

1. If two functions, $F : R^n \rightarrow R^m$, $G : R^m \rightarrow R^l$, are both Lipschitz with Lipschitz constants L_F, L_G, respectively, then the composite function $G \circ F : R^n \rightarrow R^l$ is also Lipschitz, and has Lipschitz constant $L_F L_G$.

2. If we consider the Lipschitz function F as defined above, then the constant L_F puts a bound on the derivative of F in the following sense. Let $J_F(y)$ be the Jacobian of F evaluated at the point y, so that $J_F(y)x$ is the rate of change of the function F at the point y, in the direction x. Then the following inequality holds.

$$\forall\, x,\, y\, \epsilon R^n \;\; \frac{\|J_F(y)x\|}{\|x\|} \;\leq\; L_F \tag{B.2.2}$$

We can view (B.2.2) as giving another definition of the Lipschitz constant for F.

$$L_F = \sup_{y \epsilon R^n,\; \|x\|=1} \|J_F(y)x\| \tag{B.2.3}$$

i.e. we search for the supremum over y of the induced norm of $J_F(y)$.

Consider an ANN as described in Appendix B.1. To find the Lipschitz constants for

$G(\theta, x)$, use the second definition of the Lipschitz constant (B.2.3), and the expressions for the derivatives of the ANN with respect to its parameter and input vectors (B.1.6), (B.1.7). Lipschitz constants are calculated with respect to θ and x, and are denoted L_1 and L_2 respectively.

An expression for L_2 is first calculated. The expression (B.1.7) is regarded as the composition of N linear operators. Note that, since $0 < \sigma(x) < 1$ for all x, it is true that $s^i_j(1 - s^i_j) \leq 1/4$, $\forall i,\ j$. Hence a bound on the Lipschitz constant associated with J^s_i is given by

$$L_{J^s_i} \leq \frac{\|\theta_i\|_2}{4} \tag{B.2.4}$$

Thus the Lipschitz constant, L_2, for the ANN with N layers and weight matrices θ_i, $i = 1, \ldots, N$, is bounded by

$$L_2 = 2^{-2N} \prod_{i=1}^{N} \|\theta_i\| \tag{B.2.5}$$

The equation (B.2.5) represents a bound on the Lipschitz constant for a given ANN. Due to the effects of offsets on the nodes within the net, and the effect of the restriction of the domain of input to one layer due to the range of the previous layer, it is possible that this bound will not be attained. Nevertheless this bound will not be exceeded, so it is the expression for L_2 which shall be used.

In a similar fashion, a bound on L_1 may be obtained, although the expression obtained is more complicated due to the form for $G_2(\theta, x)$. The problem that we need solve is to find L_2 such that

$$\begin{aligned} L_2 &= \max_{x,\theta,\delta} \frac{\|J^\theta_G(\theta, x)\delta\|}{\|\delta\|} \\ &= \max_{x,\theta,\|\delta\|=1} \left\| \left[J^s_N J^s_{N-1} \cdots J^s_2 J^\theta_1 \mid J^s_N J^s_{N-1} \cdots J^s_3 J^\theta_2 \mid \ldots \mid J^s_N J^\theta_{N-1} \mid J^\theta_N \right] \delta \right\| \end{aligned}$$

First note that $J^\theta_G(\theta, x)$ has a special form. Specifically it is constructed by concatenating linear operators. The following lemma may be applied.

Lemma B.1 *Given a matrix M, which is constructed by,*

$$M = [M_1 \mid M_2 \mid \ldots \mid M_n]$$

Then the 2-norm of M is bounded above as follows,

$$\|M\|_2 \leq \left[\sum_{i=1}^{n} \|M_i\|_2^2\right]^{\frac{1}{2}} \tag{B.2.6}$$

□

Proof. Recall the definition of $\|\cdot\|_2$.

$$\|M\|_2 = \sup_x \frac{\|Mx\|}{\|x\|} = \sup_{\|x\|=1} \|Mx\|$$

Consider that the vector x is made up of n sub-vectors, such that x_i corresponds to the input to M_i. Note that,

$$\begin{aligned} \|M_1x_1 + M_2x_2 + \ldots M_nx_n\| &\leq \|M_1x\| + \|M_2x\| + \ldots + \|M_nx\| \\ &\leq \|M_1\|\ \|x\| + \|M_2\|\ \|x\| + \ldots + \|M_n\|\ \|x\| \end{aligned}$$

From the construction of x, it is true that $\|x\|^2 = \sum_{i=1}^{n} \|x_i\|^2$ so that

$$\|M\| \leq \sup_{\sum_i a_i^2 = 1} \sum_{i=1}^{n} a_i \|M_i\|$$

A simple geometric argument shows that this is true for $a_i = \|M_i\| / \left[\sum_{i=1}^{n} \|M_i\|^2\right]^{\frac{1}{2}}$, which gives the result, (B.2.6). ∎

Applying this lemma twice, once to the matrix J_i^θ and once to $J_G^\theta(\theta, x)$ gives us a bound on the Lipschitz constant for the ANN with respect to the parameter vector.

First calculate a bound on the norm of J_i^θ. Recall that J_i^θ is given by (B.1.5). Thus it is only necessary to calculate the norm of the T_j^i, and then apply Lemma B.1. As the T_j^i are zero everywhere, except one row, it is only necessary to consider the effect of this row, which is given by $\left[\ s_j^i(1-s_j^i)s_1^{i-1},\ \ s_j^i(1-s_j^i)s_2^{i-1},\ \ \ldots,\ \ s_j^i(1-s_j^i)s_{n_{i-1}}^{i-1},\ \ s_j^i(1-s_j^i)\ \right]$ For $i \neq 1$ all the s_j^i are between 0 and 1, so that, the maximum norm possible for T_j^i is 1/4, and so the norm for J_i^θ must be less than $(n_{i-1}+1)^{\frac{1}{2}}/4$. For $i = 1$, the situation is somewhat different as $s^0 = (x^T, 1)^T$ which is potentially unbounded. Thus the expression obtained for $\|J_1^\theta\|$ is $\frac{1}{4}\left\|\begin{pmatrix} x \\ 1 \end{pmatrix}\right\| = (\|x\|^2 + 1)^{1/2}/4.$

Applying the same reasoning as lead to a bound on L_2, and Lemma B.1 gives the following bound on L_1.

$$\begin{aligned} L_1 \;\leq\; & \left[2^{-4N}(\|x\|^2+1)\prod_{i=2}^{N}\|\theta^i\|^2 + 2^{-4(N-1)}\,(n_1+1)\prod_{i=3}^{N}\|\theta^i\|^2 + \ldots \right. \\ & \left. +2^{-8}\,(n_{N-2}+1)\,\|\theta^N\|^2 + 2^{-4}\,(n_{N-1}+1)\right]^{\frac{1}{2}} \qquad \text{(B.2.7)} \end{aligned}$$

These results, and the specialization to the case we are interested in, are summarized in the following lemma.

Lemma B.2 *Consider the n layer ANN $G(\theta, x)$ as described in Appendix B.1. Then the Lipschitz constants with respect to the parameter and input vectors, L_1 and L_2 respectively, for $G(\theta, x)$ will not exceed the bounds given by equations (B.2.7) and (B.2.5). For the particular case we are interested in, $G(\theta, x)$ given by (B.1.8), this leads to the expressions,*

$$L_1 \;\leq\; \frac{1}{4}\left[2^{-4}\left(\|x\|^2+1\right)\|\theta^2\|^2 + (n_1+1)\right]^{\frac{1}{2}} \qquad \text{(B.2.8)}$$

$$L_2 \;\leq\; \frac{1}{16}\|\theta_1\|\;\|\theta_2\| \qquad \text{(B.2.9)}$$

□