

QUASI-SEPARABILITY AND HIERARCHICAL QUASI-SEPARABILITY IN STOCHASTIC SIGNAL PROCESSING

Patrick Dewilde

Delft University of Technology
p.dewilde@ewi.tudelft.nl

ABSTRACT

Stochastic signal processing has reached the age of 'array processing', where the various properties of the relevant system matrix play a predominant role in the efficiency of the filtering or estimation algorithm. Classical is the Toeplitz structure, and structures that are derived from it such as 'quasi-stationarity'. In this paper we focus on a different structure of equally great importance, namely 'quasi-separability'. We start out with an introduction to basic properties and algorithms, foremost of which is an extension of the square root algorithm known from Kalman filtering. Although extremely important and useful when applicable, in many applications the notion of quasi-separability has to be extended. We discuss two main classes of extensions, which both deal with hierarchy but in very different ways. One has been called 'Hierarchical Semi-Separable' or HSS, the other is of more recent date and can be termed 'hierarchies of quasi-separable form' - they look like quasi-separable but lack the main properties. We give ways of dealing with this situation which occurs in many systems characterized by partial differential equations. An example taken from image processing (optic flow analysis) puts the problem squarely in the realm of stochastic signal processing.

1. THE NOTION OF QUASI-SEPARABILITY

In this presentation we take it for granted that a signal processing algorithm has a global (functional) representation T as a matrix acting on an input space and delivering data to an output space. Where needed we assume that these spaces are of the Euclidean type - they possess an inner product and a norm derived from it. In particular they can consist of time series of vectors or

spatial series of vectors, e.g. what would happen if video data were sequentialised. The matrix may have infinite dimensions, as would happen when it acts on time sequences existing from $t = -\infty$ to $t = \infty$. In case the system is time-invariant, then T would be Toeplitz or block-Toeplitz (when the input space respect. output space consist of sequences of vectors). It is possible to incorporate finite matrices in the framework as well as infinite ones, by accepting non-uniform sequences for the input and output spaces, namely sequences of vectors of variable dimensions. When the dimension becomes zero, then we assume that it reduces to a 'placeholder' - there is no vector at that point in time. The calculus of block matrices has to be extended to cover that case, one simply assumes e.g. that the product of a $m \times 0$ matrix with an $0 \times n$ matrix is a zero matrix of dimensions $m \times n$.

Suppose now that we dispose of a linear computation that 'runs forward in time'. If the vector of data that the computer has remembered at stage n of the computation is $x(n)$, and the input at that point is $u(n)$, while the output to be produced is $y(n)$, then the computation can be modeled as follows (notice that we apply matrices to row vectors, an easy convention in signal processing):

$$\begin{cases} x(n+1) &= x(n)A(n) + u(n)B(n) \\ y(n) &= x(n)C(n) + u(n)D(n) \end{cases} \quad (1)$$

in which $\{A(n), B(n), C(n), D(n)\}$ are matrices. Let us collect these matrices in diagonal operators $A = \text{diag}[A(n)]$ etc... and define the shift operator Z on (non-uniform) sequences as $[xZ]_n = x_{n-1}$. Then the operator corresponding to this computation is block upper triangular (indicative of the forward motion of the computation on row vectors) and given by:

$$T = D + BZ(I - AZ)^{-1}C. \quad (2)$$

We recognize a state space representation of a system, now in a rather general, 'time varying' setting. The computation will be efficient if the dimension of the state vector is not too large. Let the sequence of state dimensions be $\{\delta_n\}$, and let $\delta = \sup_n \delta_n$, then we say that T is quasi-separable of order δ (δ_n can be considered a 'local' order of semi-separability.)

More generally, an operator T will be called quasi-separable (of certain orders) if for both its upper and lower parts there exists such a local computation. In that case, there will exist 'state realizations' for both the upper and the lower part and T can be represented as

$$T = B_\ell Z^*(I - A_\ell Z^*)^{-1} C_\ell + D + B_u Z(I - A_u)^{-1} C_u \quad (3)$$

Interesting is how, given an operator T one arrives at a quasi-separable representation. This topic is called 'realization theory' and its solution goes back to the work of Kronecker, Hankel and later Ho-Kalman, Silverman-Meadows, and for the time-varying case Dewilde-van der Veen [9, 7, 11, 4]. Connected to the upper (respect. lower) triangular parts of a system there are collections of matrices which are characteristic for the 'state behaviour' of the system. These matrices factor into controllability and observability operators, from which realizations can be derived. The Hankel matrices play an essential role in 'model reduction theory', the theory that describes how an operator can be approximated by another one of much lower complexity. In many cases, quasi-separable realizations of an operator are given or can be inferred from the physics of the problem - see the examples below.

Quasi-separability helps very much in reducing the complexity of matrix computations. In fact, if δ is a number indicative of the overall state complexity, then matrix inversion can be done in $n\delta^3$ operations (in several important cases even $n\delta^2$). The same holds for other important algorithms such as Cholesky decomposition (spectral factorization), QR-factorization and even eigenvalue decompositions.

The notion of quasi-separability has known less complex predecessors. These were introduced in by Gohberg-Kailath-Koltracht [6] in the study of integral equations of the Helmholtz type. There one is interested in solving an equation of the type

$$y(t) = \int K(t, s)u(s)ds \quad (4)$$

and assumes that the kernel $K(t, s)$ has (possibly different) low rank outer decompositions for $t > s$ respect. $s > t$. When discretized these decompositions amount to a special type of quasi-separable representation, namely one in which the 'state transition operators' A_ℓ and A_u are zero. It was remarked by these authors that solving the system of equations (and the derived discretized system) can be done with a number of operations that is linear in the number of equations (and square or cubic in the dimensions of the outer representation). The rather special semi-separable representation as compared to the one mentioned earlier restricts the applicability of the method and lays at the origin of numerical difficulties that can only be solved in the more general setting.

2. HIERARCHICAL QUASI-SEPARABILITY

Quasi-separability is a very nice notion for systems in which there is only one relevant dimension of evolution, often time. What to do if there are more dimensions? The approach we take is what we could call 'hierarchical quasi-separability'. We distinguish two types. One type, pioneered by Chandrasekaran and Gu [2] has been termed 'HSS', and assumes a hierarchical structure in which each level of the hierarchy gets decomposed in four subblocks (2×2 blocks) and whereby the next level of the hierarchy is only defined on the two blocks on the main diagonal. Furthermore, the off diagonal blocks are assumed of low quasi-separable dimension, and even such that their rows and columns can be generated in a simple way from the off-diagonal blocks that lay deeper in the hierarchy. This structure is exemplified in the following picture. The indexing scheme goes as follows: the first index indicates the level of hierarchy, the second the position of the columns (for the B matrices) respect. rows (for the C matrices) in the row, respect. column subdivision of the original matrix. It is thus assumed that e.g. $B_{1,2}$ can be generated from $B_{2,1}$ and $B_{2,2}$, more precisely, there are matrices $W_{2,1}$ and $W_{2,2}$ so that

$$B_{1,2} = \begin{bmatrix} B_{2,1}W_{2,1} \\ B_{2,2}W_{2,2} \end{bmatrix}$$

the philosophy being that the lower level (and larger) off diagonal blocks are much more sparse than those

of the deeper levels.

$D_{2,1}$	$B_{2,1}C_{2,2}$	$B_{1,2}C_{1,1}$	
$B_{2,2}C_{2,1}$	$D_{2,2}$		
$B_{1,1}C_{1,2}$		$D_{2,3}$	$B_{2,3}C_{2,4}$
		$B_{2,4}C_{2,3}$	$D_{2,4}$

It turns out that such HSS systems can be solved in an especially efficient way. A recent survey of the method, with many examples, can be found in the thesis of T. Pals (UC Santa Barbara). This type of structure has an independent interest in another context. In [1] it has been shown that an invariant multi-resolution system is in fact equivalent to a time-varying quasi-separable system.

Another hierarchical structure (type II) is one that originates easily in systems that satisfy partial differential equations, e.g. in two or three dimensions. In the next section we discuss a nice example in signal processing to some extent. For ease of discussion, let us consider a prototype case here, originary from optical flow analysis, known as the Horn-Chunck algorithm (see further). We assume that the operator C has the following hierarchical structure: at the top level of the hierarchy, C is a matrix of block dimension $N \times N$ such that the blocks $C_{i,j} = 0$ for $|i - j| > 1$, in other words, it is block triangular, e.g. as in:

$$C = \begin{bmatrix} M & -U & & & \\ -U & M & -U & & \\ & -U & M & -U & \\ & & -U & \ddots & \ddots \\ & & & \ddots & \ddots & -U \\ & & & & -U & M \end{bmatrix} \quad (5)$$

The matrices M and U are defined as

$$M = \begin{bmatrix} 1 & -\frac{1}{6} & & & \\ -\frac{1}{6} & 1 & -\frac{1}{6} & & \\ & -\frac{1}{6} & \ddots & \ddots & \\ & & \ddots & \ddots & \\ & & & \ddots & \ddots \end{bmatrix}, \quad U = \frac{1}{12} \begin{bmatrix} 2 & 1 & & & \\ 1 & 2 & 1 & & \\ & 1 & \ddots & \ddots & \\ & & \ddots & \ddots & \\ & & & \ddots & \ddots \end{bmatrix} \quad (6)$$

Each of the non-zero blocks in the original matrix has the same structure, and that further down one (or more) levels. Such a system does not merit the qualification 'quasi-separable', although the lowest level blocks are quasi-separable of order 3, and the next level has a block-quasi-separable structure of order 3, the overall quasi-separable rank at that next level is actually $N + 3$ - although the system is extremely sparse, the order of sparsity per row (or column) being just 9, a constant. One level higher things really get out of hand, the quasi-separable order becomes now of $O(N^2)$, in fact, it stays just one order less than the overall size of the matrices at that level. Can such a complex system be solved efficiently? The approach we propose is to lower the level of quasi-separability every time one makes a move to the next level of the hierarchy (in a LU or QR factorization) through model order reduction. The big question is whether that approach does work indeed, and results in a numerical accurate procedure. In the next section we show that that is indeed the case in a major application for stochastic signal analysis, namely optical flow analysis.

3. APPLICATION TO OPTICAL FLOW ANALYSIS

'Optical flow' is informally defined as the minimal transformation needed to map a picture on another one. It is most useful in analyzing video streams, because then the transformation of one frame into the next one is presumably a transformation of low complexity and it contains relevant information on the sequence of transformations. In fact, it has been shown that from it most of the relevant 'semantic information' can be deduced, for a quick survey we refer to [5]. While motion estimation in current standards is based on block matching, flow analysis is able to produce information on many other important aspects of the image sequence in a video stream, namely aspects relating to inter-

polation, restauration, the determination of the focus point, the identification of image content etc... For those types of applications the optic flow computation according to e.g. Horn and Schunk [8] is an attractive and effective alternative. The basic assumption underlying all methods of motion estimation is that changes in the image intensity originate from local translations of the pixels, i.e. motion of objects, but not from changes in lighting or from object occlusions or innovation. This leads to what is known as the 'optical flow constraint'. This constraint is then augmented with equations aiming at overcoming its indeterminacy. Horn and Schunk assume that the motion of pixels in a sequence (characterized by pixel dependent 2D motion vectors with components $\{v_x, v_y\}$) are varying in a smooth way. This is expressed by stating that the sum of squares of the Laplacian of the motion fields v_x and v_y has to be minimized. Hence, the optic flow problem becomes an optimization problem. To summarize a long story, the resulting equations have the following form:

$$\left(\alpha^2 \begin{bmatrix} C & 0 \\ 0 & C \end{bmatrix} + \begin{bmatrix} I_1 \\ I_2 \end{bmatrix} [I_1 \ I_2] \right) \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} I_1 \\ I_2 \end{bmatrix} I_t \quad (7)$$

where:

- C is a positive definite, block tridiagonal matrix with blocks that are themselves tridiagonal, see its definition earlier.

- In addition, C is Toeplitz block Toeplitz with fixed (data independent) entries. It represents the negative of the Laplace operator (we give details in the next paragraph);

- α is a scalar constant;

- I_k are (stochastic) diagonal matrices with light intensities as its entries;

- the unknowns x_k are the sought after components of the flow vector;

- I_t is also a (stochastic) matrix with intensities.

Hence, the optical flow problem reduces to solving a system of hierarchical quasi-separability type, the type II case discussed above. Because of the hierarchical nature, the system is not quasi-separable in the strict and the classical URV type algorithm derived for it [13, 4] will not produce a satisfactory solution, it has to be augmented with a model reduction procedure. How this is done is presented in [3]. There, arguments are given why the model order reduction procedure will

give good results in this case. The arguments run as follows:

1. The Toeplitz block Toeplitz case of finite dimensions can be thought as being embedded in an infinite dimensional case. As far as the inversion procedure is concerned, this will only affect the image borders, due to the fact that the infinite dimensional system and its inverse are stable systems. Once the systems have infinite Toeplitz representations, they can be Fourier transformed. For the block infinite dimensional system, the top Fourier representation is

$$C = -Ue^{-i\theta} + M - Ue^{i\theta}$$

in which U and M are themselves Toeplitz operators. Proceeding to a spectral factorization $C = LL^*$ we find $L = m^* - Um^{-1}e^{-i\theta}$ with $\mu = m^*m$ a solution of the Riccati equation

$$\mu = M - U\mu^{-1}U.$$

2. The inverse of this hierarchical multidimensional system can be explicitly computed:

$$L^{-*} = m^{-1} + \mu^{-1}Um^{-1}e^{i\theta} + (\mu^{-1}U)^2m^{-1}e^{2i\theta} + \dots$$

3. The computation hence requires the solution of a Riccati equation in the transform domain. This can be explicitly computed. Let $U = u^T u$ (here U happens to be positive definite), and let $\nu = u^{-T}\mu u^{-1}$, $M_u = u^{-T}Mu^{-1}$, then

$$\nu + \nu^{-1} = M_u.$$

Moving to the transform domain (we are handling Toeplitz operators!), we find

$$\tilde{\nu}(\theta) + \tilde{\nu}(\theta)^{-1} = \tilde{M}_u(\theta)$$

where the tildes are now just functions of θ , and

$$\tilde{\nu} = \frac{1}{2} \left[\tilde{M}_u + \sqrt{\tilde{M}_u^2 - 4} \right].$$

4. As the resulting spectrum has become transcendental, with a discontinuity in the first derivative at the origin, there is no exact, finite dimensional state space representation for it that would

correspond to a quasi-separable representation. Hence, the result has to be approximated with a rational (positive definite) spectrum. Such approximations can indeed be explicitly computed and high accuracies can be obtained with low order approximations (linear in the order, each additional order gains one order of magnitude in the approximation);

5. In the non-infinite case, the Riccati equation is replaced by a recursion of the Riccati type,

$$\begin{cases} m_i^* m_i &= \mu_i \\ \mu_{i+1} &= M - U \mu_i^{-1} U \end{cases}$$

whose fixed point is the solution to the infinite case.

As the recursion turns out to be extremely stable (due to the specific values), it is a reasonable assumption that an order of approximation equal to what works in the infinite case would be acceptable. This conclusion is confirmed numerically for the Horn-Chunck case considered here.

4. OTHER APPLICATIONS AND THE CONNECTION WITH KALMAN FILTERING

There are quite a few more applications in which quasi-separable systems appear, some of them are related to stochastic signal processing [12], others to optimal control (as a dual problem to optimal filtering) and to the modeling of systems governed by (partial) differential or integral equations such as the problem of inverse scattering in Electromagnetism or the modeling of (leaky) interconnects in modern VLSI systems. The main workhorse in solving quasi-separable systems is a recursive version of the QR-algorithm known as the 'square root algorithm', which can be viewed as a generalization of the Kalman algorithm in its square root version [10]. Expanding a little on these ideas, a number of observations are in order:

1. The factorization $C = LL^*$ already considered earlier for the Horn-Chunck case would lead to the square-root algorithm for Kalman filtering, if C turns out to be a covariance matrix resulting

from a process described by state space equations of the form

$$\begin{cases} x_{n+1} &= Fx_n + Gu_n \\ y_n &= Hx_n + w_n \end{cases}$$

in which u, w stand for white noise and x_n is an internal state variable. In the Horn-Chunck case the C matrix plays a rather special role and the noise comes in a different way than in the Kalman case, but the underlying matrix factorization ideas are similar.

2. The square root algorithm is an embodiment of some fundamental properties of dynamical systems. Suppose that T is a causal (time-varying) transfer operator. Then we know that there is a factorization $T = UT_o$ in which U is a causal isometric operator and T_o is causally right invertible. U captures the 'zero's' of the system, or, to put it differently, the (anticausal) dynamics of an inverse to T . The square-root algorithm precisely produces this characterization of T . We have to refer to the literature for the significance of this fact, let us suffice to state here that it is the expression in our context of the celebrated Beurling-Lax theorem of Hardy space theory.
3. Besides the square root algorithm, model reduction theory plays an important role in keeping computational complexity down. Unfortunately, quasi-separability is not hierarchically stable, which means that a higher level quasi-separable structure does not translate down the hierarchy. Model reduction is the technique allowing to restrict complexity at each level of the computations. The Horn-Schunck example shows that the technique is very promising. Because of its recent discovery, it has not been used very much yet, but it holds the promise that we soon will be able to come up with new and powerful pre-conditioners allowing us to solve ever more ambitious, multi-dimensional systems of partial differential equations steered with stochastic quantities.

5. REFERENCES

- [1] D. Alpay and D. Volok. Point evaluation and hardy space on a homogeneous tree. *Integral Equations and Operator Theory*, to appear, 2005.
- [2] S. Chandrasekaran, M Gu, and T. Pals. A fast and stable solver for smooth recursively semi-separable systems. In *SIAM Annual Conference, San Diego and SIAM Conference of Linear Algebra in Controls, Signals and Systems, Boston*, 2001.
- [3] P. Dewilde, K. Diepold, and W. Bamberger. A semi-separable approach to a tridiagonal hierarchy of matrices with application to image flow analysis. In *Proceedings MTNS*, 2004.
- [4] P. Dewilde and A.-J. van der Veen. *Time-varying Systems and Computations*. Kluwer, 1998.
- [5] K. Diepold, P. Dewilde, and W. Bamberger. Optic flow computation and time-varying system theory. In *Proceedings MTNS*, 2004.
- [6] I. Gohberg, T. Kailath, and I. Koltracht. Linear complexity algorithms for semiseparable matrices. *Integral Equations and Operator Theory*, 8:780–804, 1985.
- [7] B.L. Ho and R.E. Kalman. Effective construction of linear, state-variable models from input/output functions. *Regelungstechnik*, 14:545–548, 1966.
- [8] B.K.P. Horn and B.G. Schunk. Determining optical flow. *Artificial Intelligence*, 2:185–203, 1981.
- [9] L. Kronecker. Algebraische Reduction der schaaren bilinearer Formen. *S.B. Akad. Berlin*, pages 663–776, 1890.
- [10] M. Morf, M. Dobbins, J. Friedlander, and T. Kailath. Square-root algorithms for parallel processing in optimal estimation. *Automatica*, vol. 15:299–306, 1979.
- [11] L.M. Silverman and H.E. Meadows. Equivalence and synthesis of time-variable linear systems. In *Proc. 4-th Allerton Conf. Circuit and Systems Theory*, pages 776–784, 1966.
- [12] L. Tong, A.-J. van der Veen, and P. Dewilde. A new decorrelating rake receiver for long-code wcdma. In *Proceedings 2002 Conference on Information Sciences Systems*. Princeton University, March 2002.
- [13] A.J. van der Veen. Computation of the inner-outer factorization for time-varying systems. In P. Dewilde e.a., editor, *Challenges of a Generalized System Theory*, Essays of the Royal Dutch Academy of Sciences, pages 99–117, Amsterdam, The Netherlands, 1993. North-Holland.