Program in Applied Mathematics

Spectral Factorization and Spectral Estimation using Kalman Filtering

Jared McBride

Applied Mathematics
University of Arizona

October 15, 2021

THE UNIVERSITY OF ARIZONA

Outline

1. Definitions
2. Estimation
   - Examples
3. Spectral Factorization
   - Examples
4. Sperctral Estimation (reprisal)
   - Whitening
5. Conclusions

Stochastic process: A family of random variables indexed by an index set (discrete or continuous). E.g. $X : \Omega \times \mathbb{R} \to \mathbb{C}^n$ or $X : \Omega \times \mathbb{Z} \to \mathbb{C}^n$

Timeseries: A (deterministic) realization of a stochastic process (discrete or continuous) that is indexed by time. In this talk this is indexed over a finite set. E.g. $x : \{1, 2, \ldots, N\} \to \mathbb{C}^n$

Stationary stochastic process: (Sometimes called wide-sense stationary) A stochastic process satisfying the following conditions:

$$\mathbb{E}X_t = \mu \qquad \text{(no dependence on } t\text{)}$$

$$\mathbb{E}[(X_t - \mu)(X_s - \mu)^*] = R_X(t - s) \qquad \text{(depends only on difference } t - s\text{)}$$

Where the asterisks * denote the conjugate transpose.

Stationary timeseries: (Sometimes called wide-sense stationary) A timeseries realization of a stationary stochastic process.

(These may be vector-valued. Timeseries of this type is often referred to as *multiple timeseries*)

Definitions (naive systems analysis)

Signal: A stochastic process or a timeseries.
System: An operator. A function from a Hilbert space to a Hilbert space.
Linear time-invariant system: A linear time-invariant operator. It can be show that in the contexts of Hilbert spaces these can be represented as integral operators with kernel of one variable. So

$$\mathcal{L} : L^2 \to L^2 \qquad \text{is LTI} \qquad \mathcal{L}x(t) = \int_{-\infty}^{\infty} x(s)h(t-s)ds$$

or in discrete time

$$L : \ell^2 \to \ell^2 \qquad \text{is LTI} \qquad Lx_n = \sum_{k=-\infty}^{\infty} x_k h(n-k).$$

Notice this is just a convolution.
z-series: Given a sequence a (bilaterally infinite) the z-series is the complex function

$$\hat{a}(z) = \sum_{k=-\infty}^{\infty} a_k z^{-k}.$$

Impulse response of a system: The output of the system when the impulse signal $\delta = (\ldots, 0, 1, 0, \ldots)$ is the input. Notice that this recovers the kernel.

$$L\delta_n = \sum_{k=-\infty}^{\infty} \delta_k h(n-k) = h_n$$

Transfer function of an LTI system: The z-series of the impulse response of a system

$$\hat{h}(z) = \sum_{k=-\infty}^{\infty} h_k z^{-k}.$$

Covariance function: (Sometimes Covariance sequence in discrete time)
Given a stochastic process X it is the function

$$R_X(t,s) = \mathbb{E}[(X_t - \mu)(X_s - \mu)^*]$$

(we are exclusively concerned with discrete-time stationary processes so we have)

$$R_X(n,m) = \mathbb{E}[(X_n - \mu)(X_n - \mu)^*] = R_X(n-m) \qquad n, m \in \mathbb{Z}$$

Observe, in the vector case this is matrix valued.
Power spectrum: The Fourier series of the covariance sequence

$$S_X(\omega) = \sum_{k=-\infty}^{\infty} R_X(k) e^{-ik\omega}$$

z-spectrum: The z-series of the covariance sequence

$$\bar{S}_X(z) = \sum_{k=-\infty}^{\infty} R_X(k) z^{-k}$$

Convolution: given two processes X and Y,

$$(Y \star X)_n = (X \star Y)_n = \sum_{k=-\infty}^{\infty} Y_k X_{n-k}$$

Convolution theorem: The z-series of a convolution is the product of the z-series

$$\widehat{(a \star b)_n}(z) = \hat{a}(z)\hat{b}(z)$$

Spectrum of convolution: Suppose Y is stationary stochastic (disctrete-time) process and $r \in \ell^1$, then

$$S_{r\star Y} = \hat{r}(z) S_Y(z) \hat{r}^*(z^{-*})$$

Mean and Covariance Estimation

We are given data, x_n, for $n = 1, 2, 3, \ldots, N$

- Assume it is be a realization of the discrete-time process X_n or observations of a continuous time process X_{t_n}.
- Assume the process X_n is stationary

How do we estimate μ?
By virtue of stationary

$$\mu = \mathbb{E}X_n \approx \frac{1}{N}\sum_{n=1}^{N} x_n =: \tilde{\mu}$$

How do we estimate $R_X(n)$?
Again, by virtue of stationary

$$R_X(n) = \mathbb{E}[(X_n - \mu)(X_0 - \mu)^*] \approx \frac{1}{N}\sum_{j=1}^{N-n}(x_{n+j} - \tilde{\mu})(x_j - \tilde{\mu})^* =: \tilde{R}_X(n)$$

Spectrum Estimation (sample spectrum)

Program in Applied Mathematics

How do we estimate $S_X(\omega)$? (assume X_n is mean zero)

Peridogram: (direct approach)

$$
\begin{aligned}
\tilde{S}_X(\omega) &= \sum_n \tilde{R}_X(n) e^{-in\omega} \\
&= \sum_n \frac{1}{N} \sum_j x_{n+j} x_j^* e^{-in\omega} \\
&= \frac{1}{N} \sum_k \sum_j x_k x_j^* e^{-ik\omega} e^{ij\omega} \\
&= \frac{1}{N} \left(\sum_k x_k e^{-ik\omega} \right) \left(\sum_j x_j e^{-ij\omega} \right)^* \\
&= \frac{1}{N} \hat{x}(\omega) \hat{x}(\omega)^* = \frac{1}{N} |\hat{x}(\omega)|^2 \ \ (= \texttt{abs2.(fft(x))/N})
\end{aligned}
$$

Asymptotically unbiased but inconsistent (the variance does not vanish as N gets large).

How do else we estimate $S_X(\omega)$?

Bartlett's smoothing procedure: cut up the timeseries into k blocks. And approximate the peridogram $\tilde{S}_X^{(j)}(\omega)$ for each block of data $j = 1, 2, \ldots, k$.

$$\tilde{S}_X(\omega) = \frac{1}{k}\sum_{J=1}^{k}\tilde{S}_X^{(j)}(\omega)$$

This procedure allows us to control the variance, but at the expense of bias. This procedure can be generalized.

General class of smoothed spectral estimators:
Bartlett:

$$\tilde{S}_X(\omega) = \frac{1}{k}\sum_{n=-k}^{k}\left(1 - \frac{|n|}{k}\right)\tilde{R}_X(n)e^{-in\omega}$$

General:

$$\tilde{S}_X(\omega) = \frac{1}{k}\sum_{n=-\infty}^{\infty} w(n)\tilde{R}_X(n)$$

with

$$\begin{array}{ll} (1) & w(0) = 1 \\ (2) & w(n) = w(-n) \\ (3) & w(n) = 0, \quad |n| \geq k, \quad k < N \end{array}$$

w is called a windowing function.

Most common window functions,

Bartlett:

$$w(n) = \begin{cases} 1 - \frac{|n|}{k}, & |n| \leq k \\ 0, & |n| > k \end{cases}$$

Tukey:

$$w(n) = \begin{cases} \frac{1}{2}\left(1 + \cos\frac{\pi n}{k}\right), & |n| \leq k \\ 0, & |n| > k \end{cases}$$

Parzen:

$$w(n) = \begin{cases} 1 - 6\left(\frac{n}{k}\right)^2 + 6\left(\frac{|n|}{k}\right)^3, & |n| \leq k/2 \\ 2\left(1 - \frac{|n|}{k}\right)^3, & k/2 < |n| \leq k \\ 0, & |n| > k \end{cases}$$

Example 1: AR(2) Signal

Let us consider the stationary autoregressive process of order 2,

$$Y_n = (r_1 + r_2)Y_{n-1} - r_1 r_2 Y_{n-2} + U_n, \qquad \text{for } n > -\infty$$

for $r_1, r_2 \in \{z : |z| < 1\}$ and U_n are i.i.d. standard normal random variables.

One way to compute the z-spectrum is as follows. Recognize,

$$(r \star Y)_n = Y_n - (r_1 + r_2)Y_{n-1} + r_1 r_2 Y_{n-2} = U_n,$$

$$r = (\ldots, 0, \boxed{1}, -r_1 + r_2, r_1 r_2, 0, \ldots)$$

So that,

$$1 = S_U(z) = S_{(r\star Y)} = \hat{r}(z) S_Y(z) \hat{r}^*(z^{-*})$$

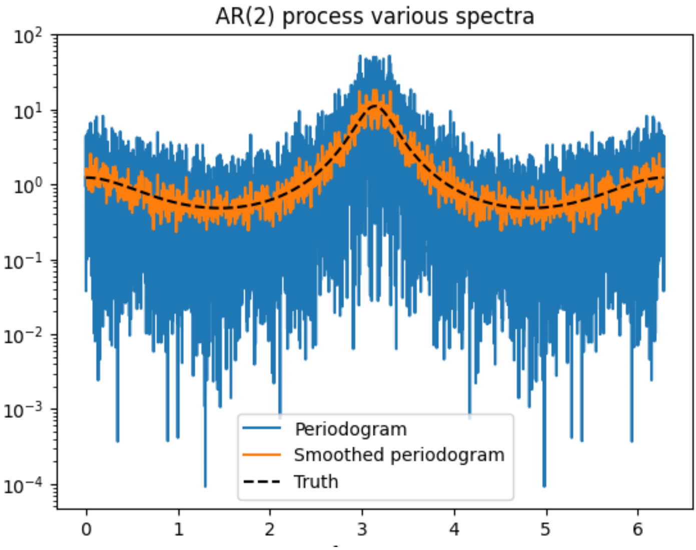
and

$$S_Y(z) = \frac{1}{\hat{r}(z)\hat{r}^*(z^{-*})} = \frac{1}{(1 - r_1 z^{-1})(1 - r_2 z^{-1})(1 - r_1^* z)(1 - r_2^* z)}$$

Example 1: AR(2) Signal

```
using DSP, PyPlot, FFTW
at  = include("../../Tools/AnalysisToolbox.jl")

r1 = .5; r2 = -.8
r = [1, -(r1 + r2), r1*r2]
f(z) = sum(r[j]*z^(1-j) for j=1:3)

N = 10^4
fil = ZeroPoleGain(zeros(0),[r1,r2],1)
y = filt(fil,randn(N))

Sy_per = abs2.(fft(y))/N
Sy_num = at.z_crossspect_dm(y,y;Nex = N)
Sy_ana = map(z -> 1/abs2(f(z)),exp.(2pi*im*(0:N-1)/N))

Θ = 2pi*(0:N-1)/N
title("AR(2) process various spectra")
semilogy(Θ,Sy_per, label = "Periodogram")
semilogy(Θ,Sy_num, label = "Smoothed periodogram")
semilogy(Θ,Sy_ana, "k--", label = "Truth")
xlabel("frequency"); legend()
```

Example 1: AR(2) Signal

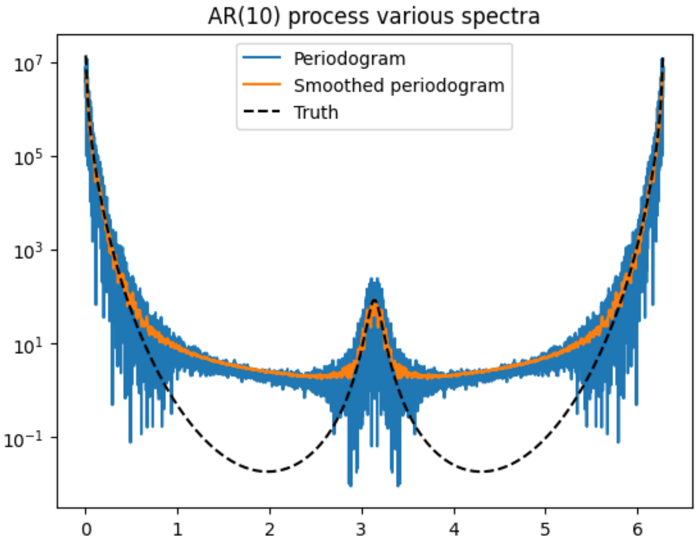


Example 2: AR(10) Signal

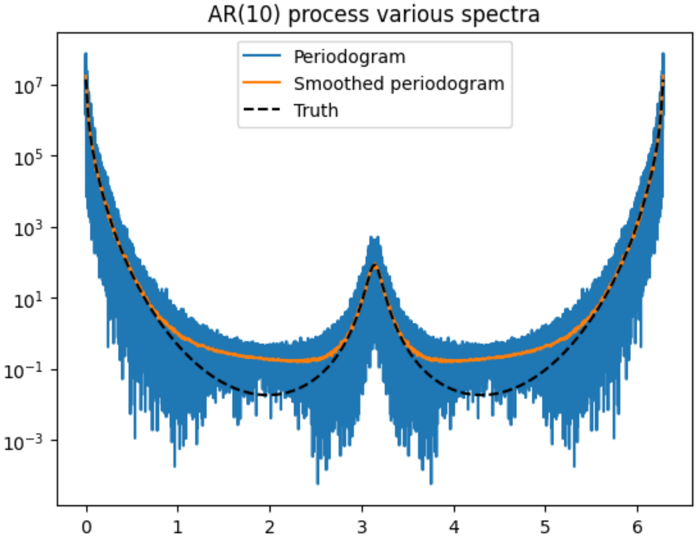
Program in Applied Mathematics

```
using DSP, PyPlot, FFTW, Polynomials
at  = include("../../Tools/AnalysisToolbox.jl")

p = 15
ri = 2*rand(p) .- 1

ri = [0.81, 0.4, 0.28, 0.35, 0.26, -0.93, -0.89, 0.991, 0.84, -0.30]

ff = Polynomial([1])*prod(Polynomial([1, -ri]) for ri in ri)
r = coeffs(ff)

N = 10^4
fil = ZeroPoleGain(zeros(0),ri,1)
y = filt(fil,randn(N))

Sy_per = abs2.(fft(y))/N
Sy_num = at.z_crossspect_dm(y,y;Nex = N)
Sy_ana = map(z -> 1/abs2(ff(z)),exp.(2pi*im*(0:N-1)/N))

Θ = 2pi*(0:N-1)/N
semilogy(Θ,Sy_per, label = "Periodogram")
semilogy(Θ,Sy_num, label = "Smoothed periodogram")
semilogy(Θ,Sy_ana, "k--", label = "Truth")
title("AR(10) process various spectra"); legend()
```

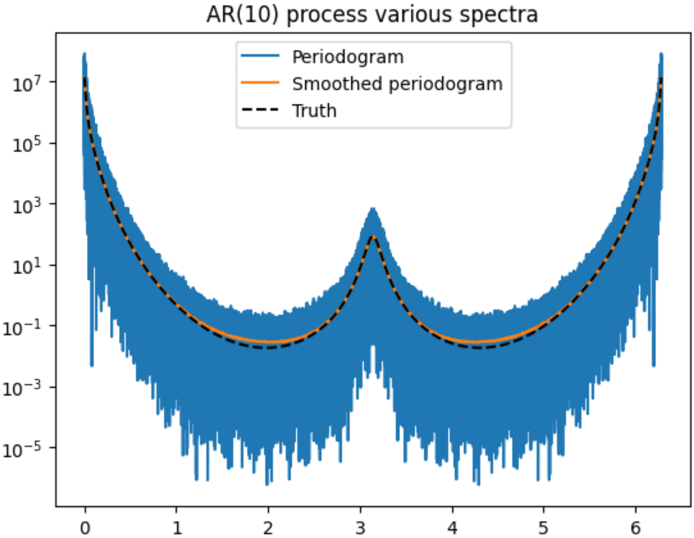
Example 2: AR(10) Signal



Example 2: AR(10) Signal



Example 2: AR(10) Signal



More Definitions

A linear filter: A sequence of deterministic elements $h_{n,k}$, $n, k > -\infty$ that operates on a signal by way of convolution.

$$Y_n = (X \star h_{n,\cdot})_n = \sum_{k=-\infty}^{\infty} h_{n,k} \cdot X_{n-k} = \sum_{k=-\infty}^{\infty} h_{n,n-k} \cdot X_k$$

Time-invariant: A filter is time-invariant when $h_{n,k}$ has no dependence on n. So, $h_{n,k} = h_k$.

Recall, any LTI system can be represented by convolving the input signal with the impulse response of the system.

Alternatively, an LTI system can be characterized by it's transfer function. Then the input-output relation can be described in the z-series (z-transform) domain.

$$\hat{Y}(z) = H(z)\hat{X}(z)$$

Causal: A linear time-invariant filter is causal if it's impulse response is causal which means $h_k = 0$ for $k < 0$.
BIBO Stability: A LTI system is stable if given a bounded input the output of the system is bounded.
Inverse: The inverse of an LTI system maps the output to the input.
Suggested by the alternative characterization above the we have

$$\hat{X}(z) = \frac{1}{H(z)}\hat{Y}(z)$$

A few facts:

- A LTI system is BIBO stable if its impulse response h_k is absolutely summable. It's transfer function $H(z)$ converges on the unit circle.
- If a system is causal (and stable) and has a rational transfer function the poles lie within the unit circle.

Minimum-phase: A linear time-invariant system $H(z)$ is minimum-phase if it and it's inverse $H(z)^{-1}$ are both causal and stable.

This means if $H(z)$ is rational all zeros and poles lie strictly with in the unit circle.

Example:

- The system that takes in a white noise signal and outputs an MA(q) process is causal it's transfer function has no poles, but it is only causally invertable (and therefore minimum-phase) if it's zeros are within the th unit circle.

Standard state-space model: A model of the following form:

$$\begin{cases} X_{i+1} &= F_i X_i + G_i u_i \\ Y_i &= H_i X_i + v_i \end{cases}$$

where $F_i \in C^{n\times n}$, $G_i \in C^{n\times m}$, and $H_i \in C^{p\times n}$ are known matrices, and $u = \{u_i\}$, $v = \{v_i\}$, and X_0 are variables with the following property

$$E\begin{pmatrix} X_0 \\ u_i \\ v_i \end{pmatrix}\begin{pmatrix} X_0 \\ u_j \\ v_j \\ 1 \end{pmatrix}^* = \begin{pmatrix} \Pi_0 & 0 & 0 & 0 \\ 0 & Q_i\delta_{ij} & S_i\delta_{ij} & 0 \\ 0 & S_i^*\delta_{ij} & R_i\delta_{ij} & 0 \end{pmatrix}$$

- Y is the output (or observations).
- X is the state variable.
- u is the process (or plant) noise
- v is the measurement noise.

Kalman filtering: Given a process represented by a standard statespace model, we may wish to estimate various quantities, usually the state variable. Kalman filtering provides a theory for computing (recursively) the least linear mean square estimators given observation $\{y_1, y_2, \ldots, y_i\}$, given observations of Y_i.

The heart of the issue is the computation of the innovations sequence

$$E_i = Y_i - \mathbb{E}[Y_i | Y_{i-1}, Y_{i-2}, \ldots, Y_1]$$

One contribution of Kalman was to devise a recursion to compute the innovations.

The Innovations Recursions [1, p. 317]

Consider the standard statespace model

$$\begin{cases} X_{i+1} &= F_i X_i + G_i u_i \\ Y_i &= H_i X_i + v_i \end{cases} \qquad i \geq 0$$

The innovations process of Y can be recursively computed using the equations

$$E_i = Y_i - H_i \theta_i, \quad \theta_0 = 0, \quad E_0 = Y_0,$$

$$\theta_{i+1} = F_i \theta_i + K_{p,i} E_i, \quad i \geq 0,$$

where $K_{p,i} = (F_i P_i H_i^* + G_i S_i) R_{e,i}^{-1}$, $R_{e,i} = R_i + H_i P_i H_i^*$, and

$$P_{i+1} = F_i P_i F_i^* + G_i Q_i G_i^* - K_{p,i} R_{e,i} K_{p,i}, \qquad P_0 = \Pi_0$$

Here, $P_i = \mathbb{E} \tilde{X}_i \tilde{X}_i^*$ where $\tilde{X}_i = X_i - \theta_i$. When $m \ll n$, $p \ll n$ to go from E_i to E_{i+1} requires $O(n^3)$ operations.

Kalman Filtering by Chadrasekhar-Kailath-Morf-Sidhu (CKMS)

It turns out that for time-invariant (constant) parameters ($F_i = F$, $H_i = H$, $G_i = G$, $Q_i = Q$, $R_i = R$, and $S_i = S$) a modified set of recursions wil achieve the same as the Kalman recursions but with significantly less effort.

The key idea is that though P_i is full rank $\delta P_i := P_{i+1} - P_i$ can have low rank.

(since the difference of Hermitian matrices is Hermitian) write

$$\delta P_i = L_i M_i L_i^*$$

rewrite recursions in in terms of L_i and M_i.

The Fast (CKMS) Kalman Recursions [1, p. 409]

The $K_{p,i}$ and $R_{e,i}$ from the Kalman recursion above can be recursively computed by the following set of coupled recursions, for $i \geq 0$

$$\begin{aligned} K_{p,i+1} &= K_{p,i} - FL_i R_{r,i}^{-1} L_i^* H^* \\ L_{i+1} &= FL_i - K_{p,i} R_{e,i}^{-1} HL_i \\ R_{e,i+1} &= R_{e,i} - HL_i R_{r,i}^{-1} L_i^* H^* \\ K_{p,i+1} &= K_{p,i} - L_i^* H^* R_{e,i}^{-1} HL_i \end{aligned}$$

The recursion is initialized as follows: $K_{p,0} = F\Pi_0 H^* + GS$ and $R_{e,0} = R + H\Pi_0 H^*$. Then factor get L_0 and $R_{r,0}$

$$\delta P_0 := F\Pi_0 F^* + GQG^* - K_{p,0} R_{e,0}^{-1} K_{p,0}^* - \Pi_0 =: -L_0 R_{r,0}^{-1} L_0^*$$

where L_0 is $n \times \alpha$ and $R_{r,0}$ is $\alpha \times \alpha$, nonsingular and Hermitian.

Why factor the spectrum?

Wiener's Matrix Spectral Factorization Theorem

If $S : \mathbb{C} \to \mathbb{C}^{d\times d}$, satisfies,

- $S \in L^1(\partial\mathbb{D})$,
- $\log\det S \in L^1(\partial\mathbb{D})$, and
- $S(z) > 0$ (positive definite) for (almost all) $z \in \partial\mathbb{D}$.

Then there exists matrix functions $S^+(z)$ and $S^-(z)$, such that $S^-(z) = S^{+*}(z^{-*})$ and

$$S(z) = S^+(z)S^-(z) \qquad \text{for } z \in \partial\mathbb{D}.$$

Furthermore, S^+ is is an outer analytic matrix function from the Hardy space H_2.

Program in Applied Mathematics

More useful version of Spectral Factorization Theorem

If $\mathbf{y}$ is a mean zero, stationary, discrete time stochastic d-vector-valued process that admits a rational z-spectrum $S_{\mathbf{y}}$ analytic on some annulus containing the unit circle, and

$$S_{\mathbf{y}} > 0 \qquad \text{everywhere on } \partial\mathbb{D}.$$

Then there exists matrix functions $S^{+}(z)$ and $S^{-}(z)$, such

- $S^{+}(z)$ is a $d \times d$ rational matrix function that is analytic on and inside the unit circle,
- $S^{+^{-1}}(z)$ is analytic on and inside the unit circle.
- $S^{-}(z) = S^{+*}(z^{-*})$ and
- $S(z) = S^{+}(z)S^{-}(z)$.

Spectral Factorization (Numerical)

Program in Applied Mathematics

Most Numerical algorithms assume $S(z)$ is rational and has the form of a Laurent Polynomial meaning it may be written as

$$S(z) = \sum_{n=-m}^{m} c_n z^{-n} \qquad \text{with } c_n = c_{-n}^*.$$

If this is assumed it may be shown that

$$S^+(z) = \sum_{n=1} L_n z^n \qquad \text{and} \qquad S^-(z) = \sum_{n=1} L_n^* z^{-n}$$

(this is what we assume here) Algorithms that use Toeplitz matrices.

- Bauer
- Schur
- Levinson-Durbin

Algorithms that use state-space formulations.

- Riccati Equation
- Kalman Filter
- Chadrasekhar-Kailath-Morf-Sidhu (CKMS)

Given $S_Y(z)$, $Y_n \in \mathbb{C}^d$ for $n > -\infty$, (stationary discrete-time stochastic process)

$$S_Y(z) = \sum_{n=-\infty}^{\infty} R_Y(n) z^{-n},$$

Now, if the decay of the covariance is sufficiently fast it is reasonable to truncate $S_Y(z)$ to a Laurent polynomial

$$\tilde{S}_Y(z) = \sum_{n=-m}^{m} R_Y(n) z^{-n}.$$

It is possible to construct [1, p. 488], $\tilde{Y}_n$ (finite state-spaces process) with

$$S_{\tilde{Y}}(z) = \tilde{S}_Y(z)$$

,

$$\begin{cases} X_{i+1} &= FX_i + Gv_i \\ \tilde{Y}_i &= HX_i + u_i \end{cases}$$

provided that

$$F = \begin{pmatrix} 0 & & & & \\ I & 0 & & & \\ & I & 0 & & \\ & & \ddots & \ddots & \\ & & & I & 0 \end{pmatrix} \in \mathbb{C}^{md \times md}$$

$$H = \begin{pmatrix} 0 & \dots & 0 & I \end{pmatrix} \in \mathbb{C}^{d \times md}$$

$$\mathbb{E} \begin{pmatrix} v_i & u_i \end{pmatrix} \begin{pmatrix} v_j^* \\ u_j^* \end{pmatrix} = \begin{pmatrix} R\delta_{ij} & S\delta_{ij} \\ S^*\delta_{ij} & Q\delta_{ij} \end{pmatrix}$$

$$\Pi = F\Pi F^* + GQG^*$$

$$GS = N - F\Pi H^*$$

$$R = R_Y(0) - H\Pi H^*$$

$$\Pi = \mathrm{cov}(X_i, X_i) = \mathbb{E}X_iX_i^* \quad \left(\in \mathbb{C}^{m \times m}\right)$$

$$N = \begin{pmatrix} R_Y(m) \\ R_Y(m-1) \\ \vdots \\ R_Y(1) \end{pmatrix}$$

So, we have a **time-invariant, stationary statespace model** that approximates the original process in the sense that the z-spectra are close.

With this finite, linear state-space model we can use the **Kalman filter** to produce an innovations model that will correspond to a **casual and causally invertable model filter** for the process $\tilde{Y}_n$ and therefore the inverse constitute a whitening filter. The result is well know and we choose the following representation [2, p. 335].

$$\begin{cases} \theta_{i+1} &= F\theta_i + K_i e_i, \qquad \theta_0 = 0 \\ \tilde{Y}_i &= H\theta_i + e_i \end{cases}$$

where $\mathbb{E}e_i e_j^* = R_{e,1}\delta_{ij}$,

$$K_i = (N - F\Sigma_i H^*)R_{e,i}^{-1}, \qquad R_{e,i} = R_Y[0] - H\Sigma_i H^*, \qquad \text{and}$$

$$\Sigma_{i+1} = F\Sigma_i F^* + K_i R_{e,i} K_i^* \qquad \Sigma_i = \mathbb{E}\theta_i\theta_i^*$$

We want to consider this system in steady state, but will K_i, $R_{e,i}$ converge? Yes, this is a consequence of F being stable. And the limits can be approximated using CKMS. Let $K = \lim_i K_i$, $R = \lim_i R_i$,

$$\begin{cases} \theta_{i+1} &= F\theta_i + Ke_i \\ \tilde{Y}_i &= H\theta_i + e_i, \qquad \mathbb{E}e_i, e_j^* = R\delta_{ij} \end{cases}$$

In our context this system may be represented as a convolution since

$$\tilde{Y}_i = e_i + \sum_{j=1}^{m} K_j e_{i-1-m+j} = (\ell * e)_i$$

where

$$\ell = (1, K_m, K_{m-1}, \ldots, K_1)$$

Program in Applied Mathematics

We have now a approximating MA(m) where the approximation is in the sense of their z-spectra being close.
Observe that

$$S_{\tilde{Y}}(z) = S_{\ell * e}(z) = L(z)S_e(z)L^*(z^{-*}) = L(z)RL^*(z^{-*})$$

where L is the z-transform of ℓ.

$$L(z) = \sum_{k=1}^{\infty} \ell_k z^{-k+1}$$

And so,

$$S_Y(z) \approx \tilde{S}_Y(z) = S_{\tilde{Y}}(z) = L(z)RL^*(z^{-*})$$

provides a spectral factorization.

Example 1 (again): AR(2) Signal

Let us again consider the stationary autoregressive process of order 2,

$$Y_n = (r_1 + r_2) Y_{n-1} - r_1 r_2 Y_{n-2} + U_n, \qquad \text{for } n > -\infty$$

for $r_1, r_2 \in \{z : |z| < 1\}$ and U_n are i.i.d. standard normal random variables.

We already computed the z-spectrum. as follows

$$S_Y(z) = \frac{1}{(1 - r_1 z^{-1})(1 - r_2 z^{-1})(1 - r_1^* z)(1 - r_2^* z)}$$

Lets compute the spectral factor.

$$L(z) = \frac{1}{(1 - r_1 z^{-1})(1 - r_2 z^{-1})} = \left(\sum_{n=0}^{\infty} r_1^n z^{-n} \right) \left(\sum_{n=0}^{\infty} r_2^n z^{-n} \right)$$

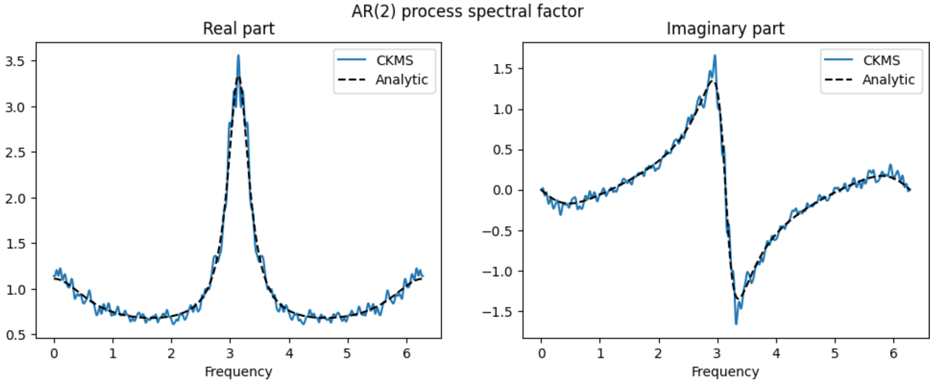
observe that $L(z)L(z^{-*}) = S_Y(z)$ and that $L(z)$ is minimum-phase.

Example 1: AR(2) Signal

Program in Applied Mathematics

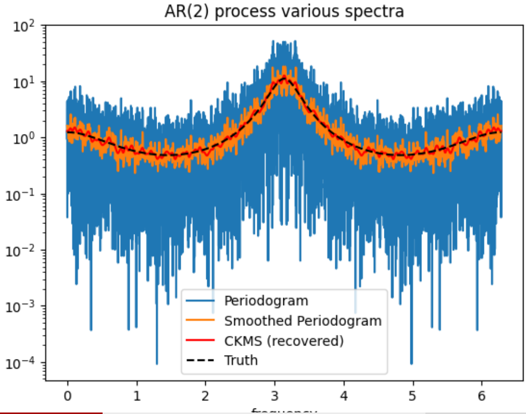
```
mr = include("../../Tools/WFMR.jl")
Nex = 10^4
A = at.my_smoothed_autocov(reshape(y,1,:); L = 200)
ℓ = mr.spectfact_matrix_CKMS(reshape(A,1,1,:))[1][:]

L_ckms = at.transferfun(ℓ;Nex)
L_ana  = map(z -> 1/f(z),exp.(2π*im*(0:Nex-1)/Nex))
θ = 2π*(0:Nex-1)/Nex

figsize = (12,4)
figure(;figsize); suptitle("AR(2) process spectral factor")
subplot(121); title("Real part"); xlabel("Frequency")
plot(θ,real(L_ckms), label = "CKMS")
plot(θ,real(L_ana), "k--",label = "Analytic")
legend()
subplot(122); title("Imaginary part"); xlabel("Frequency")
plot(θ,imag(L_ckms),label = "CKMS")
plot(θ,imag(L_ana), "k--",label = "Analytic")
legend()
```

Example 1: AR(2) Signal

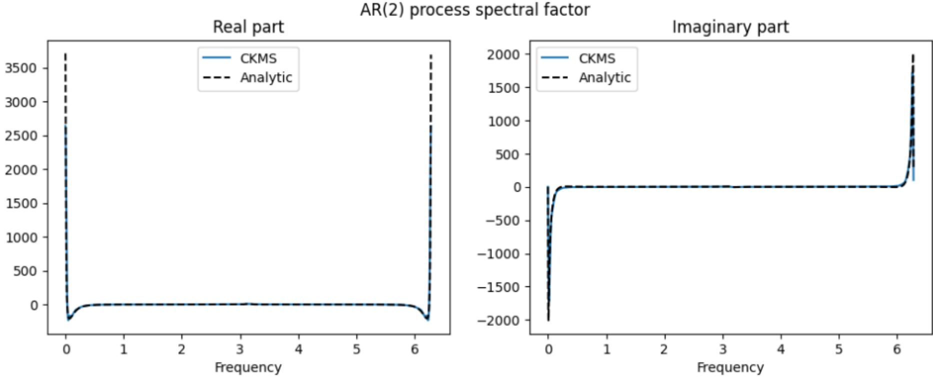


Example 1: AR(2) Signal

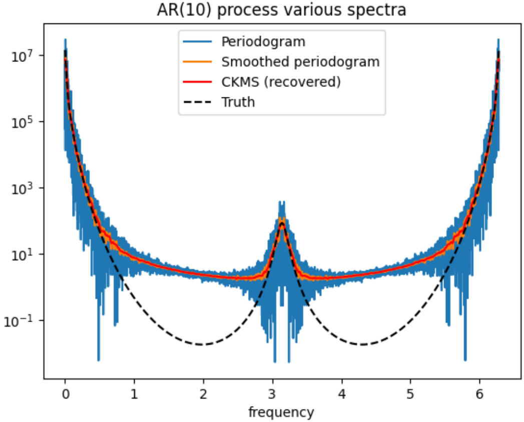
```
θ = 2pi*(0:N-1)/N
title("AR(2) process various spectra")
semilogy(θ,Sy_per, label = "Periodogram")
semilogy(θ,Sy_num, label = "Smoothed Periodogram")
semilogy(θ,abs2.(L_ckms), "r", label = "CKMS (recovered)")
semilogy(θ,Sy_ana, "k--", label = "Truth")
xlabel("frequency"); legend()
```



Example 2 (again): AR(10) Signal



Example 2: AR(10) Signal



Whitening

Recall that spectral factorization algorithm produced a **modeling** filter $L(Z)$ with impulse response ℓ. This filter was minimum-phase.

$$e \xrightarrow{L(z)} y \qquad y \xrightarrow{L(z)^{-1}} e$$

The inverse of the modeling filter is a whitening filter.

A word about DFT

For the DFT which is used frequently in this work I use `fft` from FFTW.jl which is a Julia wrapper for the FFTW library written in C.
Here is what it does:

$$v_k = \texttt{fft}(u)_k = \sum_{j=1}^{N} u_j e^{-\frac{2\pi i}{N}(j-1)(k-1)}$$

$$u_j = \texttt{ifft}(v)_j = \frac{1}{N}\sum_{k=1}^{N} v_k e^{\frac{2\pi i}{N}(k-1)(j-1)}$$

Here is why I use it so much:
Suppose we have the function $S(z) = \sum_{j=1}^{N} c_j z^{-(j-1)}$ which we wish to evaluate at N equally-spaced, unit-circle points $z_k = e^{\frac{2\pi i}{N}(k-1)}$ for $k = 1, \ldots, N$. We need only use `fft` to get

$$S(z_k) = \sum_{j=1}^{N} c_j e^{-\frac{2\pi i}{N}(j-1)(k-1)} = \texttt{fft}(c)_k.$$

So, given a causal finite impulse response (FIR) filter ℓ, it's transfer function $L(z)$ evaluated at N_{ex} evenly distributed points on the unit circle is the array

$$\left(L(z) : z = e^{2\pi ik/N_{ex}} \text{ for } k = 0, \ldots, N_{ex} - 1\right) = \texttt{fft([}\ell\texttt{;zeros(Nex-length(}\ell\texttt{))])}$$

The first entry corresponds to $L(1)$ and the points go counterclockwise.
So, to get an approximate inverse of an causal FIR a filter.

```
# Compute coefficients of spectral factorization of z-spect-pred
S⁻ = mr.spectfact_matrix_CKMS(R; ϵ = tol_ckms, N_ckms)

Err  = S⁻[2] ###
S⁻ = S⁻[1][:]                               # the model filter ###

h_m = S⁻[1:M]

S⁻ = nfft >= par ? [S⁻; zeros(eltype(S⁻), nfft - par)] : S⁻[1:nfft]

fft!(S⁻)   # z-spectrum of model filter
Sinv = (S⁻).^-1

h_w = ifft(Sinv)[1:M];
```

Let us yet again consider the stationary autoregressive process of order 2,

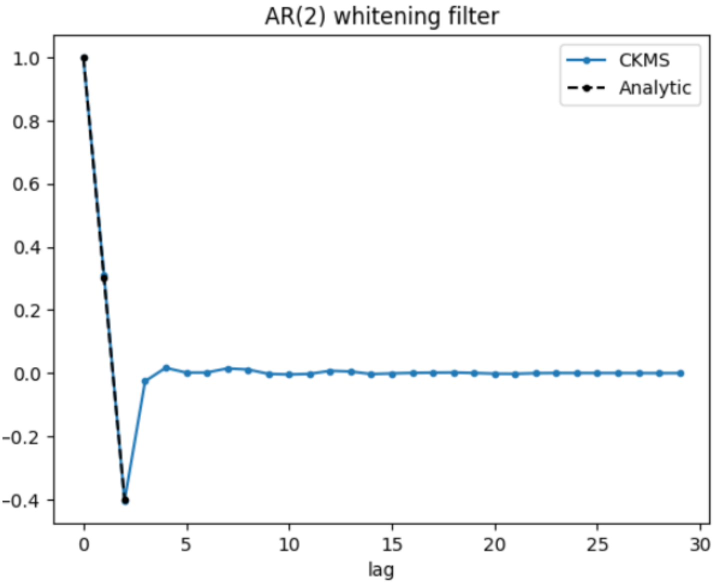
$$Y_n = (r_1 + r_2)Y_{n-1} - r_1 r_2 Y_{n-2} + U_n, \qquad \text{for } n > -\infty$$

for $r_1, r_2 \in \{z : |z| < 1\}$ and U_n are i.i.d. standard normal random variables. Clearly, $w = (1, -(r_1 + r_2), r_1 r_2)$ is a whitening filter. That is

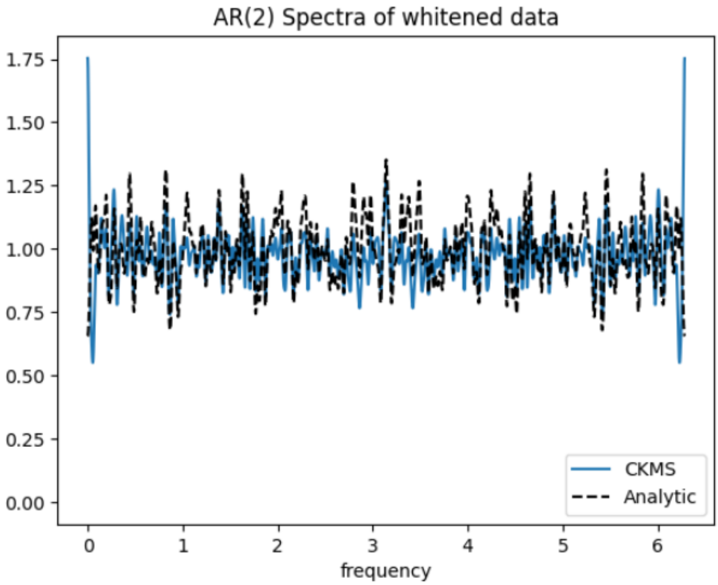
$$(w \star Y)_n = Y_n - (r_1 + r_2)Y_{n-1} + r_1 r_2 Y_{n-2} = U_n$$

```
whf = include("../../Tools/WhiteningFilters.jl")

w = real( whf.get_whf(y;par = 30)[2] )
plot(w,".-",label = "CKMS")
plot(r,"k.--", label = "Analytic")
title("AR(2) whitening filter"); xlabel("lag"); legend()
```

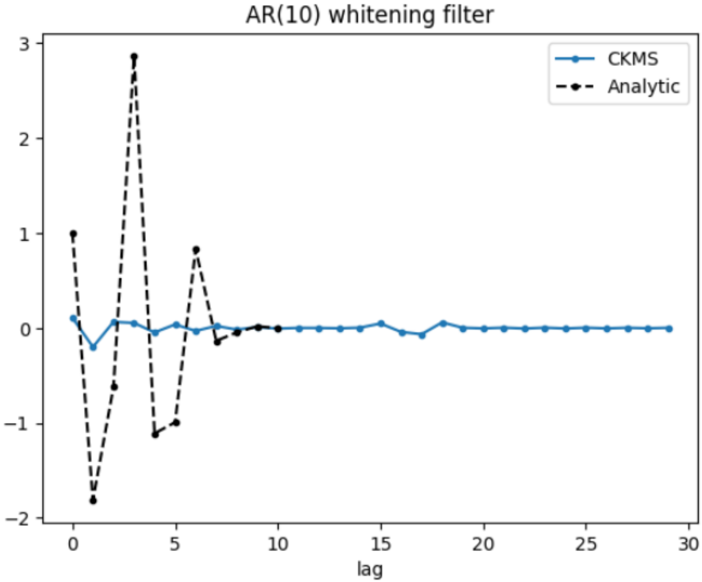
Example 1: AR(2)



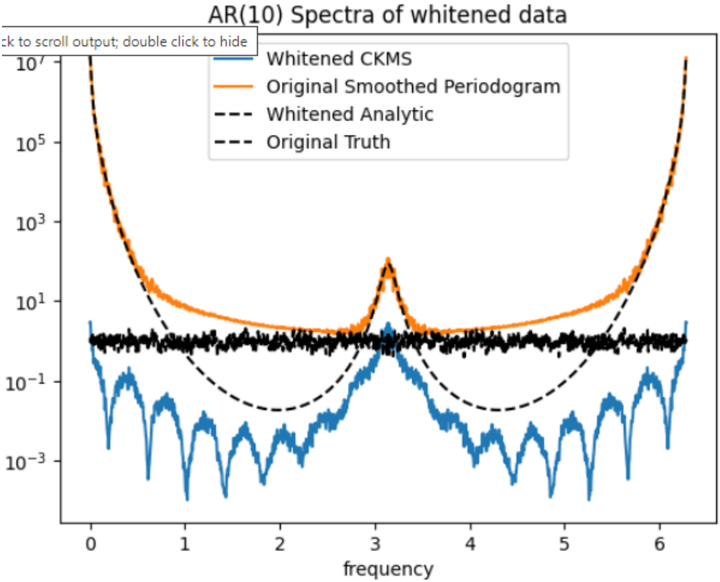
Program in Applied Mathematics



Example 2: AR(10)



Example 2: AR(10)



Example 2: AR(10)

This did not work well.

Any suggestions?

Should I whiten again?

Example 2: AR(10)

Program in Applied Mathematics

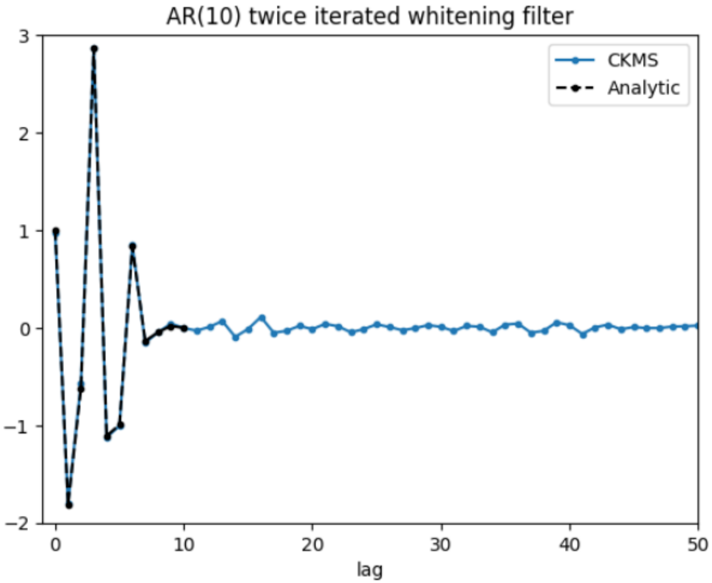
So, I whitened again.
Meaning I took the "whitened" processes as input and using the CKMS spectral factoring algorithm a second time got a second whitening filter.
Let $\mathcal{W}$ denote the operator from the space of signals to their numerical approximate causal FIR whitening filter, with a specified number of taps.

$$\begin{aligned} \mathcal{W}[Y] &= w^{(1)} \\ Y_n^{(1)} &= (w^{(1)} \star Y)_n \qquad \text{first whitening} \end{aligned}$$

$$\begin{aligned} \mathcal{W}\left[Y^{(1)}\right] &= w^{(2)} \\ Y_n^{(2)} &= \left(w^{(2)} \star Y^{(1)}\right)_n \qquad \text{second whitening} \\ &= \left((w^{(2)} \star w^{(1)}) \star Y\right)_n \end{aligned}$$

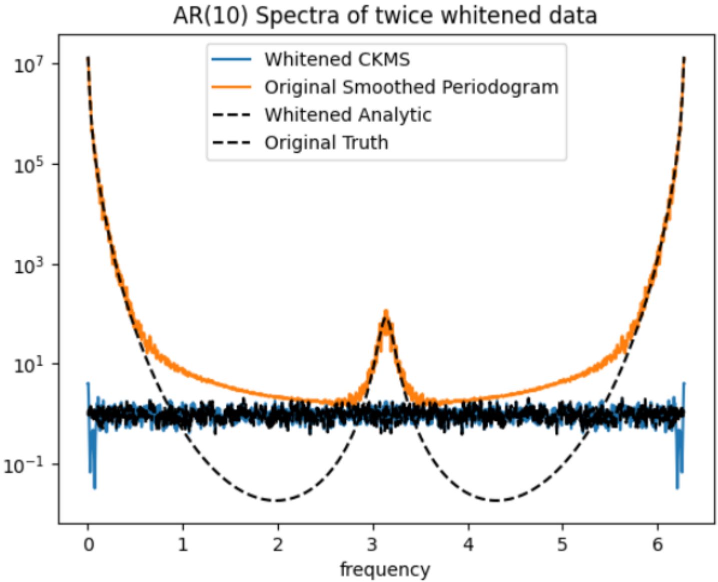
So let $w = w^{(2)} \star w^{(1)}$

Example 2: AR(10)



Example 2: AR(10)

Program in Applied Mathematics



Example 2: AR(10)

Possible explanation: *This is on going*. There are details, features in the spectrum of some processes that are not resolved by the smoothed periodogram (which are used in the whitening algorithm), given limited samples.

Those features seem to be in the data still.

When I whiten I clear out some of the distracting (more apparent features) features and the heretofore hidden features are now accessible to the smoothed periodogram estimate.

Example 3: KSE

I first saw this while trying to whiten a KSE solution.

The Kuromoto-Sivishinsky equation (KSE) can be written as follows

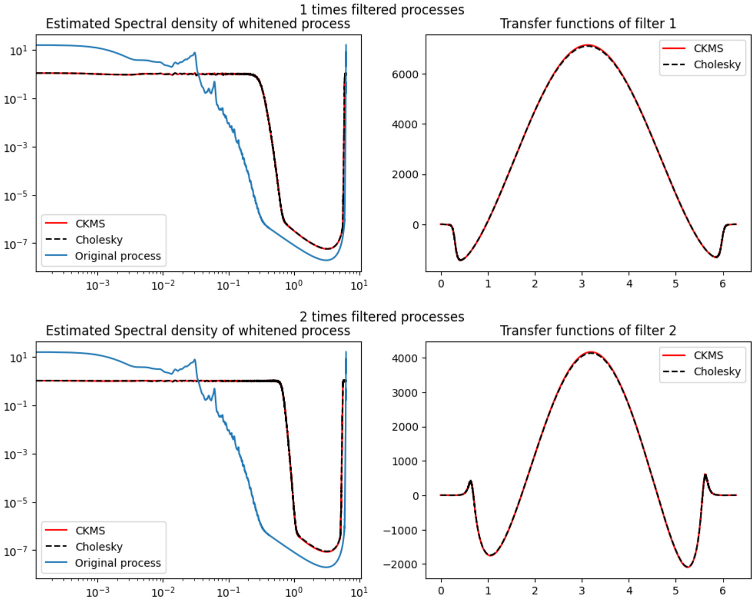
$$u_t + uu_x + u_{xx} + u_{xxxx} = 0$$

with $u(x+L,t) = u(x,t)$ for all $x \in \mathbb{R}$ and $t > 0$. And with $u(x,0) = g(x)$. Now, we use a fourier series to rewrite the KSE is Fourier space. Doing so gives

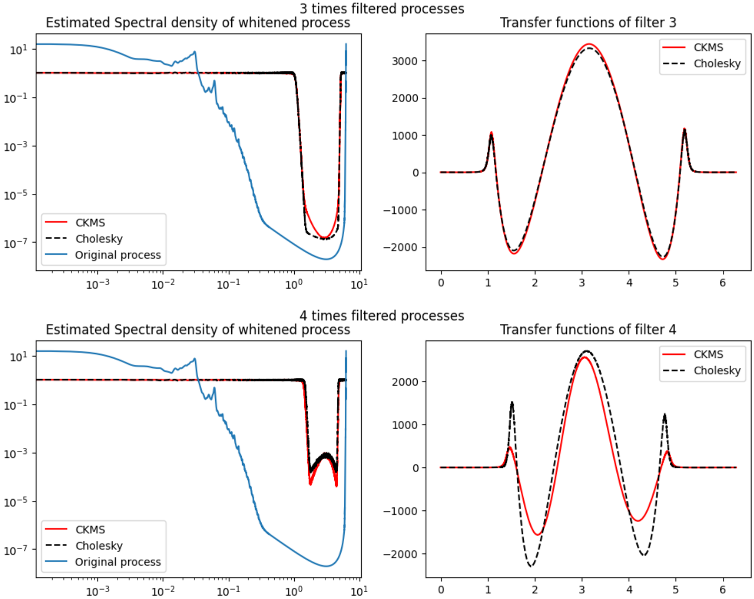
$$\dot{\hat{u}}_k = (q_k^2 - q_k^4)\hat{u}_k - \frac{iq_k}{2} \sum_{\ell=-\infty}^{\infty} \hat{u}_\ell \hat{u}_{k-\ell} \qquad (1)$$

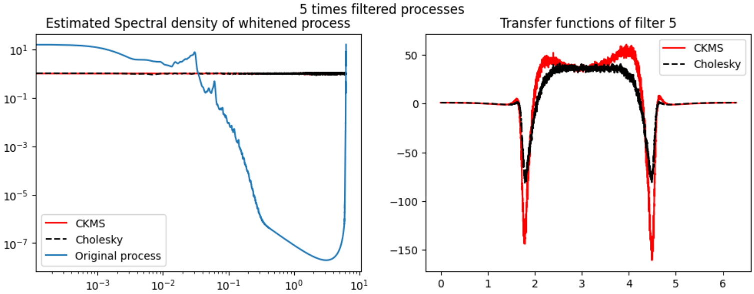
Here, $q_k = \frac{2\pi}{L}k$. Note the trick: $uu_x = \frac{1}{2}\left(u^2\right)_x$.

Example 3: KSE



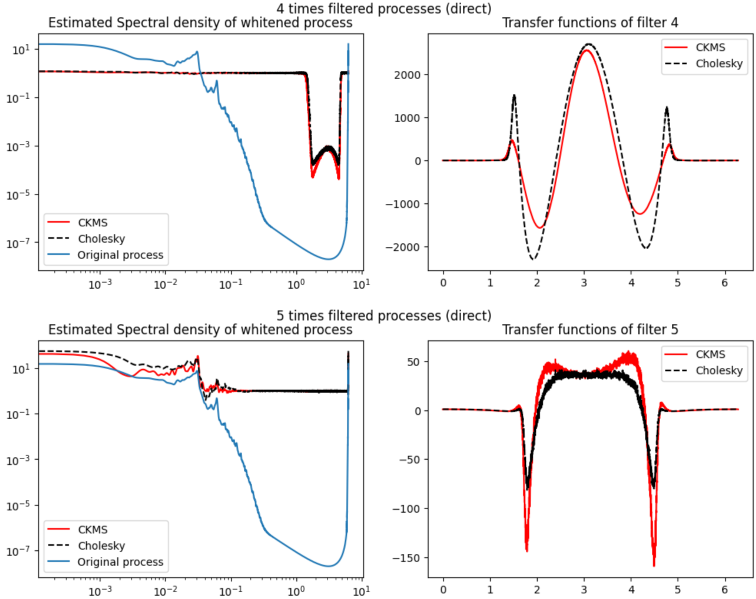
Example 3: KSE





5 times filtered processes
Estimated Spectral density of whitened process
CKMS
Cholesky
Original process
Transfer functions of filter 5
CKMS
Cholesky

Example 3: KSE



Spectrum estimation by a modeling filter

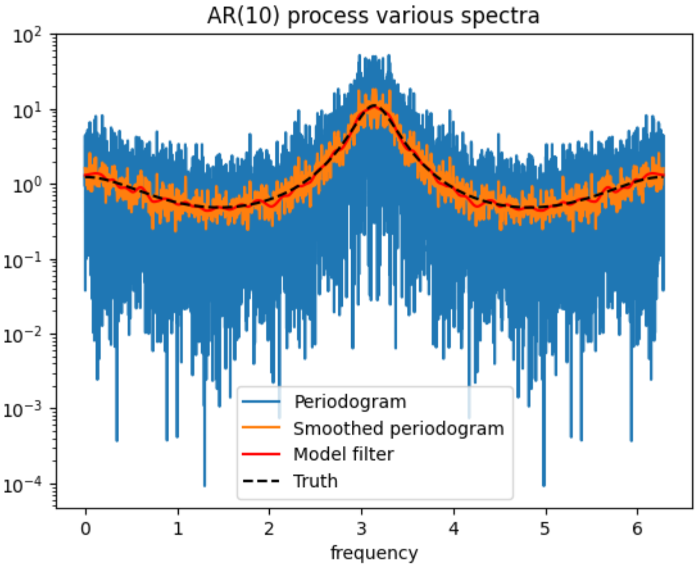
Here is the plan

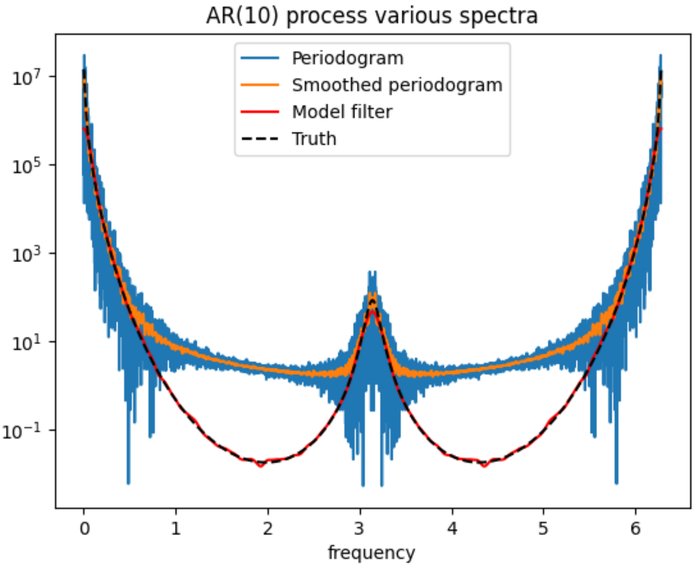
Algorithm 1 Direct Iterated Whitening

```
x = timeseries
wx = timeseries
H = M = [1] (the identity filter)
for i = 1 to max number of iterations do
    h, m = whitening filter and modeling filter (respectively) for wx discarding the first
    i × M samples in the series to eliminate transients from the computation
    H = H * h
    M = M * m
    wx = H * x
end for
return M
```

Then take the master modeling filter and compute the absolute square of its transfer function.

Example 2: AR(10)




AR(10) process various spectra
Periodogram
Smoothed periodogram
Model filter
Truth
10^7
10^5
10^3
10^1
10^{-1}
0
1
2
3
4
5
6
frequency

Thank you!

- Ali H Sayed and Thomas Kailath.
  A survey of spectral factorization methods.
  Numerical linear algebra with applications, 8(6-7):467–496, 2001.
- Thomas Kailath, Ali H Sayed, and Babak Hassibi.
  Linear estimation.
  Number BOOK. Prentice Hall, 2000.