SUPPLEMENT MATERIAL TO

STOCHASTIC MODELING OF STATIONARY SCALAR GAUSSIAN PROCESSES IN CONTINUOUS TIME FROM AUTOCORRELATION DATA*

Documentation of the corresponding Matlab implementation

MARTIN HANKE†

Last modified. March 28, 2023

1. General comments. The zip-file `stochmodel_aaa.zip` opens a directory `stochmodel_aaa` with the following files:

`readme.pdf`	documentation file
`stochmodel_aaa.m`	main script file

`aaa_real.m`
`constr_lsq.m`
`solve_sing_Lure_eq.m`
	auxiliary subroutines

`jung_data.txt`	sample data file from [4]
`shea_data_medium.txt`	sample data file from [2]
`load_jung_data.m` `load_shea_data.m`	subroutines to access the data files

An additional file `Riccati.m` is needed, with a Matlab function `X = Riccati(B,C,G)`, which computes the solution X of the algebraic Riccati matrix equation

$$B^T X + XB - XCX + G = 0\,,$$

where $-C$ and G are real symmetric and positive semidefinite matrices. An appropriate file is available on the MathWorks file exchange server [5].

The two data files provide the input data used in [3, Section 5]. To execute the algorithm simply type `stochmodel_aaa`. With the original files this employs the setting of Example 5.1. Upon successful termination of the algorithm the variables `A` and `g` contain the parameters of the approximating Markov system (2.1) of [3], and `Sigma` is the covariance matrix (2.5) of the associated stochastic process, which is needed for the initial value (7.4) of the system. The arrays `yy` and `phi` contain the values of the autocorrelation function and its approximating Prony series for the time abscissa given in `tt`. Further diagnostic information on the performance of the algorithm can be found in the output file `stochmodel_aaa.out`.

The script uses the same variable identifiers as the paper [3], and most of the code is well explained by the exposition in [3]. The program is organized in eight steps. Step 1 is concerned with the input of a data set and initialization of the basic parameters. Steps 5 and 7 of the code may need additional explanations; those are provided below.

*The research leading to this work has been done within the Collaborative Research Center TRR 146; corresponding funding by the DFG is gratefully acknowledged.

†Institut für Mathematik, Johannes Gutenberg-Universität Mainz, 55099 Mainz, Germany (`hanke@math.uni-mainz.de`)

2. Step 5: Check positivity of $\widehat{\varphi}$ (to be sure that the model is passive). Recall from (4.1) of the paper that

$$\widehat{\varphi}(\xi) \;=\; -2\sum_{j=1}^{m} \frac{\alpha_j \lambda_j}{\lambda_j^2 + \xi^2}$$

is an even function of ξ and a rational function of degree m in the variable $s = \xi^2$. If λ_{r+1} and λ_{r+2} are a pair of complex conjugate exponents, then the weights α_{r+1} and α_{r+2} are also complex conjugates of each other, and the corresponding two terms of the sum add up to

$$\frac{\alpha_{r+1}\lambda_{r+1}}{\lambda_{r+1}^2 + s} + \frac{\alpha_{r+2}\lambda_{r+2}}{\lambda_{r+2}^2 + s} \;=\; \frac{2|\lambda_{r+1}|^2 \operatorname{Re}(\alpha_{r+1}\overline{\lambda}_{r+1}) \;+\; 2\operatorname{Re}(\alpha_{r+1}\lambda_{r+1})\, s}{|\lambda_{r+1}|^4 + 2(\operatorname{Re}\lambda_{r+1}^2)s + s^2}\,.$$

The algorithm proceeds by subsequently forming the sum of all m fractions, starting with the real ones, and then turning to each pair of complex conjugate ones. Each fraction is represented by its denominator and numerator polynomials in s, using Matlab arrays for their coefficients in the usual way. Accordingly, products of polynomials are computed by convolving the corresponding arrays.

The zeros of the final numerator polynomial determines the roots of $\widehat{\varphi}$. In view of the last formula displayed in [3, Section 7], this polynomial in s has a degree of at most $m - k_* - 1$, hence this is the (maximal) number of zeros to be computed. Since $s = \xi^2$ each of these zeros corresponds to two zeros of $\widehat{\varphi}$, respectively.

3. Step 7: Variable transformation to replace $\boldsymbol{v}$, $\boldsymbol{w}$, and $\boldsymbol{J}$ by $\boldsymbol{e_1}$ and $\boldsymbol{A}$. In the script `stochmodel_aaa.m` the construction of the matrix V of Eq. (6.3) in [3] differs from the one by Bai and Freund [1, Lemma 1], referred to in [3]. Instead the matrix V is specified as follows.

For $v, w \in \mathbb{R}^m$ with $v^T w = \sigma^2 > 0$, compare [3, Section 6], let

$$u \;=\; \frac{1}{\|w\|} w \;+\; e_1\,.$$

Then the Householder transformation

$$P \;=\; I \;-\; \frac{2}{\|u\|^2}\, uu^T$$

satisfies

$$Pw \;=\; -\|w\|\, e_1\,.$$

Accordingly,

$$V \;=\; P \;+\; e_1\Big(v + \frac{1}{\|w\|} w\Big)^T$$

maps w onto

$$Vw \;=\; Pw \;+\; \big(v^T w + \|w\|\big)e_1 \;=\; (v^T w)e_1 \;=\; \sigma^2 e_1\,.$$

Further, since P is orthogonal,

$$P^T e_1 \;=\; P^{-1} e_1 \;=\; -\frac{1}{\|w\|} w\,,$$

and hence,

$$V^T e_1 \;=\; P^T e_1 \;+\; \Big(v + \frac{1}{\|w\|} w\Big) e_1^T e_1 \;=\; -\frac{1}{\|w\|} w \;+\; \Big(v + \frac{1}{\|w\|} w\Big) \;=\; v\,.$$

Finally, the matrix V is nonsingular. For, if $Vx = 0$ then the definition of V implies

$$Px \;=\; \gamma e_1 \qquad \text{with} \qquad \gamma \;=\; -\Big(v + \frac{1}{\|w\|} w\Big)^T x\,,$$

i.e.,

$$x \;=\; \gamma P^T e_1 \;=\; -\frac{\gamma}{\|w\|}\, w\,.$$

It follows that

$$\gamma \;=\; -\Big(v + \frac{1}{\|w\|} w\Big)^T x \;=\; \frac{\gamma}{\|w\|} \Big(v + \frac{1}{\|w\|} w\Big)^T w \;=\; \gamma\Big(\frac{v^T w}{\|w\|} + 1\Big)\,,$$

and since $v^T w = \sigma^2 > 0$, this is only possible for $\gamma = 0$; however, $\gamma = 0$ implies $x = 0$, and hence, the null space of V is trivial.

We thus have established that the matrix V constructed above satisfies the two identities

$$Vw \;=\; \sigma^2 e_1 \qquad \text{and} \qquad V^{-T} v \;=\; e_1$$

required in [3, (6.3)].

4. Licence. Redistribution and use of these files in source or in binary form, with or without modification, are permitted provided that the following conditions are met:

- Reference is given to [3].
- If the data `jung_data.txt` (Examples 5.1 or 5.2 in [3]) are used, reference is given to [4].
- If the data `shea_data_medium.txt` (Example 5.3 in [3]) are used, reference is given to [2].
- Redistributions of source code or data must retain this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

Acknowledgement. I am indebted to Gerhard Jung and Jeanine Shea for granting permission to include their data sets into this distribution.

REFERENCES

1. Z. Bai and R.W. Freund, *Eigenvalue-based characterization and test for positive realness of scalar transfer functions*, IEEE Trans. Autom. Control **45** (2000), pp. 2396–2402.
2. N. Bockius, J. Shea, G. Jung, F. Schmid, and M. Hanke, *Model reduction techniques for the computation of extended Markov parameterizations for generalized Langevin equations*, J. Phys.: Condens. Matter **33** (2021), 214003.
3. M. Hanke, *Stochastic modeling of stationary scalar Gaussian processes in continuous time from autocorrelation data*, manuscript, Johannes Gutenberg-Universität Mainz, 2023, submitted.
4. G. Jung, M. Hanke, and F. Schmid, *Iterative reconstruction of memory kernels*, J. Chem. Theor. Comput. **13** (2017), pp. 2481–2488.
5. `https://de.mathworks.com/matlabcentral/fileexchange/36263-algebraic-riccati-equation-solver`, Copyright (c) 2012, Emmett; downloaded on September 2022.