Proceedings of the
44th IEEE Conference on Decision and Control, and
the European Control Conference 2005
Seville, Spain, December 12-15, 2005

Interpolation with bilinear differential forms

Ishan Pendharkar* , Paolo Rapisarda** , Harish K. Pillai*

Abstract— **We present a recursive algorithm for modeling with bilinear differential forms. We discuss applications of this algorithm for interpolation with symmetric bivariate polynomials, and for computing storage functions for autonomous systems.**

I. Introduction

Modeling a system from its time series data is an important problem. Obtaining a linear (ARMA) model that explains the given data has been the preferred modeling approach, primarily because of a good understanding of linear systems and also because of the availability of efficient computational algorithms for modeling. However, there are several systems (econometric systems for example) and important applications (signal filtering,[6]) where it has been shown that going beyond a linear model has advantages. A quadratic model (or a filter) has been seen to be satisfactory in several cases. It is worth emphasizing that though the quadratic model is obviously nonlinear, determining its parameters from the data is still a linear problem. This makes the model amenable to several optimization procedures (like minimizing squared error) and tractable design procedures. It is easily seen that this linear dependence of model parameters on the data makes the modeling problem equivalent to the solution of a system of linear equations.

In this paper, the (nonlinear) modeling problem is considered in the behavioral theoretic framework [5]. Indeed, it is not incorrect to say that linear time series modeling has played a key role in the formulation and development of the behavioral approach to systems theory. The three-part seminal papers by Willems [7] put up a strong case for a linear behavioral theoretic (as opposed to an input-output) model that explains given time series data. Generalizing along similar lines as [7], the present work considers modeling of *quadratics* which explain a given time series. Modeling of a quadratic is made difficult because of the so called "cross coupling" between various modes in the time series: given trajectories with frequencies σ_1 and σ_2, the image of these trajectories under a quadratic model will, in general, contain components with frequencies $2\sigma_1$, $2\sigma_2$ and $\sigma_1 + \sigma_2$.

Most modeling schemes for quadratics can be considered as (bilinear) interpolation problems. The problem is thus to determine a bilinear form that takes prescribed values along certain prescribed directions (often specified by the data). As has been already stated above, finding parameters of a bilinear model from given data is a linear problem. This makes the it equivalent to solution of a system of linear equations. However, this brute-force way of solution presumes several things: for one, data is assumed to be known *a priori*. Modifications in the data may often necessitate re-computation of the whole solution. These and several other limitations motivate the search for an algorithm that models data iteratively, i.e. the bilinear form should be modeled depending on the current and past available data– the current model should be *updated* to explain future data as and when such data are available.

In the behavioral framework, iterative modeling of linear systems is well understood. Iterative procedures are known for computing a "most powerful" behavioral model that explains the given data [1]. In this paper, a similar behavioral theoretic scheme is developed for modeling of bilinear forms using an interpolation approach. The bilinear form in question is a Bilinear Differential Form, (BLDF). BLDFs are bilinear functionals acting on some function space, which we assume to be $\mathcal{C}^\infty$, the space of infinitely many times differentiable functions. Developed mainly by Trentelman and Willems, [8], BLDFs have been found to be extremely useful in several problems in systems theory, especially dissipative systems [8], H_∞ control, nonlinear systems analysis [3], among others.

In this paper, the following *exact modeling* problem is studied: Given k distinct trajectories indexed by $i \in \{1, 2, \dots k\}$, $c_i exp_{\lambda_i t}$ with $c_i \in \mathbb{R}^q$, $\lambda_i \in \mathbb{R}$ and distinct, and $exp_x \equiv e^x$, determine a BLDF L_Φ that takes prescribed values along these trajectories. A more precise statement of the problem shall require a few preliminaries, including definitions of all relevant terms. In the next section, some of this background is covered. Particular emphasis is given on the linear modeling problem, along with the notation and preliminary results on BLDFs.

II. Background and preliminary results

It is assumed that the reader is familiar with the basic philosophy and notation of behavioral systems theory. A detailed exposition of some of the basic concepts in this theory can be found in [5]. Linear behaviors are those that are characterized by linear laws. Trajectories in a linear behavior form a subspace of the ambient space and can be expressed as the "kernel" of a differential operator. Depending on the freedom associated with steering one trajectory of a behavior to any other, behaviors are classified as controllable or uncontrollable. Autonomous behaviors represent the

*Department of Electrical Engineering, Indian Institute of Technology Bombay Powai, Mumbai 400076, India.

** School of Electronics and Computer Science, University of Southampton, United Kingdom.

0-7803-9568-9/05/$20.00 ©2005 IEEE

extreme form of uncontrollability in that there is no freedom to switch from one trajectory to the other. Trajectories in an autonomous behavior are completely determined by the initial conditions. A linear autonomous behavior is a finite dimensional subspace of the ambient space (which is of course, infinite dimensional).

In the context of exact modeling, by "data" we mean a finite set of measurements of a phenomenon. A model M for the data is a subset of the ambient space containing the data. M is often chosen from a "model class" $\mathcal{M}$ which presumes certain features about the model like linearity, time invariance etc. Clearly, in this framework, the data admit several models. However, it is possible to order these models according to their prohibitive power: given the data, a model class $\mathcal{M}$, and two models M_1 and M_2 for the data from the class $\mathcal{M}$, M_1 is said to be more powerful than M_2 if $M_1 \subset M_2$. A model M^* from a model class $\mathcal{M}$ is said to be a Most Powerful Unfalsified Model (MPUM) for the data if it contains the data and if every other model M from $\mathcal{M}$ for the behavior satisfies $M^* \subseteq M$. In the sequel, we assume that the data is given as

$$\mathcal{D} = \{c_i exp_{\lambda_i t}, i = 1, 2 \ldots, k, c_i \in \mathbb{R}^q, \lambda_i \in \mathbb{R}\}$$

Let us address the question of linear, time-invariant (LTI) models for $\mathcal{D}$. It has been shown in [1] that there exists a unique LTI MPUM for $\mathcal{D}$ which can be expressed as the kernel of a differential operator:

$$R(\frac{d}{dt}) c_i exp_{\lambda_i t} = 0, \; i = 1, \ldots, k \tag{1}$$

Note that because of the assumption of linearity, the MPUM also contains the linear span of $c_i exp_{\lambda_i t}, i = 1, \ldots, k$. $R(\xi)$ a $q \times q$ polynomial matrix which is full rank for almost all $\xi \in \mathbb{C}$. Since $R(\frac{d}{dt})$ represents a MPUM, the points of singularity of the polynomial matrix $R(\xi)$ are precisely the points $\{\lambda_i\}$. As has been mentioned before, algorithms are known which compute a MPUM recursively, i.e. a MPUM for the span of $c_1 exp_{\lambda_1 t}, \ldots c_{k+1} exp_{\lambda_{k+1} t}$ can be computed from a MPUM for the span of $c_1 exp_{\lambda_1 t}, \ldots c_k exp_{\lambda_k t}$.

In systems theory, quadratic functionals of variables and their derivatives are often encountered, for instance, the Lyapunov function in stability theory, the Lagrangian of a dynamical system, the Hamiltonian, various cost criteria in optimal control, etc. In [8] a two variable polynomial matrix was used to represent such quadratic functionals. This notation has proved to be extremal handy. With this notation, operations on (infinite dimensional) trajectories can be conveniently represented as algebraic operations on (finite dimensional) polynomial matrices, albeit in two variables.

Let $\mathbb{R}^{q_1 \times q_2}[\zeta, \eta]$ denote the set of real polynomial matrices in the indeterminates ζ and η. An element Φ in this set is given by

$$\Phi(\zeta, \eta) = \sum_{k,l} \Phi_{kl} \zeta^k \eta^l \tag{2}$$

where the sum ranges over non-negative integers k, l and this sum is assumed finite (i.e. only a finite number of Φ_{kl} are nonzero). Such a Φ induces a *bilinear differential form*

$$L_\Phi : \mathcal{C}^\infty(\mathbb{R}, \mathbb{R}^{q_1}) \times \mathcal{C}^\infty(\mathbb{R}, \mathbb{R}^{q_2}) \to \mathcal{C}^\infty(\mathbb{R}, \mathbb{R}) \tag{3}$$

defined by

$$(L_\Phi(v, w))(t) = \sum_{k,l} (\frac{d^k v(t)}{dt^k})^T \Phi_{kl} (\frac{d^l w(t)}{dt^l}). \tag{4}$$

Here, $\mathcal{C}^\infty(\mathbb{R}, \mathbb{R}^\bullet)$ denotes the space of $\mathcal{C}^\infty$ functions from $\mathbb{R}$ to $\mathbb{R}^\bullet$. For the special case when $q_1 = q_2 = q$, Φ induces a *quadratic differential form* (QDF) which is a map

$$Q_\Phi : \mathcal{C}^\infty(\mathbb{R}, \mathbb{R}^q) \to \mathcal{C}^\infty(\mathbb{R}, \mathbb{R}) \tag{5}$$

defined by

$$Q_\Phi(w) = L_\Phi(w, w) \tag{6}$$

Φ will be called symmetric if $\Phi(\zeta, \eta) = \Phi^T(\eta, \zeta)$. The set of such symmetric two variable polynomial matrices will be denoted by $\mathbb{R}_s^{q \times q}[\zeta, \eta]$. It can be easily shown that it is enough to consider just symmetric two variable polynomial matrices and hence in this paper it will always be assumed that $\Phi \in \mathbb{R}_s^{q \times q}[\zeta, \eta]$.

Notice that if $L_\Phi(w_1, w_2)$ is a BLDF, so is $\frac{d}{dt} L_\Phi(w_1, w_2)$. Moreover, if $Q_\Phi(w)$ is a QDF, so is $\frac{d}{dt} Q_\Phi(w)$. This operation can be conveniently expressed in the notation of two variable symmetric polynomial matrices as a multiplication by $(\zeta + \eta)$: given a BLDF $L_\Phi(w_1, w_2)$

$$\begin{aligned} \frac{d}{dt} L_\Phi(w_1, w_2) &= L_\Theta(w_1, w_2) \\ \text{with } \Theta(\zeta, \eta) &= (\zeta + \eta) \Phi(\zeta, \eta) \end{aligned}$$

Clearly, $\Theta(\zeta, \eta)$ is symmetric if $\Phi(\zeta, \eta)$ is symmetric. The background developed here shall now be used to give a precise problem statement. Also, the preliminary results given above will be invoked time and again throughout this paper.

III. The problem statement

Given k distinct trajectories $c_i exp_{\lambda_i t}$ with $c_i \in \mathbb{R}^q$ and $\lambda_i \in \mathbb{R}$ and distinct, and $i = 1, 2, \ldots k$, determine a BLDF L_Φ such that

$$L_\Phi(c_i exp_{\lambda_i t}, c_j exp_{\lambda_j t}) = q_{ij} exp_{(\lambda_i + \lambda_j) t} \;\; i, j = 1, 2, \ldots k \tag{7}$$

with q_{ij} being given real constants. To simplify the treatment presented in this paper we assume that $c_i^T c_j \neq 0$ for all $i, j = 1, \ldots, k$.

Due to symmetry of $\Phi(\zeta, \eta)$, conditions (7) automatically fix values of $L_\Phi(c_j exp_{\lambda_j t}, c_i exp_{\lambda_i t})$. Clearly, when $i = j$, the problem reduces to determining a QDF which takes the values $q_{ii} exp_{2\lambda_i t}$. Since $\frac{d}{dt}^n exp_{\lambda t} = \lambda^n exp_{\lambda t}, n = 1, 2, \ldots$, conditions (7) reduce to the following: find a $\Phi(\zeta, \eta)$ such that

$$c_i^T \Phi(\lambda_i, \lambda_j) c_j = q_{ij} \tag{8}$$

Notice that now the problem has been re-formulated as an operation on polynomial matrices and real (finite dimensional) vector spaces. However, as with all polynomial interpolation problems, the following issues need to be addressed

- What is the "best" $\Phi(\zeta,\eta)$ that interpolates the data
- What is the required computational effort
- Can the interpolation be done recursively with respect to data

Indeed, often, what is a "best" interpolant is a question that can only be answered depending on the application at hand. The choice is made more complicated by the fact that there is no natural ordering on polynomials in two variables. It has often been thought advantageous to have a $\Phi(\zeta,\eta)$ with the least "effective size", i.e. minimizing the degree of the leading $\zeta^{\bullet}\eta^{\bullet}$ term. In some applications (like error correcting codes), the criterion has been the least weighted degree [2]. In this paper, the focus is on the number of parameters. The interpolation scheme suggested here depends linearly on the number of data trajectories. The most interesting feature of this scheme is that it is recursive. The required computational effort is modest– one can write down the solution by hand without much difficulty at least in the more simple cases.

IV. A RECURSIVE ALGORITHM FOR INTERPOLATING WITH QDFS

Recall that the problem of interpolating BLDFs is the following: given k distinct trajectories $c_i exp_{\lambda_i t}$ with $c_i \in \mathbb{R}^q$ and $\lambda_i \in \mathbb{R}$ and distinct, and $i = 1,2,\dots k$, determine a BLDF L_Φ with $\Phi(\zeta,\eta) \in \mathbb{R}_s^{q\times q}[\zeta,\eta]$ such that

$$c_i^T \Phi(\lambda_i,\lambda_j) c_j = q_{ij} \tag{9}$$

where q_{ij} are given real constants.

Suppose l of the possible k trajectories $c_i exp_{\lambda_i t}$ with $i \in \{1,2\dots l\}$ are given. Let Ker$M_l(\frac{d}{dt})$ denote the MPUM for these l trajectories. That is:

$$M_l(\lambda_i)c_i = 0 \;\; i = 1,2\dots l \tag{10}$$

Thus, the matrices $M_l(\lambda_i)$ are necessarily singular (except for the trivial case when $c_i = 0$). The fact that $M_l(\frac{d}{dt})$ is a *most powerful* unfalsified model (as against *any*) model implies that $M_l(\xi)$ is full rank everywhere except the points $\lambda_1,\lambda_2\dots\lambda_l$. Now consider the $q\times q$ symmetric two variable polynomial matrix

$$\Upsilon(\zeta,\eta) = F(\zeta,\eta)M_l(\eta) + M_l^T(\zeta)F^T(\eta,\zeta) \tag{11}$$

with $F(\zeta,\eta)$ an arbitrary two variable polynomial matrix. Consider the action of the BLDF L_Υ on the finite dimensional vector space S given by the span of $c_i exp_{\lambda_i t}$, $i = 1,2\dots l$. Since $M_l(\frac{d}{dt})$ represents a MPUM:

$$L_\Upsilon(v,w) = 0 \text{ for every vector } v,w \in S \tag{12}$$

Conversely, it has been shown in [8] that every BLDF L_Υ such that $L_\Upsilon(v,w) = 0$ for every $v,w \in S$ satisfies $\Upsilon(\zeta,\eta) = F(\zeta,\eta)M_l(\eta) + M_l^T(\zeta)F^T(\eta,\zeta)$ for some $F(\zeta,\eta) \in \mathbb{R}^{q\times q}[\zeta,\eta]$.

Thus, if $\Phi(\zeta,\eta)$ is a solution to the interpolation problem (9), every other solution can be written as $\Phi(\zeta,\eta) + \Upsilon(\zeta,\eta)$ with $\Upsilon(\zeta,\eta)$ given as in equation (11). The symmetric polynomial in equation (11) will be referred to as the "tail polynomial". It shall be suppressed in the actual statement of the algorithm for reasons of brevity. However, it must be emphasized that what is a "best" or "simplest" QDF will often depend on the choice of an appropriate tail polynomial. However, these issues will not be considered in this paper.

The interpolating BLDF for the first i given trajectories, $i = 1,2\dots k$ shall be denoted by L_{Φ_i}. Let

$$\Phi_1(\zeta,\eta) = \frac{q_{11}}{c_1^T c_1} I \tag{13}$$

here I, as usual, denotes the $q\times q$ identity matrix. This is the initialization step. In the next few lines the recursion formulae will be developed. Assume that the matrix $\Phi_l(\zeta,\eta)$ is given such that the BLDF L_{Φ_l} interpolates the first l given trajectories, i.e.

$$c_i^T \Phi(\lambda_i,\lambda_j) c_j = q_{ij} \;\; i,j = 1,2\dots l \tag{14}$$

The $(l+1)$ th update is carried with the following formula

$$\Phi_{l+1}(\zeta,\eta) = \Phi_l(\zeta,\eta) + E_{l+1}^T(\zeta)M_l(\eta) + M_l^T(\zeta)E_{l+1}(\eta) \tag{15}$$

Here, Ker$M_l(\frac{d}{dt})$ is the MPUM for the first l trajectories. Note that the procedure to compute this MPUM is itself recursive, [1]. The matrix $E_{l+1}(\zeta)$ denotes the "error matrix" due to reasons that will become clear in the sequel. Since ker$M_l(\frac{d}{dt})$ is the MPUM for the first l trajectories,

$$c_i^T \Phi_{l+1}(\lambda_i,\lambda_j)c_j = c_i^T\Phi_l(\lambda_i,\lambda_j)c_j = q_{ij} \;\; i,j = 1,2,\dots l \tag{16}$$

Thus, the updating of $\Phi_l(\zeta,\eta)$ has been achieved without disturbing the interpolation conditions satisfied by L_{Φ_l}. A solution to the $(l+1)$th step interpolating problem is obtained by constructing the univariate polynomial $E(\xi)$ that satisfies the conditions on $\Phi_{l+1}(\lambda_{l+1},\lambda_{l+1})$:

$$\begin{aligned} c_{l+1}^T[\Phi_l(\lambda_{l+1},\lambda_{l+1}) &+ \\ E_{l+1}^T(\lambda_{l+1})M_l(\lambda_{l+1}) &+ M_l^T(\lambda_{l+1})E_{l+1}(\lambda_{l+1})]c_{l+1} \\ &= q_{(l+1)(l+1)} \end{aligned} \tag{17}$$

Since ker$M_l(\frac{d}{dt})$ is the MPUM for the first l trajectories, and since all λ_is are assumed distinct, $M_l(\lambda_{l+1})$ is a constant nonsingular matrix. Thus, $E_{l+1}(\xi)$ has the following interpolation condition:

$$\begin{aligned} &E_{l+1}(\lambda_{l+1}) = \\ &M_l^{-T}(\lambda_{l+1})\frac{q_{(l+1)(l+1)} - c_{l+1}^T\Phi_l(\lambda_{l+1},\lambda_{l+1})c_{l+1}}{2c_{l+1}^T c_{l+1}} \end{aligned} \tag{18}$$

In addition, the choice of $E_{l+1}(\xi)$ has to take into account l cross coupling conditions between $(\lambda_{l+1},\lambda_l)$, $(\lambda_{l+1},\lambda_{l-1}),\dots,(\lambda_{l+1},\lambda_1)$. Using the recursion (15) and the property of a MPUM, (10), $E(\xi)$ can be seen to satisfy the following conditions:

$$E_{l+1}(\lambda_i) = M_l^{-T}(\lambda_{l+1})\frac{q_{(l+1)i} - c_{l+1}^T\Phi_l(\lambda_{l+1},\lambda_i)c_i}{c_{l+1}^T c_i} \quad i = 1,2\dots l \tag{19}$$

Notice that interpolation conditions for $E_{l+1}(\lambda_i)$ are well defined since $c_i^T c_j \neq 0$ for all $i,j=1,\ldots,k$ by assumption. Notice that with the scheme suggested above, the interpolation problem has reduced to one of univariate polynomial interpolation of $E_{l+1}(\xi)$, which is considerably easier. Denote by $A_1, A_2 \ldots A_l$ the l $q\times q$ matrices obtained on the right hand sides of equations (19) and A_{l+1} the $q\times q$ matrix obtained in the right hand side of equation (19). Determining the $q\times q$ matrix $E(\xi)$ from these $l+1$ interpolation conditions is a straightforward problem in Lagrange interpolation:

$$E_{l+1}(\xi) = \sum_{j=1}^{l+1} \frac{\prod_{i=1,i\neq j}^{l+1}(\xi-\lambda_i)}{\prod_{i=1,i\neq j}^{l+1}(\lambda_j-\lambda_i)} A_j \qquad (20)$$

This completes the development of the recursive interpolation algorithm. The complete algorithm is summarized below:

```
Input $c_i exp_{\lambda_i t}$, $c_i \in \mathbb{R}^q$, $\lambda_i \in \mathbb{R}$ and distinct, and
real constants $q_{ij}$ with $i,j=1,2\ldots k$
Output A BLDF $L_\Phi$ such that
$L_\Phi(c_i exp_{\lambda_i t}, c_j exp_{\lambda_j t}) = q_{ij}$ and $\Phi(\zeta,\eta)=\Phi^T(\eta,\zeta)$
Initialization: $\Phi_1(\zeta,\eta) = \frac{q_{11}}{c_1^T c_1} I$
for i = 1,2 ...$k-1$ do,
Compute a MPUM, $M_i(\frac{d}{dt})$ for
$c_1 exp_{\lambda_1 t}, c_2 exp_{\lambda_2 t}, \ldots c_i exp_{\lambda_i t}$
Compute the ith stage error matrix $E_i(\xi)$
from
$E_i(\lambda_j) = A_j$ $j=1,2\ldots i$ as in (20)
Update the current $\Phi_i$ using the
recursion:
$\Phi_{i+1}(\zeta,\eta) = \Phi_i(\zeta,\eta) + E_{i+1}^T(\zeta)M_i(\eta) + M_i^T(\zeta)E_{i+1}^T(\zeta)$
end for
$\Phi(\zeta,\eta) = \Phi_k(\zeta,\eta)$ and $L_\Phi$ is the required
BLDF
end
```

In the remaining parts of this paper, various applications of this algorithm will be explored. Since this algorithm has been developed from a behavioral theoretic viewpoint, emphasis is more on problems related to behavioral systems theory – dissipative systems in particular. Other areas where this algorithm finds use are being explored.

V. Scalar bivariate interpolation problem

An immediate application of this algorithm is a general recursive scheme for interpolation by symmetric scalar bivariate polynomials. Though this application is almost immediate, it has been treated separately since it illustrates several aspects of the algorithm presented above. Consider the interpolation problem defined in (9) for the scalar case, i.e. $q=1$. Without loss of generality, the c_is can now be assumed to be unity. Hence, the interpolation problem simplifies to the following: given k distinct real numbers $\lambda_1,\ldots\lambda_k$, together with the real numbers $q_{ij}, i,j=1,2\ldots k$, determine a scalar symmetric bivariate polynomial $\Phi(\zeta,\eta)$ such that

$$\Phi(\lambda_i,\lambda_j) = q_{ij} \qquad (21)$$

As in the previous case, the symmetry conditions dictate that $q_{ij}=q_{ji}$ so that only $k(k+1)/2$ of these can be specified independently. The fact that c_is are unity makes the computation of a MPUM almost trivial. Indeed, it can be seen that a MPUM $M_l(\frac{d}{dt})$ for the l trajectories $exp_{\lambda_1 t}, \ldots exp_{\lambda_l t}$ is given by

$$\prod_{i=1}^{l}(\frac{d}{dt}-\lambda_i)w = 0 \qquad (22)$$

where $w(t)$ is any trajectory in the linear span of $exp_{\lambda_1 t}, \ldots exp_{\lambda_l t}$. Thus, the scalar symmetric bivariate polynomial interpolation can be solved recursively as follows:

```
Input: $k$ distinct real numbers $\lambda_1,\ldots\lambda_k$,
together with $k(k+1)/2$ real numbers
$q_{ij}=q_{ji}$ $i,j=1,2\ldots k$
Output: Scalar symmetric bivariate
polynomial $\Phi(\zeta,\eta)$ such that $\Phi(\lambda_i,\lambda_j)=q_{ij}$
Initialize: $\Phi_1(\zeta,\eta)=q_{11}$
for i=1 to k-1 do,
Compute MPUM for $exp_{\lambda_1 t}, \ldots exp_{\lambda_i t}$:

    $M_i(\xi) = \prod_{j=1}^{i}(\xi-\lambda_j)$

Compute $i+1$th stage error matrix
satisfying:

    $E_{i+1}(\lambda_j) = \alpha \frac{q_{ij}-\Phi_i(\lambda_i,\lambda_j)}{\prod_{j=1}^{i}(\lambda_{i+1}-\lambda_j)}$
    $\alpha = 1/2$ if $i=j$ else $\alpha=1$

Update $\Phi_i(\zeta,\eta)$ using the recursion

    $\Phi_{i+1}(\zeta,\eta) = \Phi_i(\zeta,\eta) + E_{i+1}^T(\zeta)M_i(\eta) + M_i^T(\zeta)E_{i+1}(\eta)$

end for
$\Phi(\zeta,\eta)=\Phi_k(\zeta,\eta)$
end
```

An interesting corollary of the above algorithm is that given real and distinct λ_is, and *arbitrary* $q_{ij}=q_{ji}$s, there exist infinitely many symmetric scalar bivariate polynomials $\Phi(\zeta,\eta)$ such that $\Phi(\lambda_i,\lambda_j)=q_{ij}$.

VI. Dissipative systems

In this section, autonomous dissipative systems are considered. It is shown that the BLDF interpolation approach developed in this paper can be advantageously applied to several system theoretic problems. Recall that an autonomous behavior is one which has no free variables. Such a behavior is necessarily finite dimensional. Consider the autonomous behavior $\mathfrak{B}_a$ defined by the following kernel representation:

$$R(\frac{d}{dt})w = 0 \qquad (23)$$

with $w \in \mathcal{C}^\infty(\mathbb{R},\mathbb{R}^q)$ and $R(\xi)$ a nonsingular $q\times q$ polynomial matrix. Further, assume that roots of $\det R(\xi)=0$ are real and distinct. Let $\lambda_1,\lambda_2\ldots\lambda_k$ denote the k spectra of $R(\xi)$. Then, $R(\lambda_i)$ is singular for $i = 1, 2 \ldots k$. Consequently there exist nonzero vectors $c_i \in \mathbb{R}^q$ such that

$$R(\lambda_i)c_i = 0 \tag{24}$$

$\mathfrak{B}_a$ is said to be stable if $||w(t)|| \to 0$ as $t \to \infty$. It is well known that $\mathfrak{B}_a$ is stable if and only if $R(\xi)$ is a Hurwitz matrix, i.e. λ_is lie in the open left half complex plane. Consider the trajectories $c_i exp_{\lambda_i t}, i = 1, 2 \ldots k$. These trajectories form a basis for $\mathfrak{B}_a$. Hence, it is enough to consider just these trajectories in $\mathfrak{B}_a$.

Given the behavior $\mathfrak{B}_a$, and a supply rate Q_Φ, a QDF Q_Ψ is said to be a storage function for $\mathfrak{B}_a$ with respect to Q_Φ if the following "dissipation inequality" holds:

$$\frac{d}{dt}Q_\Psi(w) \leq Q_\Phi(w) \;\; \forall w \in \mathfrak{B}_a \tag{25}$$

The following important issues will now be addressed:

1) How is a Q_Ψ computed, given $\mathfrak{B}_a$ and Q_Φ.
2) Can one find a Q_Ψ even when $\mathfrak{B}_a$ is not stable.

Computing a storage function for $\mathfrak{B}_a$ has been addressed in [8]. The suggested solution scheme has been through the solution of a certain Polynomial Lyapunov Equation (PLE), a solution scheme for which was subsequently reported in [4]. In this paper, an alternate method is described.

It is easy to see that for the dissipation inequality to hold along every trajectory in $\mathfrak{B}_a$, it is necessary and sufficient that the $k \times k$ symmetric matrix $D = [d_{ij}]_{i,j=1}^k$ be negative semidefinite with:

$$d_{ij} = c_i^T[(\lambda_i + \lambda_j)\Psi(\lambda_i, \lambda_j) - \Phi(\lambda_i, \lambda_j)]c_j \tag{26}$$

It is now shown that the interpolation approach developed in this paper can be used to compute a storage function in an extremal simple manner. Most computations are trivial and all of them are operations with constant matrices. Using the interpolation formula results in a shift from handling bivariate polynomial matrices to constant real matrices – making it easy and numerically efficient to implement on a computer. Translating the dissipation inequality (25) to the realm of bivariate polynomial matrices, it is easy that *a solution* to the problem of computing storage functions is to stipulate that the d_{ij}s in (26) be zero, i.e.

$$(\lambda_i + \lambda_j)c_i^T\Psi(\lambda_i, \lambda_j)c_j = c_i^T\Phi(\lambda_i, \lambda_j)c_j \;\; i, j = 1, 2 \ldots k \tag{27}$$

If $\mathfrak{B}_a$ is assumed to be stable, $\lambda_i + \lambda_j$ is a negative real number. Consequently, $\Psi(\zeta, \eta)$ can be seen to satisfy the following $k(k+1)/2$ distinct interpolation conditions:

$$c_i^T\Psi(\lambda_i, \lambda_j)c_j = c_i^T\frac{\Phi(\lambda_i, \lambda_j)}{(\lambda_i + \lambda_j)}c_j \tag{28}$$

This is the standard interpolation problem considered in Section IV. It has already been shown that a solution can be obtained recursively in k iterations.

The assumption of stability of $\mathfrak{B}_a$ is sufficient for the solvability of (27)– it is by no means necessary. Indeed, if $\mathfrak{B}_a$ is such that $\lambda_i + \lambda_j \neq 0$, a storage function can always be found for $\mathfrak{B}_a$, given any supply Q_Φ. Further, under this assumption, every such $\mathfrak{B}_a$ is actually Φ-lossless, i.e. given any Q_Φ there exists a QDF Q_Ψ such that $\frac{d}{dt}Q_\Psi$ exactly equals Q_Φ along every trajectory in $\mathfrak{B}_a$.

Assume that $\lambda_a + \lambda_b = 0$ for some $1 \leq a, b \leq k, a \neq b$. It is now shown that one can always find a $\Psi(\zeta, \eta)$ such that D is negative definite. To start with, let

$$c_i^T\Psi(\lambda_i, \lambda_j)c_j = c_i^T\frac{\Phi(\lambda_i, \lambda_j)}{\lambda_i + \lambda_j}c_j \;\; i \neq j \;\;,\;\; \lambda_i + \lambda_j \neq 0 \tag{29}$$

$$c_i^T\Psi(\lambda_i, \lambda_j)c_j = \text{ arbitrary } i \neq j \;\;,\;\; \lambda_i + \lambda_j = 0 \tag{30}$$

It is not difficult to see that D can now be written as the sum of two matrices Λ and D':

$$D = \Lambda + D' \tag{31}$$

with $\Lambda = diag[2\lambda_i c_i^T\Psi(\lambda_i, \lambda_i)c_i]_{i=1}^k$. D' is a $q \times q$ matrix which has zeros everywhere on the off-diagonal terms, except the positions $(a, b), (b, a)$ for all $a, b, a \neq b$ such that $\lambda_a + \lambda_b = 0$. Denote by σ the maximum eigenvalue of D'. Clearly, Λ can always be chosen such that $\Lambda + D'$ is negative semidefinite. One possible solution is $\Lambda = \alpha I^{q \times q}$ with $\alpha \leq -\sigma$. Thus, along with the conditions given in (30),

$$c_i^T\Psi(\lambda_i, \lambda_i)c_i = (c_i^T\Phi(\lambda_i, \lambda_i)c_i + \alpha)/2\lambda_i \;\; i = 1 \ldots k \tag{32}$$

where $\sigma = \max \text{spec}\, D'$. The matrix $\Psi(\zeta, \eta)$ can now be determined from these interpolation conditions using the recursive algorithm outlines in Section IV. The QDF Q_Ψ is then a storage function for $\mathfrak{B}_a$ since the matrix D (26) is now negative semidefinite.

The above discussion has proved the following result about autonomous behaviors:

Theorem 6.1: Consider a behavior $\mathfrak{B}_a \equiv \text{span}\{c_i exp_{\lambda_i t}\}$ with $c_i \in \mathbb{R}^q$ and $\lambda_i \in \mathbb{R}$ and distinct, and $i = 1, 2 \ldots k$. Let Q_Φ be any QDF that induces the supply rate $Q_\Phi(w)$ trajectories $w \in \mathfrak{B}_a$. Then, $\mathfrak{B}_a$ is Φ-dissipative, i.e. there exists a QDF Q_Ψ such that

$$\frac{d}{dt}Q_\Psi(w) \leq Q_\Phi(w) \;\; w \in \mathfrak{B}_a,$$

if $\lambda_i \notin i\mathbb{R}$, $i = 1, \ldots, k$.

VII. Conclusion

In this paper, a recursive algorithm has been developed for interpolation with symmetric bilinear differential forms. A bivariate interpolation problem has first been converted into univariate interpolation problems that can be solved using known methods like Lagrange interpolation. The algorithm works with constant (rather than polynomial) matrices and hence can be easily and efficiently implemented on any general purpose computational package. The algorithm has been applied for interpolation with symmetric bivariate polynomials and for computation of storage functions for autonomous behaviors.

References

[1] A.C Antoulas, J.C. Willems, “A Behavioral Approach to Linear Exact Modeling”, *IEEE Transactions on Automatic Control*, 38 (1993), pp 1776-1801.

[2] M. Kuijper, J.W. Polderman, “Behavioral models for list decoding”, *Mathematical and computer modelling of dynamical systems*, 8 (2002), pp 429-443.

[3] I. Pendharkar, H.K. Pillai, “On a theory for nonlinear behaviors ”, *Proceedings of 16th International Symposium on Mathematical Theory of Networks and Systems, MTNS, Leuven, Belgium 2004*.

[4] R. Peeters, P. Rapisarda, “A two variable approach to solve the polynomial Lyapunov equation”, *Systems and control letters*, 42(2001), pp 117-126.

[5] J.W. Polderman, J.C. Willems, *“Introduction to mathematical systems theory: A behavioral approach”*, Springer-Verlag, 1997.

[6] G.L. Sicuranza, “Quadratic filters for signal processing”, *Proceedings of IEEE*, 80(1992), pp 1263-1285.

[7] J.C. Willems, “From Time Series to Linear System, Parts I,II and III”, *Automatica*, 22(1986), 561-580, 22(1986), 675-694, 23(987), 87-115.

[8] J.C. Willems, H.L. Trentelman, “On Quadratic Differential Forms”, *SIAM Journal of Control and Optimization*, 36 (1998), pp 1703-1749.