

# INVERSION OF NONLINEAR AND TIME-VARYING SYSTEMS

*Thomas A. Baran and Alan V. Oppenheim*

Massachusetts Institute of Technology  
Digital Signal Processing Group  
77 Massachusetts Avenue, Cambridge MA 02139  
tbaran@mit.edu, avo@mit.edu

## ABSTRACT

This paper is concerned with the inversion of implementations for systems that may generally be nonlinear and time-varying. Specifically, techniques are presented for modifying an implementation of a forward system, represented as an interconnection of subsystems, in such a way that an implementation for the inverse system is obtained. We focus on a class of modifications that leave subsystems in the inverse system unchanged with respect to those in the forward implementation. The techniques are therefore well-suited to the design of matched pre-emphasis and de-emphasis filters, as approximations due to coefficient quantization in the forward system are naturally matched in the inverse. In performing the inversion, an explicit input-output characterization of the system is not required, although the forward system must be known to be invertible. The techniques are applied to the inversion of nonlinear and time-varying systems, as well as to the problem of sparse matrix inversion.

*Index Terms*— Inverse systems, nonlinear filters, nonlinear systems, signal flow graphs.

## 1. INTRODUCTION

In the design of signal processing algorithms, it is often of interest to implement the inverse of a pre-specified forward system. This can occur, for example, within the context of system equalization, where the forward system may consist of or be modeled by an interconnection of nonlinear and time-varying elements. In other applications where the forward and inverse systems are designed jointly, as for example in applications employing matched pre-emphasis and de-emphasis filters, an important design criterion is that the implementations of the forward and inverse systems are true inverses. It is therefore desirable for approximations made in the implementation of the forward system to be naturally accounted for in the implementation of its inverse.

This paper is concerned with the design of implementations for inverse systems where an implementation of an invertible, generally nonlinear and time-varying forward system has been specified. We specifically focus on techniques for finding implementations of inverse systems that leave certain of the subsystems unchanged with respect to the forward system. As such, the methods developed in this paper are concerned not with the modification of subsystems, but rather with modifying the way in which they are interconnected. The presented techniques do not require an input-output characteri-

zation of the forward system, although the system must be known to be invertible.

We proceed by introducing a form of system representation wherein a system is regarded as a set of subsystems coupled to a linear interconnection. We then present a necessary and sufficient condition under which an alternative interconnection exists such that the resulting overall system is inverted. We make use of the condition in arriving at a graph-based theorem pertinent to interconnections implemented as a specific class of signal flow graphs, and the theorem is applied to the inversion of generally nonlinear and time-varying systems. In inverting systems having interconnections within this class, a forward interconnection composed of unity branch gains implies that the inverse interconnection will be composed of unity branch gains as well, thereby ensuring that any approximations made in the forward system, e.g. coefficient quantization, affect only the subsystems, resulting in an inverse system that is naturally matched to the forward system even in the presence of such approximations. The discussion is focused on discrete-time systems involving linear interconnections that are memoryless and time-invariant, as many of the key issues carry over to arbitrary linear interconnections and continuous-time systems.

## 2. SYSTEM REPRESENTATION

The inversion techniques presented in this paper are facilitated by a form of system representation that takes the behavioral viewpoint, discussed in detail in [1]. We discuss the general problem of system inversion from a behavioral perspective, and we present a form of system representation that will lay the groundwork for development of the presented techniques.

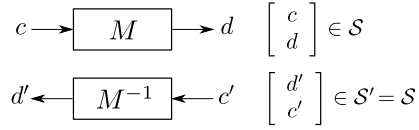
### 2.1. Inversion from a behavioral perspective

There are potentially many different notions of inversion that can be used in developing techniques for inverting systems. Some are discussed in, e.g., [2, 3, 4]. The specific concept of inversion that forms the basis for this paper is related to the idea that a system can be viewed as a map that is representative of constraints between sets of input and output signals, as in [1]. Referring to the forward and inverse systems in Fig. 1, a forward system  $M$  may be regarded as a map from the set of input signals  $\{c[n]\}$  in its domain to the set of output signals  $\{d[n]\}$  in its range. For the purpose of this paper, we will focus on single-input, single-output systems that may be linear or nonlinear, time-invariant or time-varying, and which may or may not contain memory.

We adopt the convention that “the behavior” of a system refers to the entire collection of input-output signal pairs consistent with

---

This work was supported in part by the Texas Instruments Leadership University Program, BAE Systems PO 112991, and Lincoln Laboratory PO 3077828.



**Fig. 1.** A system and its inverse.

the map  $M$ . Formally, we represent the behavior as a set  $\mathcal{S}$  such that

$$\begin{bmatrix} c[n] \\ d[n] \end{bmatrix} \in \mathcal{S}, \quad (1)$$

where  $c[n]$  and  $d[n]$  are signals at the terminal branches connected to the system.

Referring again to Fig. 1, we regard a system  $M$  as invertible if it implements a bijective map from  $\{c[n]\}$  to  $\{d[n]\}$ , i.e. if  $M$  is a one-to-one and onto map from the set of signals  $c[n]$  composing the domain over which the system is defined to the set of signals  $d[n]$  composing the range over which the system is defined. The inverse system  $M^{-1}$  in turn implements the inverse map from the set  $\{d[n]\}$  to  $\{c[n]\}$ , or alternatively from  $\{c'[n]\}$  to  $\{d'[n]\}$ . We conclude that every input-output signal pair corresponding to an invertible system  $M$  also corresponds to its inverse  $M^{-1}$ , in the sense that

$$\begin{bmatrix} c[n] \\ M(c[n]) \end{bmatrix} \in \mathcal{S} \Leftrightarrow \begin{bmatrix} M^{-1}(c'[n]) \\ c'[n] \end{bmatrix} \in \mathcal{S}', \quad (2)$$

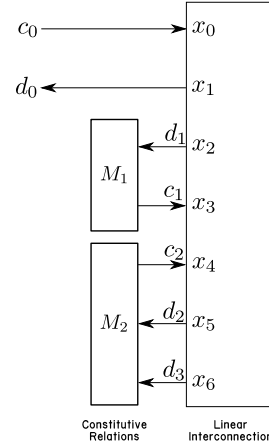
and we write  $\mathcal{S} = \mathcal{S}'$ . An invertible system realized as a map  $M$  from  $\{c[n]\}$  to  $\{d[n]\}$  therefore has the same behavior as its inverse  $M^{-1}$ .

## 2.2. Interconnective system representation

The key emphasis of the paper is on methods for generating implementations of inverse systems from implementations of forward systems, where identical subsystems are used in both. The techniques presented for system inversion therefore focus not on the modification of subsystems but rather on manipulating the way in which they are connected. As such, we discuss a form of system representation that we refer to as *interconnective* and that is designed to separate the behaviors of the subsystems from the relationships that couple them together. In particular, we view a system as having two parts: constitutive relations, e.g. a set of possibly nonlinear and time-varying subsystems that are allowed to have memory, and a linear, time-invariant, memoryless interconnecting system to which the subsystems and overall system input and output are connected.<sup>1</sup> As was mentioned previously, the conditions of memorylessness and time-invariance are introduced to focus the scope of the discussion, and many of the subsequent results generalize naturally to systems involving arbitrary linear interconnections. The interconnective form of system representation is depicted in Fig. 2. As the interconnection is time-invariant and memoryless, we have dropped the explicit dependence on  $n$  in writing the terminal variables.

We are concerned with describing the behavior of the constitutive relations and the behavior of the interconnection first as uncoupled systems, with the intersection of their behaviors being that of the overall system when the two are coupled together. We have

<sup>1</sup>Willems mentions essentially this representation in his work on dissipative systems, e.g. in [5]. It forms the cornerstone of various arguments in this paper, and as such we feel that it is deserving of the special designation.



**Fig. 2.** A signal processing system in an interconnective representation.

specified that the interconnection is linear, and its behavior  $W$  is consequently a vector space.

From an input-output perspective, the interconnection can be represented as a matrix multiplication that maps from the set of vectors of interconnection input terminal variables  $\underline{c} = [c_0, \dots, c_{N_i-1}]^T$  to the set of vectors of interconnection output terminal variables  $\underline{d} = [d_0, \dots, d_{N_o-1}]^T$ , where  $N_i$  and  $N_o$  denote the number of respective input and output terminal branches directed to and from the interconnection in the uncoupled representation. That is, the interconnection is represented by an  $N_o \times N_i$  matrix  $L$  of coefficients where

$$\underline{d} = L\underline{c}. \quad (3)$$

In discussing the behavior of the interconnection, we are interested in the set of all possible values taken on by a vector  $\underline{x} = [x_0, \dots, x_{N-1}]^T$  that contains the  $N = N_i + N_o$  interconnection terminal variables. Referring again to Fig. 2, elements of  $\underline{x}$  will generally correspond to both inputs and outputs, and we introduce a permutation matrix  $P$  that encodes the correspondence between  $\underline{c}$ ,  $\underline{d}$  and  $\underline{x}$ :

$$\underline{x} = P \begin{bmatrix} \underline{c} \\ \underline{d} \end{bmatrix}. \quad (4)$$

A map from the vector of interconnection input variables  $\underline{c}$  to the vector containing the entire set of interconnection terminal variables  $\underline{x}$  may be obtained by combining Eqns. 3-4, resulting in

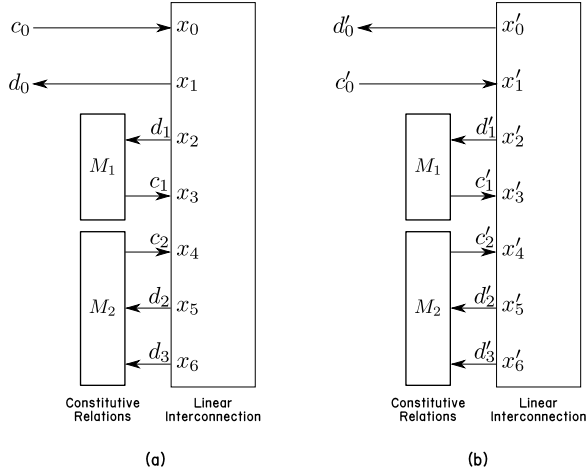
$$\underline{x} = P \begin{bmatrix} I_{N_i} \\ L \end{bmatrix} \underline{c}, \quad (5)$$

where  $I_{N_i}$  is the  $N_i \times N_i$  identity matrix. Consequently the set of allowable vectors  $\underline{x}$ , i.e. the behavior of the interconnection when uncoupled from the constitutive relations, is the vector space

$$W = \text{range} \left\{ P \begin{bmatrix} I_{N_i} \\ L \end{bmatrix} \right\}. \quad (6)$$

## 3. INVERSION TECHNIQUES

Drawing upon the interconnective form of representation, we develop techniques for system inversion that leave the subsystems implementing the constitutive relations unchanged. The general approach is to begin with an invertible system in this representation,



**Fig. 3.** Illustration of the interconnective approach to system inversion. (a) Forward (CF) system. (b) Inverse (CI) system obtained by replacing the LIF in the forward system with an LII.

whose input  $c_0 = x_0$  and output  $d_0 = x_1$  are included in the vector of interconnection terminal variables  $\underline{x}$ . This system, referred to as the coupled forward system (CF), is regarded as a map  $M$  from  $x_0$  to  $x_1$ . The corresponding uncoupled interconnection is referred to as the linear interconnection for the forward system (LIF).

The goal in inverting the system is to determine an alternative interconnection that has the same behavior as the LIF but that has  $c'_0 = x'_1$  as an input and  $d'_0 = x'_0$  as an output, with the directions of all other interconnection terminal branches remaining unchanged. The realization of this interconnection is referred to as the linear interconnection for the inverse system (LII). The coupled system involving the constitutive relations from the CF, coupled to the LII, is referred to as the coupled inverse system (CI).

The CF implements a map from  $c_0 = x_0$  to  $d_0 = x_1$ , and the CI implements a map from  $c'_0 = x'_1$  to  $d'_0 = x'_0$ . As the behavior of the LII is equivalent to the behavior of the LIF, the behavior of the CI is equivalent to that of the CF. The CI consequently implements the map  $M^{-1}$  that is the inverse of the map  $M$  implemented by the CF, and it does so without requiring inversion of any of the subsystems implementing the constitutive relations. The strategy for inversion additionally does not require that the input-output map  $M$  is known explicitly, although  $M$  must be known to be invertible. The general approach is summarized in Fig. 3.

### 3.1. Equivalence of interconnections

One consideration in employing the previously-mentioned approach pertains to the question of whether an appropriate LII exists. We address this issue, presenting a necessary and sufficient condition for the existence of an LII given an LIF.

**Theorem 1.** *Given an LIF where  $x_0 = c_0$  is an interconnection input and  $x_1 = d_0$  is an interconnection output, an LII having the same behavior and having  $x'_1 = c'_0$  as an input and  $x'_0 = d'_0$  as an output exists if and only if the gain in the LIF from  $c_0$  to  $d_0$  is nonzero, i.e. if and only if the LIF matrix  $L$  has the property*

$$L_{1,1} \neq 0. \quad (7)$$

*Proof.* We first show that Eq. 7 is a necessary condition for the existence of an appropriate LII. If  $L_{1,1} = 0$ , i.e. if the gain from  $c_0$  to  $d_0$  is zero, the map realized by the LIF from  $c_0$  to  $d_0$  is many-to-one. Consequently there is no map from  $c'_0$  to  $d'_0$  that has the same behavior, and no appropriate LII exists.

We now show that Eq. 7 is a sufficient condition, i.e. that  $L_{1,1} \neq 0$  implies that there exists an  $L'$  corresponding to an LII having the same behavior as the LIF. Adopting the convention established previously, we denote the number of inputs to the LIF and LII as  $N_i$  and the number of outputs from the LIF and LII as  $N_o$ . The behavior  $W$  of the LIF is given by Eq. 6. Similarly, the behavior  $W'$  of the LII can be written

$$W' = \text{range} \left\{ P' \begin{bmatrix} I_{N_i} \\ L' \end{bmatrix} \right\}, \quad (8)$$

where  $P'$  is the permutation matrix encoding the correspondence between the vector  $\underline{x}'$  containing the entire set of LII terminal variables and the vectors  $\underline{c}'$  and  $\underline{d}'$  respectively containing the LII input and output terminal variables.

We proceed by showing that if  $L_{1,1} \neq 0$ , there exists a full-rank  $N_i \times N_i$  matrix  $A$  such that

$$P' \begin{bmatrix} I_{N_i} \\ L' \end{bmatrix} = P \begin{bmatrix} I_{N_i} \\ L \end{bmatrix} A, \quad (9)$$

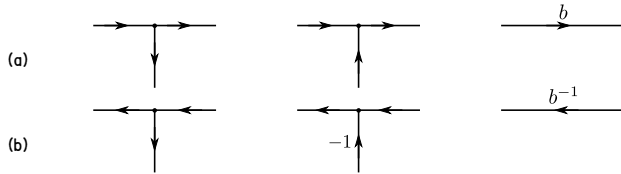
where  $L'$  is the resulting LII matrix. Combining Eqns. 8-9 and 6, we conclude that the existence of such an  $A$  would result in equivalence of the LIF and LII behaviors, i.e.

$$\begin{aligned} W' &= \text{range} \left\{ P \begin{bmatrix} I_{N_i} \\ L \end{bmatrix} A \right\} \\ &= \text{range} \left\{ P \begin{bmatrix} I_{N_i} \\ L \end{bmatrix} \right\} \\ &= W. \end{aligned} \quad (10)$$

It remains to be shown that there exists a full-rank  $N_i \times N_i$  matrix  $A$  as is required for Eq. 10 to hold. We begin by noting that  $P'^{-1}P$  is itself a permutation matrix that as a matrix multiplication swaps elements 1 and  $(N_i + 1)$  of a vector, following from the requirement that  $c'_0 = d_0$  and  $d'_0 = c_0$ . We consequently show that a full-rank  $N_i \times N_i$  matrix  $A$  exists such that the following equation, obtained by multiplying both sides of Eq. 9 by  $P'^{-1}$ , is satisfied:

$$\begin{bmatrix} I_{N_i} \\ L' \end{bmatrix} = \begin{bmatrix} L_{1,1} & L_{1,2} & \cdots & L_{1,N_i} \\ 0 & 1 & & \\ \vdots & & \ddots & \\ 0 & & & 1 \\ 1 & 0 & \cdots & 0 \\ L_{2,1} & L_{2,2} & \cdots & L_{2,N_i} \\ \vdots & \vdots & \ddots & \vdots \\ L_{N_o,1} & L_{N_o,2} & \cdots & L_{N_o,N_i} \end{bmatrix} A. \quad (11)$$

Under the condition that  $L_{1,1} \neq 0$ , standard column reduction operations can be used to transform the upper partition in the matrix on the right hand side of Eq. 11 to  $I_{N_i}$  as desired.  $A$  is selected to encode these operations, and since the standard column reduction operations are invertible,  $A$  is full-rank. Consequently Eq. 10 holds, completing the proof.  $\square$



**Fig. 4.** Elements pertinent to Theorem 2 composing (a) the mentioned path from  $c_0$  to  $d_0$  and (b) the path from  $c'_0$  to  $d'_0$ .

### 3.2. Obtaining an LII flow graph from an LIF

With the conditions under which a desired LII exists now in place, we discuss the issue of generating a signal flow graph for an LII given a signal flow graph for an LIF. The entries of the LIF interconnection matrix  $L$  may generally take on any scalar value, and as such the branches in a flow graph implementation for the LIF may contain non-unity branch gains. The following theorem, which applies to a class of LIF flow graphs that occur in common signal processing structures, presents a method for generating a flow graph for the LII, drawing a correspondence between the branch gains in the two implementations.

**Theorem 2.** *The theorem applies to a signal flow graph for an LIF consisting of a single path from the input  $c_0 = x_0$  to the output  $d_0 = x_1$  where the interconnection inputs  $c_1, \dots, c_{N_i-1}$  form incident branches directed toward the path and the interconnection outputs  $d_1, \dots, d_{N_o-1}$  form incident branches directed away from the path, and where each branch gain along the path is nonzero. Given such a flow graph for an LIF, a flow graph for the LII can be realized by reversing the direction of all of the branches forming the path, inverting each of the branch gains along the path, and negating any incident branches that are directed toward the path, resulting in a path from  $c'_0 = x'_1$  to  $d'_0 = x'_0$ . The directions of the incident branches remain unchanged.*

*Proof.* We begin by noting that the condition for the existence of an LII as stated in Theorem 1 is satisfied by the requirement that all gains along the path from  $c_0$  to  $d_0$  are nonzero. One strategy in completing the proof involves representing the signal flow graph in terms of its matrix  $L$  as in Eq. 3 and showing that the prescribed operations result in a new matrix  $L'$  corresponding to an interconnection that has the same behavior. As the proof becomes somewhat lengthy using this approach, we instead opt to present the proof of Theorem 2 using a behavioral argument.

Consider a flow graph of the mentioned form that implements the LIF. The process of reversing the branches along the path from  $c_0$  to  $d_0$  modifies the interconnection to be in the input-output configuration desired of the LII. It remains to be shown that the resulting flow graph has the same behavior as that of the LIF. In illustrating this, we view the signal flow graph as being composed of the elements indicated in Fig. 4(a), coupled together by equality constraints between the branch variables. Fig. 4(b) illustrates the corresponding elements in the reversed path. Each of the elements in Fig. 4(b) constrains its terminal variables in the same way as the corresponding element in Fig. 4(a), and the elements in each pair consequently have the same behavior. The connected elements forming the LII therefore have the same behavior as the connected elements forming the LIF.  $\square$

Given an LIF flow graph of the form required by Theorem 2 whose branch gains along the mentioned path are unity, the corre-

sponding gains in the LII graph will be unity as well. Approximations such as coefficient quantization that are made in implementing the CF will therefore be manifest only in the CF constitutive relations, which are identical to those in the CI. The CF and CI systems will in this case be naturally matched, even in the presence of such approximations.

## 4. APPLICATIONS

In this section, we apply Theorem 2 in the development of inversion techniques that can be directly applied to systems represented as signal flow graphs. The methods are illustrated through examples involving known inverse systems and are used to arrive at an efficient algorithm for computing inverse maps of linear operators for which the forward map, represented as a matrix multiplication, is sparse.

### 4.1. A graph-based inversion technique

Direct application of Theorem 2 to a system in interconnective form may result in delay-free loops that pass through the constitutive relations, complicating implementation of the inverse system. While the technique in, e.g., [6] may be used in this situation, it would be desirable to avoid delay-free loops altogether. We present a corollary to Theorem 2 that addresses this concern, facilitating the inversion of systems realized as signal flow graphs, without requiring representation in an interconnective form.

**Corollary 1.** *The corollary applies to a system realized as an invertible single-input, single-output signal flow graph having a linear, time-invariant, memoryless path from the input  $c_0$  to the output  $d_0$  that is the only (linear or nonlinear) memoryless path from  $c_0$  to  $d_0$ , and along which each branch gain is nonzero. Given such a flow graph, a flow graph for the inverse system can be realized by reversing the direction of all of the branches forming the path, inverting each of the branch gains along the path, and negating any incident branches that are directed toward the path. The inverse system will not contain any delay-free loops.*

*Proof.* Viewing the flow graph as being in an interconnective form, where the LIF contains the mentioned path exclusively, and where the remainder of the flow graph comprises the constitutive relations, the proof follows from Theorem 2. The indicated modifications result in an LII having the same behavior as the LIF and also having the desired input-output configuration. The CI is therefore the inverse of the CF. The presence of a single memoryless path in a signal flow graph implies that reversal of the path will not result in the introduction of any delay-free loops.  $\square$

### 4.2. Inversion of time-varying linear filters

Fig. 5 illustrates the application of Corollary 1 in the inversion of a time-varying linear filter, realized in a direct form structure. The values for the time-varying coefficients  $b_k[n]$  are assumed to result in a map from  $c[n]$  to  $d[n]$  that is invertible in the sense discussed in Section 2.1. Note that this does not imply that the inverse system will necessarily be causally invertible. For example in the time-invariant case where the coefficients  $b_k[n] = b_k$  are constants, the values of  $b_k$  can be chosen so that the forward system is not minimum phase but is still invertible, although the stable inverse will be noncausal. Alternatively, a causal implementation of the inverse system will be unstable.

Another consideration regarding the implementation of the inverse system relates to the issue of choosing initial conditions. As



