The αSGN(m) MATLAB toolbox

Ahmed Mahmood (July 2018)

Brief Intro.

The aim of this toolbox is to incorporate emerging techniques related to α-stable processes within Matlab's functionality. Our topic of interest is the αSGN(m) model. As of now, literature on the topic can be found primarily at https://arl.nus.edu.sg/twiki6/bin/view/ARL/Publications and consists of (1)--(6) amongst others.

This document is offers a quick overview of the toolbox. For more details, revert to the actual scripts. Before listing down the functions within this toolbox, we introduce a few concepts to aid the discussion:

The heavy-tailed αSGN(m) process is derived from an $(m+1)$-dimensional α-sub-Gaussian (αSG) distribution, which in turn is parameterized by the characteristic exponent $\alpha \in (0,2)$ and the covariance matrix $\mathbf{R}$. The αSG distribution is elliptic and can be denoted by αSG($\alpha, \mathbf{R}$) or equivalently by αSG($\alpha, \delta, \mathbf{Cov}$), where $\mathbf{Cov}$ is the normalized covariance matrix and $\delta \in (0, \infty)$ is the scale, i.e., $\mathbf{R} = \delta^2 \mathbf{Cov}$. The αSGN(m) process essentially constrains any of its $(m+1)$ adjacent samples to follow the aforementioned αSG distribution. This ensures $\mathbf{R}$ (and thus $\mathbf{Cov}$) is a symmetric Toeplitz matrix. We note that each sample of an αSGN(m) process is a symmetric α-stable (SαS) random variable. The latter's distribution is completely characterized by the tuple (α, δ) and can be expressed as $\mathcal{S}(\alpha, \delta)$.

For more information about the employed parameterization, revert to (3). For a great introduction to αSG distributions, do go through Nolan's discussion on the topic at http://fs2.american.edu/jpnolan/www/stable/EllipticalStable.pdf.

Functions

Add all directories to Matlab's path before use.

Random variates

- `X = asgn(alpha, R, N, _')`: Generates `N` samples of αSGN(m) with underlying distribution αSG(`alpha, R`). The function accepts optional inputs as well (highlighted by `_`).

- `x = stabrnd(alpha, beta, delta, mu, N, M)`: McCulloch's original script that returns independent outcomes of the stable distribution $\mathcal{S}$(`alpha, beta, delta, mu`), where `beta` is the skew parameter and `mu` is the location. Note that $\mathcal{S}$(`alpha, 0, delta, 0`) is statistically equivalent to $\mathcal{S}$(`alpha, delta`). The outcomes are returned as the `N` $\times$ `M` matrix `x`.

Fitting

- `[alpha, mu, delta] = sstabfit(x)`: Mandar Chitre's script that fits an SαS distribution to the data vector `x`.

- `[alpha, delta, Cov] = asgnfit(X, m, _)`: Estimates αSGN(m) parameters from the data vector `X` for a given `m`. Also, `_` signifies optional inputs.

- `[pxx, pasg, f] = asgnsd(X, fs)`: Determines `pxx`, the normalized non-parameterized spectral density, for the data vector `X` sampled at `fs` kHz. The associated frequency vector `f` (in kHz) is also returned. Moreover, `pasg` is the closed-form *parametric* spectral density of `X` evaluated under the assumption that `X` holds consecutive samples of αSGN(m). Comparing `pxx` to `pasg` allows discerning a suitable m to characterize `X` within the αSGN(m) framework.

Computing PDFs

- `logf_XN = asgnpdf(X, alpha, R)`: Samples the joint-pdf and returns its logarithm for each column (outcome) of `X`, under the assumption that the columns hold consecutive samples of αSGN(m) with underlying distribution αSG(`alpha, R`).
- `f_X = saspdf(X, alpha, delta)`: Samples and returns the pdf associated with $\mathcal{S}(\texttt{alpha}, \texttt{delta})$ at each element (outcome) in the matrix `X`. Note that `f_X` is the same size of `X`.

Storing and retrieving αSGN(m) data

As αSGN(m) is a GARCH process, samples have to be sequentially computed. Consequently, `asgn()` can become a potential bottleneck for computation time when running intensive performance tests/simulations. To circumvent this, realizations may be pre-computed and stored offline, only to be retrieved for later use. The following functions are helpful in this regard:

- `asgn_write(alpha, Cov, FileSize, NumFiles, fs)`: Computes and stores a realization of αSGN(m) with underlying distribution αSG($\texttt{alpha}, 1, \texttt{Cov}$)= αSG($\texttt{alpha}, \texttt{Cov}$). The samples are stored as `NumFiles` files, each of which is sized `FileSize` MBs. Also, `fs` is the associated sampling frequency. The files are stored in a sub-directory of the function's root folder. As an example, if $\texttt{alpha} = 1.57$ and $m =$4 (implying `Cov` is a 5×5 matrix), the files are saved in "\a1_57__m_4\" and are labeled as "asgn_1.bin", "asgn_2.bin" and so on.

- `[x, _] = asgn_read(fpath, samps, _)`: Reads `samps` samples from the files written by `asgn_write()`. The absolute path directory of these files is passed as the input string `fpath`.

Miscellaneous

- `vr_validate`: refer to the script for details.

References

1. A. Mahmood and M. Chitre, *Temporal Analysis of Stationary Markov α-sub-Gaussian Noise,* in OCEANS 2016 MTS/IEEE, (Monterey, CA, USA), September 2016.
2. A. Mahmood, M. Chitre, and V. Hari, *Locally optimal inspired detection in snapping shrimp noise,* IEEE Journal of Oceanic Engineering, vol. 42, pp. 1049--1062, October 2017.

3. A. Mahmood and M. Chitre, *Generating Random Variates for Stable Sub-Gaussian Processes with Memory,* Signal Processing, vol. 131, pp. 271--279, February 2017.
4. A. Mahmood and M. Chitre, *Optimal and Near-Optimal Detection in Bursty Impulsive Noise,* IEEE Journal of Oceanic Engineering, vol. 42, pp. 639--653, October 2016.
5. A. Mahmood and M. Chitre, *Uncoded Acoustic Communication in Shallow Waters with Bursty Impulsive Noise,* in Underwater Communications Networking (Ucomms 2016), (Lerici, Italy), September 2016. (Invited).
6. A. Mahmood, V. Hari, and M. Chitre, *Model-Based Signal Detection in Snapping Shrimp Noise,* in Underwater Communications Networking (Ucomms 2016), (Lerici, Italy), September 2016. (Invited).