K.U.Leuven

Department of Electrical Engineering (ESAT) **SISTA**

Technical report 96-69

Matrix factorization and minimal state space realization in the max-plus algebra*

B. De Schutter and B. De Moor

If you want to cite this report, please use the following reference instead:
B. De Schutter and B. De Moor, "Matrix factorization and minimal state space realization in the max-plus algebra," in *Proceedings of the 1997 American Control Conference (ACC'97)*, Albuquerque, New Mexico, pp. 3136–3140, June 1997.

ESAT-SISTA
K.U. Leuven
Kardinaal Mercierlaan 94
B-3001 Leuven (Heverlee)
Belgium
phone: +32-16-32.17.09 (secretary)
fax: +32-16-32.19.70
URL: `http://www.esat.kuleuven.ac.be/sista`

*This report can also be downloaded via `http://lcewww.et.tudelft.nl/~deschutt/pub/abstracts/96_69_abstr.html`

Matrix Factorization and Minimal State Space Realization in the Max-Plus Algebra

Bart De Schutter* and Bart De Moor†
ESAT-SISTA, K.U.Leuven, Kardinaal Mercierlaan 94,
B-3001 Leuven (Heverlee), Belgium.
bart.deschutter@esat.kuleuven.ac.be,
bart.demoor@esat.kuleuven.ac.be

Abstract

The topics of this paper are matrix factorizations and the minimal state space realization problem in the max-plus algebra, which is one of the modeling frameworks that can be used to model discrete event systems. We present a heuristic algorithm to compute a factorization of a matrix in the max-plus algebra. Next we use this algorithm to determine the minimal system order (and to construct a minimal state space realization) of a max-linear time-invariant discrete event system.

1. Introduction

Typical examples of discrete event systems (DESs) are flexible manufacturing systems, telecommunication networks, parallel processing systems and railroad traffic networks. There exists a wide range of frameworks to model and to analyze DESs: Petri nets, formal languages, computer simulation, perturbation analysis, etc. In this paper we concentrate on a subclass of DESs that can be described using the max-plus algebra [1, 2], which has maximization and addition as basic operations. Although the description of DESs that belong to this subclass is nonlinear in conventional algebra, the model becomes "linear" when we formulate it in the max-plus algebra. One of the main advantages of an analytic max-algebraic model of a DES is that it allows us to derive some properties of the system (in particular the steady state behavior) fairly easily, whereas in some cases brute force simulation might require a large amount of computation time.

Although there are many analogies between the max-plus algebra and linear algebra (there exist max-algebraic equivalents of Cramer's rule, the Cayley-Hamilton theorem, eigenvectors and eigenvalues), there are also some major differences that prevent a straightforward translation of properties and algorithms from linear algebra to the max-plus algebra. As a result many problems that can be solved rather easily in linear system theory are not that easy to solve in max-algebraic system theory. In this paper we address such a problem: the minimal state space realization problem in the max-plus algebra.

2. The max-plus algebra

In this section we give a short introduction to the max-plus algebra. For a more complete overview the interested reader is referred to [1, 3].

Define $\varepsilon = -\infty$ and $\mathbb{R}_\varepsilon = \mathbb{R} \cup \{-\infty\}$. The basic max-algebraic operations are defined as follows:

$$\begin{aligned} a \oplus b &= \max(a, b) \\ a \otimes b &= a + b \end{aligned}$$

where $a, b \in \mathbb{R}_\varepsilon$. The structure $\mathbb{R}_{\max} = (\mathbb{R}_\varepsilon, \oplus, \otimes)$ is called the *max-plus algebra*. The $\oplus$ operation is called the max-algebraic addition and $\otimes$ is called the max-algebraic multiplication.

The basic max-algebraic operations are extended to matrices as follows. If $A, B \in \mathbb{R}_\varepsilon^{m \times n}$ then

$$(A \oplus B)_{ij} = a_{ij} \oplus b_{ij} \qquad \text{for all } i, j;$$

if $A \in \mathbb{R}_\varepsilon^{m \times p}$ and $B \in \mathbb{R}_\varepsilon^{p \times n}$ then

$$(A \otimes B)_{ij} = \bigoplus_{k=1}^{p} a_{ik} \otimes b_{kj} \qquad \text{for all } i, j\,.$$

The m by n zero matrix in the max-plus algebra is denoted by $\mathcal{E}_{m \times n}$: $(\mathcal{E}_{m \times n})_{ij} = \varepsilon$ for all i, j. A square matrix D is a max-algebraic diagonal matrix if $d_{ij} = \varepsilon$ for all i, j with $i \neq j$. The max-algebraic matrix power of a square matrix A is defined as follows:

$$A^{\otimes^k} = \underbrace{A \otimes A \otimes \ldots \otimes A}_{k \text{ times}} \qquad \text{for } k \in \mathbb{N}_0\,.$$

We shall also use the min-algebraic operations $\oplus'$ (min) and $\otimes'$ (+). These operations are extended to matrices in the same way as $\oplus$ and $\otimes$.

Let $A \in \mathbb{R}_\varepsilon^{m \times n}$ and $b \in \mathbb{R}_\varepsilon^{m}$. Although the system $A \otimes x = b$ does not always have a solution, the largest solution $\tilde{x}$ of the system $A \otimes x \leq b$ always exists. This

---

*Senior Research Assistant with the F.W.O. (Fund for Scientific Research – Flanders).

†Senior Research Associate with the F.W.O.

unique solution $\tilde{x}$ is called the *greatest subsolution* of $A \otimes x = b$ and can be computed as $\tilde{x} = (-A^T) \otimes' b$.

If A is a matrix then $A_{.i}$ is the ith column of A. We use $A_{\{1,2,\dots,n\},.}$ to denote the submatrix of A obtained by removing all rows of A except for the first n rows, and $A_{.,\{1,2,\dots,m\}}$ to denote the submatrix of A obtained by removing all columns of A except for the first m columns. The b-norm of a matrix A is defined by $\|A\|_{\mathrm{b}} = \sum_{i,j} |a_{ij}|$.

Definition 1 (Max-algebraic weak column rank) *Let $A \in \mathbb{R}_\varepsilon^{m \times n}$ with $A \neq \mathcal{E}_{m \times n}$. The max-algebraic weak column rank of A is defined by*

$$\operatorname{rank}_{\oplus,\mathrm{wc}}(A) = \min \left\{ \#I \;\middle|\; \begin{array}{l} I \subseteq \{1,2,\dots,n\} \text{ and} \\ \forall k \in \{1,2,\dots,n\}, \exists l \in \mathbb{N}_0, \\ \exists i_1, i_2, \dots, i_l \in I, \\ \exists \alpha_1, \alpha_2, \dots, \alpha_l \in \mathbb{R}_\varepsilon \\ \text{such that } A_{.k} = \bigoplus_{j=1}^{l} \alpha_j \otimes A_{.i_j} \end{array} \right\} .$$

If $A = \mathcal{E}_{m \times n}$ then we have $\operatorname{rank}_{\oplus,\mathrm{wc}}(A) = 0$.

A more formal definition of the max-algebraic weak column rank can be found in [8]. Efficient methods to compute the max-algebraic weak column rank of a matrix are described in [3, 8].

3. A heuristic algorithm for max-algebraic matrix factorization

Consider the following problem:

> Given a matrix $A \in \mathbb{R}_\varepsilon^{m \times n}$ and an integer $r \in \mathbb{N}_0$, find $U \in \mathbb{R}_\varepsilon^{m \times r}$ and $V \in \mathbb{R}_\varepsilon^{r \times n}$ such that $A = U \otimes V$, or show that no such factorization exists.

In [4, 7] we have shown that this problem can be considered as a special case of an Extended Linear Complementarity Problem (ELCP), which is a kind of mathematical programming problem. However, the general ELCP is NP-hard [4, 6]. Furthermore, the general ELCP algorithm that we have developed in [4, 6] would on the average require a CPU time of the order $(mnr)^{(m+n)r}$ to compute a factorization $U \otimes V$ with inner dimension r of an m by n matrix A. Therefore, we now present a heuristic algorithm to compute such a matrix factorization.

In this algorithm we start with an initial guess U_0 for U and we compute V as the greatest subsolution of $U_0 \otimes X = A$. Next we compute U as the greatest subsolution of $X \otimes V = A$. Then we start an iterative procedure in which in each step we select an entry of U or V, adapt it and recompute U and V as the greatest subsolution of respectively $X \otimes V = A$ and $U \otimes X = A$. In that way we construct a sequence of matrices U_k, V_k such that $\|A - U_k \otimes V_k\|_{\mathrm{b}}$ decreases monotonously as k increases.

Given a tolerance $\tau \geq 0$ and an initial guess U_0 for U, the following algorithm either returns U and V such that $\|A - U \otimes V\|_{\mathrm{b}} \leq \tau$ or exits with an error message.

A heuristic algorithm to compute a max-algebraic matrix factorization

Input: m, n, r, A, U_0, τ

Initialization:

- $V_0 \leftarrow (-U_0^T) \otimes' A$
- $U_0 \leftarrow A \otimes' (-V_0^T)$
- $D \leftarrow A - U_0 \otimes V_0$
- $f \leftarrow \|D\|_{\mathrm{b}}$

Main loop:

```
while f > τ do
    for all pairs (i,j) ∈ {1,2,...,m} ×
          {1,2,...,n} do
        for k = 1,2,...,r do
            Ũ ← U
            ũ_ik ← a_ij − v_kj
            Ṽ ← (−Ũ^T) ⊗' A
            Ũ ← A ⊗' (−Ṽ^T)
            f̃ ← ||A − Ũ ⊗ Ṽ||_b
            if f̃ < f then
                exit from the outer for loop
            endif
        endfor
        for k = 1,2,...,r do
            Ṽ ← V
            ṽ_kj ← a_ij − u_ik
            Ũ ← A ⊗' (−Ṽ^T)
            Ṽ ← (−Ũ^T) ⊗' A
            f̃ ← ||A − Ũ ⊗ Ṽ||_b
            if f̃ < f then
                exit from the outer for loop
            endif
        endfor
    endfor
    if f̃ < f then
        U ← Ũ
        V ← Ṽ
        D ← A − U ⊗ V
        f ← f̃
    else
        exit (no solution found)
    endif
endwhile
```

Output: U, V, f

We have presented the algorithm in its most elementary form. Several optimizations are possible: it is obvious that in the for loops we do not have to consider the combinations of i, j, k for which $u_{ik} + v_{kj} = a_{ij}$ since then we have $\tilde{U} = U$ and $\tilde{V} = V$. Also note that the b-norm of D and $A - \tilde{U} \otimes \tilde{V}$ is equal to the sum of the entries of the matrix since the entries of these matrices are always nonnegative by construction.

We have tested our heuristic algorithm as follows. Given a number $n > 4$ we define $r = \lfloor \frac{n}{2} \rfloor$. Next we construct random matrices $P \in \mathbb{R}^{n \times r}$ and $Q \in \mathbb{R}^{r \times n}$ with entries that are uniformly distributed in the interval $[-b, b]$ where b is a random integer in the interval $[4, 3 + 2rn]$. Next we define $A = P \otimes Q$. Then we scale[1] the matrix A such that the largest entry in each row and in each column is equal to 0. This yields the matrix $\hat{A}$.

We select an initial $\hat{U}_0$ with random entries in the interval $[c, 0]$ where c is the smallest (negative) entry in the scaled matrix $\hat{A}$. Then we use our heuristic algorithm to factorize $\hat{A}$ as $\hat{U} \otimes \hat{V}$ with $\hat{U} \in \mathbb{R}^{n \times r}$ and $\hat{V} \in \mathbb{R}^{r \times n}$. If the algorithm does not yield a result, we select an initial $\hat{V}_0$ with entries in the interval $[c, 0]$ and use the algorithm again (but now on $\hat{A}^T$ since $\hat{U} \otimes \hat{V} = \hat{A}$ is equivalent to $\hat{V}^T \otimes \hat{U}^T = \hat{A}^T$). If the algorithm still does not yield a result, we select another $\hat{U}_0$ and so on.

If we finally get a solution, we use the inverse scaling to obtain a factorization $U \otimes V$ of A.

In Figure 1 we have plotted the average total CPU time needed by our algorithm to find a factorization of the matrix A as a function of n (the size of A). This figure shows that the total CPU time is about $10^{-8} n^8$ for this experiment. This is certainly a major improvement compared with a total CPU time of the order $(n^2 r)^{2nr}$ required by an ELCP-based algorithm! The ragged appearance of the curve on Figure 1 is caused by the fact that the inner dimension of the product $P \otimes Q$ is equal to $\lfloor \frac{n}{2} \rfloor$.

There are many variations possible in the algorithm (such as e.g, considering the pairs (i, j) for which d_{ij} is the largest first, or using a different processing order in the inner for loop), but extensive experiments have shown that the algorithm given above in combination with scaling and the choice of initial matrices used in the experiment described above outperforms other variants with respect to both timing and the number of successes for the same initial conditions.

4. State space realization

Consider a DES with m inputs and l outputs that can be described by an nth order max-linear time invariant state space model of the form

$$x(k+1) = A \otimes x(k) \oplus B \otimes u(k) \quad (1)$$
$$y(k) = C \otimes x(k) \quad (2)$$

with $A \in \mathbb{R}_\varepsilon^{n \times n}$, $B \in \mathbb{R}_\varepsilon^{n \times m}$ and $C \in \mathbb{R}_\varepsilon^{l \times n}$. The vector x represents the state, u is the input vector and y is the output vector of the system. We shall characterize a model of the form (1)–(2) by the triple (A, B, C) of system matrices. A DES the behavior of which can be described by equations of the form (1)–(2) will be called a *max-linear time-invariant DES*.

If we apply a unit impulse: $e(k) = 0$ if $k = 0$, and $e(k) = \varepsilon$ if $k \neq 0$, to the ith input of the system and if $x(0) = \mathcal{E}_{n \times 1}$, we get $y(k) = C \otimes A^{\otimes^{k-1}} \otimes B_{.i}$ for $k = 1, 2, \ldots$ as the output of the system. We could do this for all inputs and store the outputs in l by m matrices $G_k = C \otimes A^{\otimes^k} \otimes B$ for $k = 0, 1, \ldots$ The G_k's are called the *impulse response matrices*, and the sequence $\{G_k\}_{k=0}^{\infty}$ is called the *impulse response*.

Suppose that A, B and C are unknown, and that we only know the impulse response. The *state space realization problem* consists in constructing the system matrices A, B and C starting from the impulse response. The smallest possible size of the system matrix A over all triples (A, B, C) of state space realizations of the impulse response is called the *minimal system order* and the corresponding triple (A, B, C) is called a *minimal state space realization*.

If $\{G_k\}_{k=0}^{\infty}$ is the impulse response of a max-linear

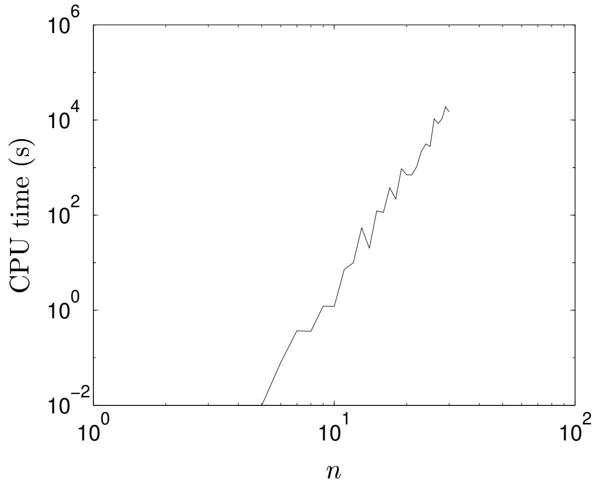

Figure 1: The average CPU time used by our algorithm to factorize a random n by n matrix as a max-algebraic product of an n by r and an r by n matrix with $r = \lfloor \frac{n}{2} \rfloor$.

[1] This can be done by pre- and post-multiplying A by appropriate max-algebraic diagonal matrices.

time-invariant DES and $N \in \mathbb{N}_0 \cup \{\infty\}$ then we define

$$H_N \stackrel{\text{def}}{=} \begin{bmatrix} G_0 & G_1 & G_2 & \dots & G_{N-1} \\ G_1 & G_2 & G_3 & \dots & G_N \\ G_2 & G_3 & G_4 & \dots & G_{N+1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ G_{N-1} & G_N & \dots & \dots & G_{2N-2} \end{bmatrix} .$$

If we want to compute a minimal state space realization of a max-linear time-invariant DES starting from its impulse response $\{G_k\}_{k=0}^{\infty}$, we first have to determine the minimal system order n.

We could use max-algebraic matrix ranks of H_∞ (or H_N, where N is a large integer with $N \gg n$) to determine upper and lower bounds for the minimal system order the max-algebraic weak column rank of H_∞ is an upper bound for the minimal system order and the max-algebraic minor rank is a lower bound for the minimal system order [8, 9]. However, at present there does not exist an efficient polynomial time algorithm to compute the max-algebraic minor matrix rank. Therefore, we propose to use our heuristic algorithm to compute a lower bound r for the minimal system order.

We shall use the following proposition which is a generalization to the multiple-input multiple-output case of Proposition 2.3.1 of [8, p. 175]. It is also an adaptation to max-linear systems of a similar theorem for linear nonnegative systems [10, Theorem 5.4.10].

Proposition 2 *Let $\{G\}_{k=0}^{\infty}$ be the impulse response of a max-linear time-invariant DES with m inputs and l outputs. Let n be the smallest integer for which there exist matrices $A \in \mathbb{R}_\varepsilon^{n\times n}$, $U \in \mathbb{R}_\varepsilon^{\infty\times n}$ and $V \in \mathbb{R}_\varepsilon^{n\times\infty}$ such that*

$$H_\infty = U \otimes V \quad (3)$$
$$U \otimes A = \overline{U} \quad (4)$$

where $\overline{U}$ is the matrix obtained by removing the first l rows of U. Then n is equal to the minimal system order.

Remark: This proposition also holds if we replace H_∞ in (3) by the matrix that contains the first m columns of H_∞ and if V is an n by m matrix. However, we have noticed that the weaker formulation given above leads to better results if we use our heuristic matrix factorization algorithm to determine the minimal system order and the system matrices of a minimal state space realization.

It is easy to verify that if we have a minimal decomposition of the form (3) – (4) of H_∞ then the triple $(A, V_{.,\{1,2,\dots,m\}}, U_{\{1,2,\dots,l\},.})$ is a minimal state space realization of the given impulse response.

Now we make the following assumptions:

- the entries of all the G_k's are finite,
- the DES exhibits a periodic steady state behavior of the following kind:

$$\exists n_0, d \in \mathbb{N},\ \exists c \in \mathbb{R} \text{ such that}$$
$$\forall n > n_0 : G_{n+d} = c^{\otimes^d} \otimes G_n \ . \quad (5)$$

It can be shown [1, 8] that a sufficient condition for (5) to hold is that the system matrix A is irreducible, i.e., $(A \oplus A^{\otimes^2} \oplus \dots \oplus A^{\otimes^n})_{ij} \neq \varepsilon$ for all i, j. This will, e.g., be the case for DESs without separate independent subsystems and with a cyclic behavior or with feedback from the output to the input like, e.g., flexible production systems in which the parts are carried around on a limited number of pallets that circulate in the system. The kind of steady state behavior mentioned above can also occur if the system matrix A is not irreducible.

If the assumptions stated above hold, then it can be proved [4] that the *partial realization problem* — in which we look for a realization that only fits the first, say, N Markov parameters — if solvable, always admits a solution with finite entries (which is necessary in order to apply our heuristic algorithm). If N is large enough then every partial realization will also be a realization of the entire impulse response. In practice it appears that we should at least include the transient behavior and the first cycles of the steady state behavior.

Now we use our heuristic algorithm in combination with a binary search procedure to compute a lower bound r for the minimal system order. We start with two extreme values for r: $l = 1$ and $u = \text{rank}_{\oplus,\text{wc}}(H_N)$. Then we set $r = \left\lfloor \frac{l+u}{2} \right\rfloor$ and we try to decompose H_N as $U \otimes V$ with inner dimension r. If this is possible, we set $u = r$; otherwise, we set $l = r+1$. We repeat the above procedure until we finally find the smallest integer r such that H_N can be decomposed as $U \otimes V$ with $U \in \mathbb{R}^{Nl\times r}$ and $V \in \mathbb{R}^{r\times Nm}$. As a consequence of Proposition 2, r will be a lower bound for the minimal system order. Note that even if we do not take N large enough, r will still be less than or equal to the minimal system order.

Experiments in which random triples of system matrices were constructed using a method similar to the one described in Section 3 have shown that our heuristic algorithm performs significantly better for matrices of the form H_N than for general, arbitrary matrices: the average CPU time needed to compute a factorization is several orders of magnitude smaller, and in almost all cases (99.9 %) only one run or initial matrix U_0 is needed to find a factorization.

Once (a lower bound for) the minimal system order has been determined, we can determine the system matrices. In [4, 5] we have developed an ELCP-based approach to compute the system matrices (but this requires CPU times that on the average depend exponentially on n, m and l, and polynomially on N). Therefore, it would certainly be useful to extend our heuristic algorithm such that condition (4) is also taken into account. This will be a topic for further research.

However, in our experiments we have noticed that the results of our binary search procedure can often directly be used to determine A, B and C. In general the matrices U and V that result from this binary procedure only satisfy (3). However, if we define $A = (-\underline{U}^T) \otimes' \overline{U}$ where $\underline{U}$ is the matrix obtained by removing the last l rows of U, then many times A and U also satisfy (4): e.g., in a generic experiment with single-input single-output systems this held true for 95 % of the cases with $n = 6$ and decreased almost linearly to 50 % for $n = 20$. Note that if necessary we could augment these percentages significantly by repeating the last step of the binary search procedure with an other initial matrix U_0.

Note that if assumption (5) does not hold we could compute a finite realization of a finite number N of impulse response matrices and we see how the entries of the resulting matrices A, B and C evolve as N becomes larger and larger. A similar limit procedure could be used if some of the entries of the G_k's are equal to ε.

5. Conclusions and future research

We have presented a heuristic algorithm that can be used to compute a max-algebraic matrix factorization of a given matrix. Experiments indicate that the average CPU time needed by this algorithm is polynomial in the size of the given matrix, which is a significant improvement over the existing algorithm for computing max-algebraic matrix factorizations, which requires a CPU time that increases exponentially as the size of the given matrix increases.

We have shown how our algorithm can be used to determine a lower bound for the minimal system order of a max-linear time-invariant DES starting from its impulse response. Experiments have shown that our algorithm performs significantly faster on the matrices that appear in this procedure than on arbitrary matrices. Moreover, our procedure for determining a lower bound for the minimal system order can often also be used to compute the system matrices of a minimal state space realization.

Future research topics are: improvement of the current algorithm, further investigation of the effects of preprocessing and an appropriate choice for the initial matrices, and development of algorithms that will always generate the exact minimal system order (instead of a lower bound) and the corresponding system matrices.

Acknowledgments

This research was sponsored by the Concerted Action Project of the Flemish Community, entitled "Model-based Information Processing Systems", by the Belgian program on interuniversity attraction poles (IUAP-02), and by the ALAPEDES project of the European Community Training and Mobility of Researchers Program.

References

[1] F. Baccelli, G. Cohen, G.J. Olsder, and J.P. Quadrat, *Synchronization and Linearity.* New York: John Wiley & Sons, 1992.

[2] G. Cohen, D. Dubois, J.P. Quadrat, and M. Viot, "A linear-system-theoretic view of discrete-event processes and its use for performance evaluation in manufacturing," *IEEE Transactions on Automatic Control*, vol. 30, no. 3, pp. 210–220, Mar. 1985.

[3] R.A. Cuninghame-Green, *Minimax Algebra*, vol. 166 of *Lecture Notes in Economics and Mathematical Systems.* Berlin, Germany: Springer-Verlag, 1979.

[4] B. De Schutter, *Max-Algebraic System Theory for Discrete Event Systems.* PhD thesis, Faculty of Applied Sciences, K.U.Leuven, Leuven, Belgium, 1996.

[5] B. De Schutter and B. De Moor, "Minimal realization in the max algebra is an extended linear complementarity problem," Tech. rep. 93-70, ESAT-SISTA, K.U.Leuven, Leuven, Belgium, Dec. 1993.

[6] B. De Schutter and B. De Moor, "The extended linear complementarity problem," *Mathematical Programming*, vol. 71, no. 3, pp. 289–325, Dec. 1995.

[7] B. De Schutter and B. De Moor, "A method to find all solutions of a system of multivariate polynomial equalities and inequalities in the max algebra," *Discrete Event Dynamic Systems: Theory and Applications*, vol. 6, no. 2, pp. 115–138, Mar. 1996.

[8] S. Gaubert, *Théorie des Systèmes Linéaires dans les Dioïdes.* PhD thesis, Ecole Nationale Supérieure des Mines de Paris, France, July 1992.

[9] S. Gaubert, "On rational series in one variable over certain dioids," Tech. rep. 2162, INRIA, Le Chesnay, France, Jan. 1994.

[10] J.M. van den Hof, *System Theory and System Identification of Compartmental Systems.* PhD thesis, Faculty of Mathematics and Natural Sciences, University of Groningen, Groningen, The Netherlands, Nov. 1996.