[10] F. Giri, J. M. Dion, M. M'Saad, and L. Dugard, "A globally convergent pole placement indirect adaptive controller," in *Proc. 1987 Conf. Decision Contr.*, vol. 1, Los Angeles, CA, 1987, pp. 1-6.
[11] ——, "Robust pole placement direct adaptive control," in *Proc. 1987 Conf. Decision Contr.*, vol. 1, Los Angeles, CA, 1987, pp. 1-6.
[12] J. W. Polderman, "A state space approach to the problem of adaptive pole placement assignment," *Math Contr. Signals. Syst.*, vol. 2, pp. 71-94, 1989.
[13] E. W. Bai and S. Sastry, "Global stability proofs for continuous-time indirect adaptive control schemes," *IEEE Trans. Automat. Contr.*, vol. AC-32, no. 6, pp. 537-543, 1987.
[14] K. S. Narendra and A. M. Annaswamy, *Stable Adaptive Systems.* Englewood Cliffs, NJ: Prentice-Hall, 1989.

A System-Theoretic Appropriate Realization of the Empty Matrix Concept

C. N. Nett and W. M. Haddad

Abstract—**Inthis note we propose an algebraic realization of the empty matrix concept which is appropriate for system-theoretic applications. This realization differs considerably from the realization currently implemented by the MathWorks, Inc. within their MATLAB program. We demonstrate by repeated example the utility of our realization of the empty matrix concept, and through these same examples indicate the deficiencies of the current MATLAB realization of this concept. These examples fully delineate how the empty matrix concept can be utilized to transparently handle static and / or single vector input, single vector output systems within the more general context of dynamic, two vector input, two vector output systems.**

I. Introduction

For many system-theoretic matrix formulas there exist special cases for which one or more of the matrices involved in the formula do not exist, and hence the formula is inappropriate. One illustration of this point can be given by considering the formula for the transfer matrix of a continuous-time, lumped, linear, time invariant, state-space system:

$$G(s) = C(sI - A)^{-1}B + D. \tag{1}$$

This formula involves the matrices A, B, C, and D. However, A, B, and C do not exist in the special case where the underlying state-space system is static, for in this case the system has no states. Correspondingly, in this special case the formula given above is clearly inappropriate.

In special cases of the type described above, one can often simply note the special case, and then invoke an alternative formula. Indeed, in the above example, one can simply note the special case of a static system, and then invoke the alternative formula $G(s) = D$.

An elegant alternative to the procedure described above can be advanced by introducing the concept of an empty matrix. Intuitively, an empty matrix has no entries, and hence can be substituted in a matrix formula for any matrix which does not exist. An algebraic realization of the empty matrix concept is specified to indicate how the usual rules of matrix addition, matrix multiplication, etc., are extended to apply to empty matrices. If this algebraic realization is chosen appropriately, the matrix formula can be manipulated, using the algebraic properties of empty matrices, to arrive at the desired result without ever having to explicitly note the special case or invoke an alternative formula.

The elegance afforded by the empty matrix approach can potentially be exploited to great advantage in the context of large-scale software development. Indeed, the traditional approach to special cases one must include in the software explicit checks for special cases along with appropriate provisions for each special case. This would be done for each matrix formula implemented in the software with potential for special cases of the type described above. Using the empty matrix approach, one would simply introduce the empty matrix as an object, or data type, and then code its algebraic realization. This would be done only once. Subsequent code would then be devoid of cumbersome explicit checks and provisions for special cases of the type described above.

A. Previous Work

The authors were first exposed to the empty matrix concept through their use of the **MATLAB** program developed by The MathWorks, Inc. [1]. The empty matrix was first introduced into **MATLAB** several years ago. A specific realization of the empty matrix concept is implemented in the current version of **MATLAB** (Version 3.5). Quoting from [1]:

> "We're not sure we've done it correctly, or even consistently, but we have found the idea useful."

Our opinion is that the current **MATLAB** realization of the empty matrix concept is neither correct, consistent, or useful, at least not for system-theoretic applications. To give some indication of why we have formed this opinion, consider once again the matrix formula (1). Let [ ] denote an empty matrix. Substituting $A = [\ ]$, $B = [\ ]$, and $C = [\ ]$ into (1) and using the algebraic realization of the empty matrix concept currently implemented within **MATLAB**, one obtains $G(s) = [\ ]$, as opposed to the desired result $G(s) = D$. A more fundamental illustration of the deficiencies inherent in the current **MATLAB** realization of the empty matrix concept can be given by considering the matrix formula below:

$$\Delta = UN + VD. \tag{2}$$

Since we may write

$$UN + VD = [U \quad V]\begin{bmatrix} N \\ D \end{bmatrix}$$

in the case where $V = [\ ]$ and $D = [\ ]$, one desires that $\Delta = UN$. However, upon substituting $V = [\ ]$ and $D = [\ ]$ in (2) and using the algebraic realization of the empty matrix concept currently implemented within **MATLAB**, one obtains $\Delta = [\ ]$. We could continue here, and indeed many other deficiencies inherent in the current **MATLAB** realization of the empty matrix concept are expounded upon in the sequel. The above examples should, however, provide ample evidence to support the opinion put forth above.

Manuscript received July 12, 1992; revised April 17, 1992. The work of the first author was supported by the NSF and ONR. The work of the second author was supported by the AFOSR.
C. N. Nett is with the Laboratory for Identification and Control of Complex Highly Uncertain Systems (LICCHUS), School of Aerospace Engineering, Georgia Institute of Technology, Atlanta, GA 30332-0150.
W. M. Haddad is with the Department of Mechanical and Aerospace Engineering, Florida Institute of Technology, Melbourne, FL 32901.
IEEE Log Number 9204989.

0018-9286/93$03.00

As far as other previous work on the empty matrix concept is concerned, we once again quote from [1]:

> "As far as we know, the literature on the algebra of empty matrices is itself empty."

After a thorough search of the literature, we have found the above assessment to be incorrect. Indeed, a concept very similar to the empty matrix concept has been considered in the literature. This concept is known as the "null string," and is employed in the APL programming language (see, for example, [3]). Clearly then, the literature on the empty matrix prior to this note is not comprised solely of [1], contrary to previous thought.

B. Contributions Summary

In this note we propose an algebraic realization of the empty matrix concept which differs considerably from the current **MATLAB** realization of the empty matrix concept. This new realization of the empty matrix concept is believed to be most appropriate for system-theoretic applications. Indeed, with this realization of the empty matrix concept one obtains $G(s) = D$ upon substituting $A = [\ ]$, $B = [\ ]$, and $C = [\ ]$ into (1), and also obtains $\Delta = UN$ upon substituting $V = [\ ]$ and $D = [\ ]$ into (2), as desired. We demonstrate by repeated examples such as these the utility of our realization of the empty matrix concept, and through these same examples indicate the deficiencies of the current **MATLAB** realization of this concept. Additionally, the examples are used to show how the empty matrix concept can be utilized to transparently handle static and/or single vector input, single vector output (SVISVO) systems within the more general context of dynamic, two vector input, two vector output (TVITVO) systems.

C. Addendum

After this work was completed [2], [8] the results were communicated to Dr. Jie Chen. Some time thereafter, Dr. Chen discovered that a realization of the empty matrix concept similar to the one advanced here had already been given in a book by Stoer and Witzgall [10]. Several months after Dr. Chen's discovery, Professor Carl deBoor called the authors attention to a recent paper of his [5], wherein yet another realization of the empty matrix concept similar to the one advanced here had been given. Though these three similar realizations of the empty matrix concept were all independently derived, it is clear that the work of Stoer and Witzgall [10] predates both the current work [2], [8] and the work of deBoor [5]. It should be noted, however, that the development given in [10] is quite brief, occupying less than two paragraphs. More importantly, perhaps, no attempt is made in either [10] or [5] to delineate the system-theoretic utility of the empty matrix concept, and this is a key thrust of the current note.

D. Outline

An outline of this note is as follows: In Section II we review the current **MATLAB** realization of the empty matrix concept and then present a new realization of this concept. In Section III we show how the empty matrix concept can be utilized to transparently handle static and/or SVISVO systems within the more general context of dynamic TVITVO systems. Finally, concluding remarks are given in Section IV.

II. Algebraic Realization of the Empty Matrix Concept

In this section, we propose an algebraic realization of the empty matrix concept which is believed to be most appropriate for system-theoretic applications. We begin by reviewing the current **MATLAB** realization of the empty matrix concept. This realization may be summarized as follows:

A1) Matrices X of dimension $p \times m$, with p and m both strictly positive integers (nonzero), or $p = m = 0$, are considered. The matrix with dimensions 0×0 is said to be an empty matrix, and denoted $[\ ]$.

A2) The matrix $[\ ]$ can be multiplied by scalars and added to or multiplied by matrices of arbitrary dimensions. Furthermore, doing so propagates the matrix $[\ ]$. Specifically, for any scalar c and matrix X:

$$c \cdot [\ ] = [\ ] \cdot c = [\ ],$$
$$X + [\ ] = [\ ] + X = [\ ],$$
$$X \cdot [\ ] = [\ ] \cdot X = [\ ].$$

A3) With the above definitions, the usual associative and distributive laws hold for scalar multiplication, matrix addition, and matrix multiplication.

A4) With the above definitions, $[\ ] = 0_{0\times 0} = I_{0\times 0} = ([\ ])^{-1}$.

The new realization of the empty matrix proposed here may be summarized as follows:

B1) Matrices $X_{p\times m}$ of dimension $p \times m$, with p and m nonnegative integers (possibly zero), are considered. A matrix with one or both dimensions zero is said to be an empty matrix, and denoted $[\ ]$. The dimensions of an empty matrix may be explicitly denoted through the use of the notation $[\ ]_{p\times m}$, $[\ ]_{0\times m}$, $[\ ]_{p\times 0}$, $[\ ]_{0\times 0}$. In the case of $[\ ]_{p\times m}$ it is implicit that one or both of p or m is zero. We regard $[\ ]_{0\times m}$, $[\ ]_{p\times 0}$, and $[\ ]_{0\times 0}$ as defining fat, tall, and square empty matrices, respectively.

B2) Matrices $[\ ]$ can be multiplied by scalars and added to or multiplied by matrices of compatible dimensions only. Here compatible dimensions are defined in the usual sense. The dimensions of the resulting sum or product are determined by the usual rules. As such, additions and multiplications involving empty matrices propagate empty matrices in all but one case. Specifically, for any scalar c and $m \times p$ matrix $X_{m\times p}$:

$$c \cdot [\ ]_{p\times m} = [\ ]_{p\times m} \cdot c = [\ ]_{p\times m}, \tag{3}$$

$$[\ ]_{p\times m} + [\ ]_{p\times m} = [\ ]_{p\times m}, \tag{4}$$

$$[\ ]_{0\times m} \cdot X_{m\times p} = [\ ]_{0\times p}, \tag{5}$$

$$X_{m\times p} \cdot [\ ]_{p\times 0} = [\ ]_{m\times 0}, \tag{6}$$

$$[\ ]_{p\times 0} \cdot [\ ]_{0\times m} = 0_{p\times m}. \tag{7}$$

B3) With the above definitions, the usual associative and distributive laws hold for scalar multiplication, matrix addition, and matrix multiplication. In fact, (7) has been dictated by our desire for matrix multiplication to distribute over addition. This implication is readily seen by writing $0_{p\times m} = [\ ]_{p\times 0} \cdot [\ ]_{0\times m} - [\ ]_{p\times 0} \cdot [\ ]_{0\times m} = [\ ]_{p\times 0}([\ ]_{0\times m} - [\ ]_{0\times m}) = [\ ]_{p\times 0} \cdot [\ ]_{0\times m}$.

B4) With the above definitions, $[\ ]_{p\times m} = 0_{p\times m}$, $[\ ]_{0\times 0} = I_{0\times 0}$, $([\ ]_{0\times 0})^{-1} = [\ ]_{0\times 0}$.

As can be seen from the above, the realization of the empty matrix concept proposed here differs considerably from the one currently implemented in **MATLAB**. Note especially that the current **MATLAB** realization considers only the square empty matrix, yet allows addition and multiplication of this matrix with other matrices of arbitrary dimensions. The realization proposed here considers both square and nonsquare empty matrices, and addition and multiplication are allowed only between matrices of compatible dimensions.

Using the above summaries, we can revisit the examples considered in Section I and verify the results which were stated there without proof. For (1), we make the substitution $A = [\ ]$, $B = [\ ]$, and $C = [\ ]$. In the case of the current **MATLAB** realization, each of these empty matrices is necessarily square, and one obtains:

$$\begin{aligned} G(s) &= C(sI - A)^{-1}B + D \\ &= [\ ]\cdot(sI - [\ ])^{-1}\cdot[\ ] + D \\ &= [\ ]\cdot([\ ])^{-1}\cdot[\ ] + D \\ &= [\ ]\cdot[\ ]\cdot[\ ] + D \\ &= [\ ] + D \\ &= [\ ]. \end{aligned}$$

In the case of our realization, each of the empty matrices is necessarily appropriately dimensioned, and one obtains:

$$\begin{aligned} G(s) &= C(sI - A)^{-1}B + D \\ &= [\ ]_{p\times 0}\cdot(sI_{0\times 0} - [\ ]_{0\times 0})^{-1}\cdot[\ ]_{0\times m} + D_{p\times m} \\ &= [\ ]_{p\times 0}\cdot(s\cdot[\ ]_{0\times 0} - [\ ]_{0\times 0})^{-1}\cdot[\ ]_{0\times m} + D_{p\times m} \\ &= [\ ]_{p\times 0}\cdot([\ ]_{0\times 0} - [\ ]_{0\times 0})^{-1}\cdot[\ ]_{0\times m} + D_{p\times m} \\ &= [\ ]_{p\times 0}\cdot([\ ]_{0\times 0})^{-1}\cdot[\ ]_{0\times m} + D_{p\times m} \\ &= [\ ]_{p\times 0}\cdot[\ ]_{0\times 0}\cdot[\ ]_{0\times m} + D_{p\times m} \\ &= [\ ]_{p\times 0}\cdot[\ ]_{0\times m} + D_{p\times m} \\ &= 0_{p\times m} + D_{p\times m} \\ &= D_{p\times m}. \end{aligned}$$

For (2), we make the substitution $V = [\ ]$ and $D = [\ ]$. In the case of the current **MATLAB** realization, each of these empty matrices is again necessarily square, and one obtains:

$$\begin{aligned} M &= UN + VD \\ &= UN + [\ ]\cdot[\ ] \\ &= UN + [\ ] \\ &= [\ ]. \end{aligned}$$

In the case of our realization, each of the empty matrices is again necessarily appropriately dimensioned, and one obtains:

$$\begin{aligned} M &= U_{p\times n}\cdot N_{n\times m} + [\ ]_{p\times 0}\cdot[\ ]_{0\times m} \\ &= U_{p\times n}\cdot N_{n\times m} + 0_{p\times m} \\ &= U_{p\times n}\cdot N_{n\times m}. \end{aligned}$$

This analysis validates the results stated without proof in the introduction. More to the point, it makes clear both the basic flaws underlying the current **MATLAB** realization of the empty matrix concept, and the consistency of the new realization of the empty matrix concept proposed here.

It should be clear to the reader from B1)–B4) that all the usual algebraic properties for matrices hold for the algebraic realization of the empty matrix concept given here. It follows that one need not distinguish between empty and nonempty matrices in purely algebraic matrix formulas if this realization is adopted, as consistent results will be obtained in any case; to wit, substituting appropriately dimensioned empty matrices in a purely algebraic matrix formula for matrices which do not exist will automatically yield the corresponding special case of the formula. This property was clearly demonstrated in the examples discussed above.

At this point we call attention to the fact that we have purposely avoided any and all discussion of anything but the purely algebraic properties of empty matrices. For example, we have purposely not discussed the determinant, eigenvalues, singular values, or norm of an empty matrix. The reason for this is because we feel strongly that the empty matrix concept has utility only from a purely algebraic standpoint. Note, however, that determinants, eigenvalues, singular values, and norms of empty matrices are considered in the current version of **MATLAB**.

From this point onward in this note we will deal only with the new realization of the empty matrix concept as summarized in B1)–B4). Furthermore, any indicated matrix expression will be tacitly assumed to involve matrices of compatible dimensions, and any indicated matrix inverse will be tacitly assumed to exist.

III. Use with TVITVO Systems

In this section, we show how the empty matrix concept can be utilized to transparently handle the special cases of static and/or SVISVO systems within the more general context of dynamic TVITVO systems. In addition, we show that as a consequence of the use of empty matrices, interconnection of two TVITVO systems may be regarded as a completely general form of system interconnection which encompasses the usual parallel, cascade, and feedback interconnections, and also the linear fractional transformation, as special cases. We begin by considering the TVITVO system depicted in Fig. 1. Here T is a partitioned proper real-rational transfer matrix:

$$T := \begin{bmatrix} T_{11} & T_{12} \\ T_{21} & T_{22} \end{bmatrix} \tag{8}$$

to which we associate a partitioned standard state-space realization:

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix} := \left[\begin{array}{c|cc} A & B_1 & B_2 \\ \hline C_1 & D_{11} & D_{12} \\ C_2 & D_{21} & D_{22} \end{array}\right] \tag{9}$$

denoting the correspondence as follows:

$$T \rightsquigarrow \begin{bmatrix} A & B \\ C & D \end{bmatrix}.$$

To show how the TVITVO system representation can be made completely general by allowing the use of empty matrices in (8) and (9), consider first the case of a static systems T_{static}. Using the empty matrix, a state-space realization of T_{static} can be written as follows:

$$T_{\text{static}} \rightsquigarrow \begin{bmatrix} [\ ] & [\ ] \\ [\ ] & T_{\text{static}} \end{bmatrix}. \tag{10}$$

Note here that while the empty matrices in (10) can be replaced by zero matrices without invalidating (10), to do so adds uncontrollable and unobservable unstable modes to the realization, which is clearly undesirable.

Consider now the case of a SVISVO system T_{svisvo}. The properties of the empty matrix can be used to write:

$$T_{\text{svisvo}} = \begin{bmatrix} T_{\text{svisvo}} & [\ ] \\ [\ ] & [\ ] \end{bmatrix}. \tag{11}$$

This construction corresponds to setting the second input and second output null in Fig. 1. Of course, three other distinct constructions are possible, corresponding to the three remaining choices for null input–output pairs in Fig. 1.

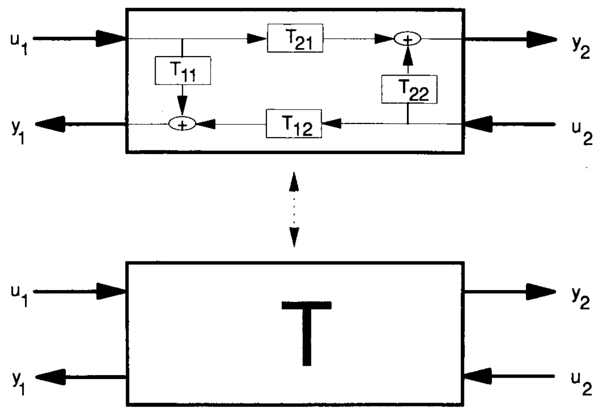

Fig. 1. TVITVO system representation.

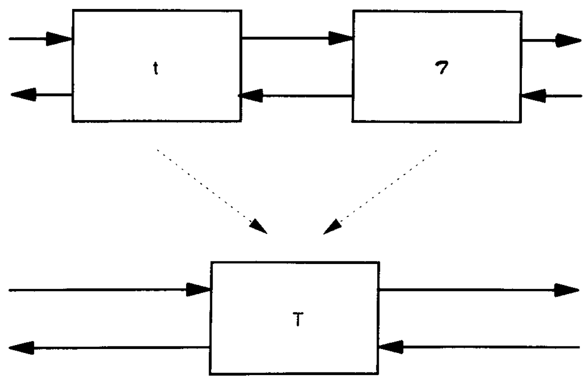

Fig. 2. Interconnection of two TVITVO systems.

The above constructions show how static and/or SVISVO systems can be transparently represented as TVITVO systems using the empty matrix concept. Another important consequence of using the empty matrix concept in the context of TVITVO systems is that the interconnection of two TVITVO systems may be regarded as a completely general form of system interconnection. To show this we begin by considering two TVITVO systems

$$t \rightsquigarrow \begin{bmatrix} a & b \\ c & d \end{bmatrix}, \quad \mathcal{T} \rightsquigarrow \begin{bmatrix} \mathcal{A} & \mathcal{B} \\ \mathcal{C} & \mathcal{D} \end{bmatrix}.$$

Assuming that the dimensions of t_{22} and the transpose of $\mathcal{T}_{11}$ are identical, we may interconnect t and $\mathcal{T}$ as indicated in Fig. 2 to yield yet another TVITVO system, which we denote by T. Recalling the top half of Fig. 1, it should be apparent to the reader that standard parallel interconnection, cascade interconnection, and feedback interconnection are all special cases of the TVITVO interconnection depicted in Fig. 2. Indeed, with

$$t = \begin{bmatrix} M & I \\ I & 0 \end{bmatrix}, \quad \mathcal{T} = \begin{bmatrix} N & [\ ] \\ [\ ] & [\ ] \end{bmatrix}, \tag{12}$$

the interconnection depicted in Fig. 2 describes the standard parallel interconnection of the two systems M and N. Similarly, standard cascade interconnection of M and N is described by Fig. 2 when

$$t = \begin{bmatrix} 0 & I \\ M & 0 \end{bmatrix}, \quad \mathcal{T} = \begin{bmatrix} N & [\ ] \\ [\ ] & [\ ] \end{bmatrix}, \tag{13}$$

and standard feedback interconnection of M and N is described by Fig. 2 when

$$t = \begin{bmatrix} 0 & I \\ I & M \end{bmatrix}, \quad \mathcal{T} = \begin{bmatrix} N & [\ ] \\ [\ ] & [\ ] \end{bmatrix}. \tag{14}$$

Another important form of system interconnection is the the general interconnection of a TVITVO system with a SVISVO system as depicted in Fig. 3. Through the use of empty matrices, this interconnection can also be transparently regarded as a special case of the TVITVO interconnection. Indeed, substituting

$$\mathcal{T} = \begin{bmatrix} \mathcal{T}_{11} & [\ ] \\ [\ ] & [\ ] \end{bmatrix}, \tag{15}$$

in the interconnection depicted in Fig. 2 results in the interconnection depicted in Fig. 3.

The interconnections represented in Fig. 2 and in Fig. 3 correspond, respectively, to certain mathematical operations know as the star-product and the linear fractional transformation [9]. These operations are defined as follows

Definition 3.1: Consider the two TVITVO systems t and $\mathcal{T}$. The star product of $\mathcal{T}$ with t, denoted $t * \mathcal{T}$, is defined as the TVITVO system T with:

$$T_{11} = t_{11} + t_{12}\mathcal{T}_{11}(I - t_{22}\mathcal{T}_{11})^{-1}t_{21}, \tag{16}$$

$$T_{12} = t_{12}(I - \mathcal{T}_{11}t_{22})^{-1}\mathcal{T}_{12}, \tag{17}$$

$$T_{21} = \mathcal{T}_{21}(I - t_{22}\mathcal{T}_{11})^{-1}t_{21}, \tag{18}$$

$$T_{22} = \mathcal{T}_{22} + \mathcal{T}_{21}t_{22}(I - \mathcal{T}_{11}t_{22})^{-1}\mathcal{T}_{12}. \tag{19}$$

Definition 3.2: Consider the TVITVO system t and the SVISVO system $\mathcal{T}_{11}$. The linear fractional transformation of $\mathcal{T}_{11}$ under t, denoted $t\diamond\mathcal{T}_{11}$, is defined as the SVISVO system T_{11} given by:

$$T_{11} = t_{11} + t_{12}\mathcal{T}_{11}(I - t_{22}\mathcal{T}_{11})^{-1}t_{21}.$$

Implicit in the above definitions for systems (i.e., proper real rational transfer matrices) are corresponding definitions for appropriately partitioned constant (i.e., real/complex) matrices. As such, we may view the star product and linear fractional transformation on systems as, respectively, pointwise star products and linear fractional transformations on system frequency evaluations.

The reader can readily verify that upon substituting (15) into the expression $t * \mathcal{T}$, one obtains $t\diamond\mathcal{T}_{11}$. This makes very explicit the fact that the interconnection depicted in Fig. 3 can be viewed as a special case of the interconnection depicted in Fig. 2 through the use of empty matrices.

A state-space formula for TVITVO interconnection (the star product) can be derived via a series of straightforward but tedious algebraic manipulations [6].

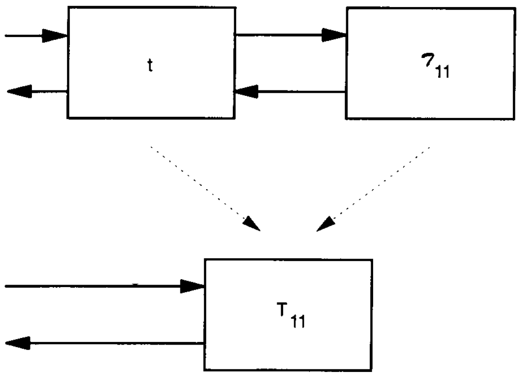

Fig. 3. Interconnection of a TVITVO system with a SVISVO system.

Fact 3.1: Consider the three TVITVO systems t, $\mathcal{T}$, and $T = t * \mathcal{T}$. Denote the state-vectors corresponding to given realizations of t and $\mathcal{T}$ by x_t and $x_{\mathcal{T}}$, respectively. Under these conditions

$$A = \begin{bmatrix} a & b_2 \\ c_2 & d_{22} \end{bmatrix} * \begin{bmatrix} \mathcal{D}_{11} & \mathcal{C}_1 \\ \mathcal{B}_1 & \mathcal{A} \end{bmatrix}, \tag{20}$$

$$B = \begin{bmatrix} b_1 & b_2 \\ d_{21} & d_{22} \end{bmatrix} * \begin{bmatrix} \mathcal{D}_{11} & \mathcal{D}_{12} \\ \mathcal{B}_1 & \mathcal{B}_2 \end{bmatrix}, \tag{21}$$

$$C = \begin{bmatrix} c_1 & d_{12} \\ c_2 & d_{22} \end{bmatrix} * \begin{bmatrix} \mathcal{D}_{11} & \mathcal{C}_1 \\ \mathcal{D}_{21} & \mathcal{C}_2 \end{bmatrix}, \tag{22}$$

$$D = d * \mathcal{D}, \tag{23}$$

and the state-vector x_T defined by the above realization of T is the composite of x_t and $x_{\mathcal{T}}$:

$$x_T = \begin{bmatrix} x_t \\ x_{\mathcal{T}} \end{bmatrix}.$$

A state-space formula for the linear fractional transformation follows immediately from the above result for the star product via the substitution:

$$\mathcal{T} = \begin{bmatrix} \mathcal{T}_{11} & [\,] \\ [\,] & [\,] \end{bmatrix} \rightsquigarrow \left[\begin{array}{c|cc} \mathcal{A} & \mathcal{B}_1 & [\,] \\ \hline \mathcal{C}_1 & \mathcal{D}_{11} & [\,] \\ [\,] & [\,] & [\,] \end{array}\right],$$

assuming that

$$\mathcal{T}_{11} \rightsquigarrow \begin{bmatrix} \mathcal{A} & \mathcal{B}_1 \\ \mathcal{C}_1 & \mathcal{D}_{11} \end{bmatrix}.$$

We omit the resulting formula to underscore the point that it can be transparently obtained from the corresponding formula for the star product via a simple substitution of empty matrices.

The state-space formula for TVITVO interconnection given in Fact 3.1 is particularly convenient for numerical implementation. Indeed, to implement (20)–(23) in **MATLAB** we would first implement the constant matrix version of (16)–(19) as the m-file **star_frq**, which could of course be called on a frequency-by-frequency basis to compute the transfer matrix version of (16)–(19). Next, we would implement (20)–(23) as the m-file **star_ss**. Here **star_ss** would simply call **star_frq** four times, with appropriate arguments. The resulting implementation of (20)–(23) would be comprised of only two **MATLAB** m-files, and each m-file would contain only a few lines of code.

Recalling that we have shown that TVITVO interconnection can be viewed as a completely general form of system interconnection if empty matrices are allowed, we close by noting that the elegant implementation of TVITVO interconnection outlined above could in turn be called by other m-files, with arguments as indicated in (12)–(14), to generate more standard system interconnections. Continuing in this vein one could in principle implement a completely general, yet extremely compact computational facility for system interconnection, with only a modest effort. Consequently, we would expect to soon see the TVITVO system representation and the star product incorporated within existing CACSD packages such as [7], [4].

IV. Concluding Remarks

This note has shown that the empty matrix concept has potential to provide an elegant, general means for transparently handling special cases in matrix formulas corresponding to the absence of one or more of the matrices involved in the formula. It has also been pointed out that this potential can be realized only if an appropriate algebraic realization of this concept is adopted. One such realization is believed to have been given in this paper. It is our hope that this realization of the empty matrix will soon be implemented within software packages such as **MATLAB**.

Acknowledgment

The authors would like to acknowledge several helpful discussions with D. Bernstein, J. Chen, M. Fan, D. Minto, and B. Morton concerning the subject of this note. The authors would also like to thank J. Little and C. deBoor—J. Little did us a service by passing along a copy of this note to C. deBoor, and C. deBoor did us a service by providing us with a copy of [5].

References

[1] *MATLAB Reference Manual.* MA: The MathWorks, 1986.
[2] D. S. Bernstein, W. M. Haddad, and C. N. Nett, "Minimal complexity control law synthesis, part 1: Problem formulation and reduction to optimal static output feedback," in *Proc. 1989 Amer. Contr. Conf.*, June 1989, pp. 2056–2064.
[3] J. A. Brown and M. A. Jenkins, "The apl identity crisis," *APL Quote Quad*, vol. 12, pp. 62–66, 1981.
[4] R. Y. Chiang and M. G. Safonov, *MATLAB Robust Control Toolbox User's Guide*. Sherborn, MA: The MathWorks 1988.
[5] C. deBoor, "An empty exercise," *SIGNUM*, vol. 25, pp. 2–6, 1990.
[6] I. Gohberg and S. Rubinstein, "Cascade decompositions of rational matrix functions and their stability," *Int. J. Contr.*, vol. 46, no. 2, pp. 603–629, 1987.
[7] A. J. Laub and J. N. Little, *MATLAB Control system Toolbox User's Guide*. MA: The MathWorks, 1986.
[8] C. N. Nett and W. M. Haddad, "A systems-theoretic appropriate realization of the empty matrix concept," in *Proc. Amer. Contr. Conf.*, June 1991 pp. 978–983.
[9] R. M. Redheffer, "On a certain linear fractional transformation," *J. Math. Phys.*, vol. 39, pp. 269–286, 1960.
[10] J. Stoer and C. Witzgall, *Convexity and Optimization in Finite Dimensions I.* New York: Springer-Verlag, 1970.