A Computationally Empirical Adaptation of the Prony Method which Assures Consistent Higher Precision and Stability in the Reconstructed Output Signal Components

Parthasarathy Srinivasan

Oracle Corporation, USA

Abstract

The Prony method for approximating signals comprising sinusoidal/exponential components is known through the pioneering work of Prony in his seminal dissertationin the year1795. However, the Prony method saw the light of real world application only upon the advent of the computational era, which made feasible the extensive numerical intricacies and labor which the method demands inherently. While scientific works (such as Total Least Squares method) exist , which focus on alleviating some of the problems arising due to computational imprecision, they do not provide a consistently assured level of highly precise results. This study improvises upon the Prony method by observing that a better(more precise) computational approximation can be obtained under the premise that adjustment can be made for computational error, in the autoregressive model setup in the initial step of the Prony computation itself. This adjustment is in proportion to the deviation of the coefficients in the same autoregressive model. The results obtained by this improvisation live up to the expectations of obtaining consistency and higher value in the precision of the output (recovered signal) approximations as shown in this current work.

Keywords

Prony Method, Fourier Series, Auto Regression, imprecision.

1. Introduction

The Prony method is an effective observation on the transformation of an exponential expression to a mathematically convenient and tractable polynomial form.

The cornerstone of the method is to setup the autoregressive model where the subsequent values of the input signal (are assumed to) depend regressively on the prior values and then solve the model (by dividing the Toeplitz matrix with the input signal and obtaining the remainder) to obtain the auto regression coefficients(a).

The characteristic polynomial summing the auto regression coefficients(a) is then solved to obtain the roots. The roots obtained directly provide the value of the damping factor and frequency of the output approximating signal. The roots obtained are then also, used (in the form of knowns) to

again setup and solve the original autoregressive model to obtain the Amplitude, and phase, of the approximating (or smoothened for lack of a better term) output signal. As obvious the procedure described above is mathematically(computationally) intensive and precise only up to the precision allowed for in the computational process.

Thus, the Prony method was practicable only with modern computers came to aide. This made the computing task easier however introduced a limit (albeit smaller than manual computation) on the precision to which the resultant (approximating) signal resembled the original input signal.

In this current work, the author makes the empirical premise that the computer introduces an imprecision proportional to the variance of the signals' autoregression coefficients and therefore can be thus corrected to yield a more precise output estimate of the Amplitude, frequency, dampener, and phase.

2. CONTEXT OF CURRENT WORK IN REGARD TO RECENT LITERATURE OF RELATED WORKS

In the recent (century) time frame (~2014,2022) there have been various attempts to comprehend, adapt, implement, and optimize the Prony method with modern electronic and computing platforms doing the heavy duty tasks. These adaptations take the form of varying the parameters in the core Prony Algorithm (with optimization in perspective)[7], generating usable intermediary output byproduct from the core algorithm (for use in related settings and contexts)[5], evaluating the utility of the transformative power (in the co-domain) of the algorithm for ease of scientific analysis[6]etc. However, the objective of this present work is more subtle and profound, in that the adaptation proposed here-in has wide-reaching consequences (in terms of results) in critical applications across diverse domains and application functional areas.

3. SCOPE AND SIGNIFICANCE OF WORK

This work is focused on obtaining reliable bounds in the computational precision of the output from the Prony method. The bounds so obtained are significant in the practical application of the method owing to the requirement that, for the diverse domains in which the method is applied, such as biomedical engineering, power systems etc. the level of precision significantly impacts the ends for which the method is used , for example the monetary cost effectiveness in the case of power systems and the effectiveness of diagnosis in the case of bio-medical engineering. The work is unique in that though works with similar objectives exist , the particular premise and methodology described are not found in existing literature.

4. FUNDAMENTAL PREMISE IN THIS ADAPTATION OF PRONY'S METHOD

The implementation of the Prony's method in MATLAB is described in [2]. As mentioned above, in this adaptation of the Prony method, the coefficients of the characteristic polynomial, computed in the first step autoregressive model are self-adjusted by a constant multiple of the standard deviation of their own values and this assuredly compensates for the imprecision in the computation due to the computing device's limitation in capacity to perform the computation to an arbitrary precision.

5. METHODOLOGY EMPLOYED IN THIS ADAPTATION OF PRONY'S METHOD

Step 1 :

This step consists of setting up an Auto Regressive Model with the random sample observations, where the assumption is that the observations are naturally organized such that the nth observation is linearly dependent on the n-1 pre-ceding observations thus yielding a model which regressively relies on computing the Toeplitz matrix set up to relate those observations in accordance with the above assumption.

Step 2 :

Solving the above auto-regressive model yields coefficients of the Prony characteristic polynomial which is pre-requisite to obtain the Amplitude and Frequency of the final approximating signal output as yielded by the polynomial roots.

Step 3 : (Nuanced Prony method proposed by this work)

Adjust the coefficients of the characteristic polynomial by a standard deviation of their own values. This work proves that this adjustment helps overcome the computational imprecision introduced due to limitations of the computing devices' arbitrary precision arithmetic.

Step 4 : Work with the adjusted polynomial coefficients to obtain the Amplitude and Frequency of the output signal components and also compute the damping factor and phase by yet another derivative system of equations obtained by substitution of obtained roots in the original equation.

Step 5 : Compute the Precision Metric defined and observe that the nuance proposed here-in lives up to the expectation of yielding constant lower bound in the precision of the output signal approximation.

6. DETAILS OF HARDWARE AND SOFTWARE USED IN THE EXPERIMENTAL EMULATION

Hardware	Operating System	Software Library
PC 64 bit (AMD64)	Windows 10	MATLAB R2024a

7. DEFINITION OF METRIC TO QUANTIFY THE PRECISION OF THE RECOVERED OUTPUT SIGNAL

To quantify the improvement in precision with this adaptation to the Prony method the following formulation (similar to **[1]**) is used as a measure of the comparative similarity in the input and output signals in both the unchanged Prony method and the adaptation introduced in this work.

Precision Measure,

PM = N – (norm(grecons[i]-g[i])/norm(grecons[i]-mean(g[i]))),

where N=size(g[i]), g[i] = input signal vector, grecons[i] = reconstructed output signal vector, norm is the standard 2 norm,
and mean is the usual mean definition.

8. EXPERIMENTAL RESULTS OF THE PRECISION MEASURE

The results are divided into 3 distinct tiers based on the size of the input signal set vector, to the computation, as follows :

Input Vector Size	Coefficient Adjustment Factor	Precision Measure Value (consistently for this adaptation of Prony method) (constant across several program runs)
100	1*sigma(coefficients)	90
1000	10*sigma(coefficients)	968.3772
10000	100*sigma(coefficients)	9900

The intuitive rigor behind the empirical constant results obtained above by experiment seems to be the fact that the amount memory required for the computer's mathematical operations in terms of precision grows linearly with the variance of the values being computed itself. This is likely an aggregate manifestation of the optimizations of the memory architecture implementation.

9. CONCLUSION AND FUTURE WORK

This work has proven that with the adaptation of the Prony method set forth by the Author there is a consistent improvement of the Lower Bound in the recovered signal precision, thus improvising the Prony Method for advanced use in diverse and sensitive fields such as Bio-Medical engineering and Electrical Power Systems. The Bio-Medical applications include improvement to disease detection by signal analysis as in [1].

There is scope for rigorous Stability Analysis (of recovered signal) with the adjustment founded by this work in place.

Also, there is scope to apply this work in the realm of Quantum Computing to evaluate whether the same empirical results are upheld in that realm as well.

REFERENCES

[1]   Fernández Rodríguez et al. BMC Bioinformatics (2018) 19:451 https://doi.org/10.1186/s12859-018-2473-y

[2]   80 IEEE Transactions on Power Systems, Vol. 5, No. I, February 1990 INITIAL RESULTS IN PRONY ANALYSIS OF POWER SYSTEM RESPONSE SIGNALS J.F. Hauer Senior Member C• J. Demeure Member L.L. Scharf Fellow

[3]   Parameter Estimation in Electrical Power Systems Using Prony's Method Claudius Ortbandt et al 2015 J. Phys.: Conf. Ser. 659 012013

[4]   https://amses-journal.springeropen.com/fourier-based- computational-approaches-pdes

[5]   Research on Fourier and Prony analysis algorithm of inter-harmonic in power system Hong Lei https://www.sciencedirect.com/science/article/pii/S2352484722004528

[6]   Online identification and suppression of low frequency oscillation in power system based on WAMS(Conference   Paper)   https://www.scopus.com/record/display.uri?eid=2-s2.0-84897735036&origin=inward&txGid=c7699b92fff1ead1b2030efc1fc825f7

[7]   A Fault Signal Processing Method Based On An Improved Prony Algorithm https://www.frontiersin.org/journals/energy-research/articles/10.3389/fenrg.2021.831347/full

Appendix A -→ MATLAB Code of Modified Prony Method

```
%Code 1 Polynomial Implementation
function [Amp,alfa,freq,theta]=polynomial_method (x,p,Ts,method)
% method: 'classic', 'Is' or 'tls' (case insensitive)
% define the solving methods
CLASSIC = 0;
LS = 1;
TLS = 2;

N = length(x);

if strcmpi(method,'classic')
if N ~= 2*P
disp ('ERROR: length of x must be 2*p samples in classical method.');
Amp = [];
alfa = [];
freq = [];
theta = [];
return;
end
else solve_method = CLASSIC;
end

%% step 1

T = toeplitz(x(p:N-1) ,x(p:-1:1));

switch solve_method
case {CLASSIC, LS}
a = -T\x(p+1:N);

case TLS

a= tls(T,-x(p+1:N));

end

%% step 2
c = transpose([1; a]);
r = roots(c);

rprim = r - (std(r)/n);
%rprim = r - std(r); %TBTD

alfa = log(abs(r))/(2*pi*Ts);

% In case alfa equals to +/-Inf the signal will not be recovered for n=0
%(Inf*0 = Nan). Making alfa = +/-realmax that indeterminance will be solved
alfa(isinf(alfa))=realmax*sign(alfa(isinf(alfa)));

%% step3
switch solve_method
case CLASSIC
len_vandermonde = p; % exact case (N=2p) find h with p samples
case LS
len_vandermonde = N; % overdetermined case (N>2p) find h with N samples
```

```
case TLS
len_vandermonde =N; % overdetermined case (N>2p) find h with N samples
end

Z = zeros(len_vandermonde,p);
for i=1:lenght(r)

z(:,i) = transpose(r(i).^(0:len_vandermonde-1));
end

rZ = real(Z);
iZ = imag (Z);
% here Inf values are substituted by realmax values
rZ(isinf(rZ))=realmax*sign(rZ(isinf(rZ)));
iZ(isinf(iZ))=realmax*sign(iZ(isinf(iZ)));

z = rZ+1i*iZ

switch solve_method
case {CLASSIC,LS}
h = Z\x(1:len_vandermonde);
case TLS
% if exists nan values SVD won't work
indeterminate_form = sum(sum(isnan(Z) | isinf(Z)));
if (indeterminate_form)
Amp = []; alfa = []; freq = []; theta = [];
return;
else
h = tls(Z,x(1:len_vandermonde))
end
end
Amp = abs(h);
theta = atan2(imag(h),real(h));
```