Data Article

Supplemental data for the paper “low-complexity detection of small frequency deviations by the generalized LMPU test”

Eyal Levy, Tirza Routtenberg*

School of Electrical and Computer Engineering, Ben-Gurion University of the Negev, Beer-Sheva 84105, Israel

Abstract

This document contains supplemental material for the paper [2]. The notations in this document are the same as in [2]. In particular, we first present here the proof of Theorem 1 in [2]. This theorem expresses the locally most powerful unbiased (LMPU) test, which is a general method for local detection, in the presence of known nuisance parameters. Second, we present here the Matlab code of the LMPU and the generalized LMPU for the special case of detection of a small deviation in the frequency of sinusoidal signals, which arises in various signal processing applications.

© 2021 The Authors. Published by Elsevier Inc. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/)

DOI of original article: 10.1016/j.sigpro.2020.107851
* Corresponding author.
E-mail address: tirzar@ee.bgu.ac.il (T. Routtenberg).

https://doi.org/10.1016/j.dib.2020.106714
2352-3409/© 2021 The Authors. Published by Elsevier Inc. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/)
Specifications Table

| Subject | Data Mining and Statistical Analysis |
|---------|--------------------------------------|
| Specific subject area | Detection Theory |
| Type of data | Mathematical derivations, MATLAB code |
| How data were acquired | All source codes written in Matlab software. |
| Data format | Raw |
| Parameters for data collection | All the codes were implemented in MATLAB-R2019a on a system with Intel Core(TM) i7-10TH GEN CPU computer, 2.80 GHz. |
| Description of data collection | The MATLAB code was configured as a function. The experiments were conducted by generating Monte-Carlo simulations of the model from (1) in [2], where the noise is generated by the Matlab function: wgn(N,’complex’), in which N is the number of measurements. |
| Data source location | All relevant data is contained in this paper |
| Data accessibility | With the article |
| Related research article | E. Levy and T. Routtenberg, “Low-complexity detection of small frequency deviations by the generalized LMPU test,” Signal Processing, https://doi.org/10.1016/j.sigpro.2020.107851. |

Value of the Data

- This document contains the proof of Theorem 1 in [2], which expresses the LMPU test that is a function of a known parameter vector, $\theta_n$. This is used in [2] to develop the GLMPU. This proof is important for the sake of completeness and since the LMPU it is less widely used than other detectors.
- This proof can be used in the future for developing new tests for various scenarios.
- The provided codes can be used for the detection of frequency deviations with low complexity.

1. Data Description

This article contains supplementary material for the paper in [2]. Specifically, it contains the proof of Theorem 1 in [2] and the code files in Matlab that were used to generate the graphs in [2]. Theorem 1 in [2] presents the locally most powerful unbiased (LMPU) test, which is a general test that maximizes the probability of detection under the $\alpha$-size and unbiasedness constraints. This theorem presents the explicit test as a function of the likelihood function and as a function of the unknown nuisance parameters. The Matlab code contains the functions of the LMPU and generalized LMPU (GLMPU) tests that were used to generate Figs. 1–7 in [2], and is attached to this paper.

2. Experimental Design, Materials and Methods

3. Proof of Theorem 1 in [2]

A general non-random test, based on the observation vector, $\mathbf{x}$, can be defined as

$$
\Phi(\mathbf{x}) \triangleq \begin{cases} 
1, & \mathbf{x} \in \mathcal{S}_1, \\
0, & \mathbf{x} \notin \mathcal{S}_1.
\end{cases}
$$

(3.1)

where $\mathcal{S}_1 \subset \Omega_x$ is the rejection region, which includes the values of $\mathbf{x}$ that lead to rejection of $\mathcal{H}_0$ (acceptance of $\mathcal{H}_1$).
In (6) in [2] we describe the general two-sided composite hypothesis testing problem, in which \( \theta_1 = \theta_0 \) under the null hypothesis and \( \theta_1 \neq \theta_0 \) under the alternative hypothesis, and \( \theta_n \) is a nuisance parameter vector that appears under both hypotheses.

The power function of a general test (3.1), \( \Phi \), for this hypothesis testing is defined by (p. 69 in [1]):

\[
\beta_\Phi(\theta_1, \theta_n) \triangleq \int_{\Omega_x} \Phi(x) f(x; \theta_1, \theta_n) \, dx, \quad \theta_1 \neq \theta_0. \tag{3.2}
\]

The pdf of \( x \) under both hypotheses, \( f(x; \theta_1, \theta_n) \), is assumed to be a continuous and twice differentiable function w.r.t. the local parameter, \( \theta_1 \in \mathbb{R} \), for any unknown nuisance parameter vector, \( \theta_n \in C^K \). Therefore, the power function \( \beta_\Phi(\theta_1, \theta_n) \) is also a continuous function w.r.t. the local parameter, \( \theta_1 \in \mathbb{R} \), especially at \( \theta_1 = \theta_0 \).

A level-\( \alpha \), unbiased test, \( \Phi \), is said to be the LMPU test (p. 340 in [1]) if, for any other given level-\( \alpha \), unbiased test \( \tilde{\Phi} \), there exists \( \delta \) such that

\[
\beta_\Phi(\theta_1, \theta_n) \geq \beta_{\tilde{\Phi}}(\theta_1, \theta_n), \quad \forall \theta_1 \in \Omega_\delta. \tag{3.3}
\]

where \( \Omega_\delta \) is the two-sided local neighborhood around \( \theta_1 \), as defined in (7) in [2]. Thus, the LMPU test is obtained by maximizing the power function, \( \beta_\Phi(\theta_1, \theta_n) \), among all locally unbiased tests under the \( \alpha \)-size and unbiasedness constraints from (10) and (11) in [2], respectively, in the neighborhood of \( \Omega_\delta \). The constraints from (10) and (11) in [2] can be rewritten by using (3.2) as follows:

\[
P_{FA}(\theta_0, \theta_n) = \beta_\Phi(\theta_1, \theta_n)|_{\theta_1=\theta_0} = \alpha. \tag{3.4}
\]

\[
P_{BD}(\theta_1, \theta_n) = \beta_\Phi(\theta_1, \theta_n)|_{\theta_1 \neq \theta_0} \geq \alpha, \quad \forall \theta_1 \in \Omega_\delta. \tag{3.5}
\]

Together, these constraints indicate that \( \beta_\Phi(\theta_1, \theta_n) \) has a minimum point at \( \theta_1 = \theta_0 \) on the set \( \Omega_\delta \). Since we assume that the common pdf, \( f(x; \theta_1, \theta_n) \), is twice differentiable in the local neighborhood of \( \theta_1 \) for any \( \theta_n \in C^K \), the constraint in (3.5) can be replaced by the stationary condition

\[
\frac{\partial \beta_\Phi(\theta_1, \theta_n)}{\partial \theta_1} \bigg|_{\theta_1=\theta_0} = 0, \tag{3.6}
\]

together with the condition

\[
\frac{\partial^2 \beta_\Phi(\theta_1, \theta_n)}{\partial \theta_1^2} \bigg|_{\theta_1=\theta_0} > 0. \tag{3.7}
\]

Therefore, by concluding the constraints in (3.4), (3.6), and (3.7), the LMPU test can be obtained from the solution of the following constrained optimization problem:

\[
\max_{\Phi(x)} \beta_\Phi(\theta_1, \theta_n) \text{ s.t. } \begin{cases} 
\beta_\Phi(\theta_0, \theta_n) = \alpha \\
\frac{\partial \beta_\Phi(\theta_1, \theta_n)}{\partial \theta_1} \bigg|_{\theta_1=\theta_0} = 0 \\
\frac{\partial^2 \beta_\Phi(\theta_1, \theta_n)}{\partial \theta_1^2} \bigg|_{\theta_1=\theta_0} > 0, \quad \forall \theta_1 \in \Omega_\delta
\end{cases}. \tag{3.8}
\]

Under the assumptions of differentiability, the Taylor series expansion of the power function, \( \beta_\Phi(\theta_1, \theta_n) \), around \( \theta_1 = \theta_0 \) is given by:
\[ \beta_{\Phi}(\theta_1, \theta_n) = \beta_{\Phi}(\theta_0, \theta_n) + (\theta_1 - \theta_0) \frac{\partial \beta_{\Phi}(\theta_1, \theta_n)}{\partial \theta_1} \bigg|_{\theta_1 = \theta_0} + \frac{1}{2} (\theta_1 - \theta_0)^2 \frac{\partial^2 \beta_{\Phi}(\theta_1, \theta_n)}{\partial \theta_1^2} \bigg|_{\theta_1 = \theta_0} + O(\delta^2) \]

\[ = \alpha + (\theta_1 - \theta_0)^2 \frac{\partial^2 \beta_{\Phi}(\theta_1, \theta_n)}{\partial \theta_1^2} \bigg|_{\theta_1 = \theta_0} + O(\delta^2). \quad (3.9) \]

where the last equality is obtained by substituting the constraint on the false alarm probability from (3.4) and the unbiasedness constraint from (3.6). Thus, according to (3.9), in order to obtain the highest power, \( \beta_{\Phi}(\theta_1, \theta_n) \), for a given \( \alpha \) and \( \theta_n \), we need to maximize the second order term, \( \frac{\partial^2 \beta_{\Phi}(\theta_1, \theta_n)}{\partial \theta_1^2} \bigg|_{\theta_1 = \theta_0} \), for both \( \theta_1 > \theta_0 \) and \( \theta_1 < \theta_0 \), and this leads to the LMPU test. In addition, under the constraint \( \beta_{\Phi}(\theta_0, \theta_n) = \alpha \), (3.9) implies that the constraint in (3.7) is redundant for the maximization of \( \beta_{\Phi}(\theta_0, \theta_n) \) (which is always equal to or larger than \( \alpha \)). Thus, the maximization in (3.8) is equivalent to the following optimization:

\[ \max_{\Phi(\mathbf{x})} \int_{\Omega_{\mathbf{x}}} \Phi(\mathbf{x}) \left( \frac{\partial f(\mathbf{x}; \theta_1, \theta_n)}{\partial \theta_1} \right) d\mathbf{x} \quad (3.10) \]

By using (3.2), it can be verified that

\[ \frac{\partial \beta_{\Phi}(\theta_1, \theta_n)}{\partial \theta_1} \bigg|_{\theta_1 = \theta_0} = \frac{\partial}{\partial \theta_1} \left( \int_{\Omega_{\mathbf{x}}} \Phi(\mathbf{x}) f(\mathbf{x}; \theta_1, \theta_n) d\mathbf{x} \right) \bigg|_{\theta_1 = \theta_0}. \quad (3.11) \]

Under the assumption that the test, \( \Phi(\mathbf{x}) \), is independent of the parameter \( \theta_1 \), the integration and derivatives in (3.11) can be reordered to obtain

\[ \frac{\partial \beta_{\Phi}(\theta_1, \theta_n)}{\partial \theta_1} \bigg|_{\theta_1 = \theta_0} = \int_{\Omega_{\mathbf{x}}} \Phi(\mathbf{x}) \left( \frac{\partial f(\mathbf{x}; \theta_1, \theta_n)}{\partial \theta_1} \right) \bigg|_{\theta_1 = \theta_0} d\mathbf{x}. \quad (3.12) \]

Similarly,

\[ \frac{\partial^2 \beta_{\Phi}(\theta_1, \theta_n)}{\partial \theta_1^2} \bigg|_{\theta_1 = \theta_0} = \int_{\Omega_{\mathbf{x}}} \Phi(\mathbf{x}) \left( \frac{\partial^2 f(\mathbf{x}; \theta_1, \theta_n)}{\partial \theta_1^2} \right) \bigg|_{\theta_1 = \theta_0} d\mathbf{x}. \quad (3.13) \]

Therefore, by substituting (3.12) and (3.13) in (3.10), the integral form of (3.10) is

\[ \max_{\Phi(\mathbf{x})} \int_{\Omega_{\mathbf{x}}} \Phi(\mathbf{x}) \left( \frac{\partial^2 f(\mathbf{x}; \theta_1, \theta_n)}{\partial \theta_1^2} \right) \bigg|_{\theta_1 = \theta_0} d\mathbf{x} \quad (3.14) \]

s.t.

\[ \int_{\Omega_{\mathbf{x}}} \Phi(\mathbf{x}) f(\mathbf{x}; \theta_0, \theta_n) d\mathbf{x} = \alpha \]

\[ \int_{\Omega_{\mathbf{x}}} \Phi(\mathbf{x}) \left( \frac{\partial f(\mathbf{x}; \theta_1, \theta_n)}{\partial \theta_1} \right) \bigg|_{\theta_1 = \theta_0} d\mathbf{x} = 0 \]

By using the auxiliary lemma of the Generalized Neyman-Pearson lemma (p. 77 in [1]) with \( m = 2 \). \( f_1 = f(\mathbf{x}; \theta_0, \theta_n) \). \( f_2 = \frac{\partial f(\mathbf{x}; \theta_1, \theta_n)}{\partial \theta_1} \bigg|_{\theta_1 = \theta_0} \). \( f_3 = \frac{\partial^2 f(\mathbf{x}; \theta_1, \theta_n)}{\partial \theta_1^2} \bigg|_{\theta_1 = \theta_0} \). \( c_1 = \alpha \), and \( c_2 = 0 \), the LMPU test that solved (3.14) rejects the null hypothesis when

\[ \frac{\partial^2 f(\mathbf{x}; \theta_1, \theta_n)}{\partial \theta_1^2} \bigg|_{\theta_1 = \theta_0} > \tilde{\kappa}_1 \frac{\partial f(\mathbf{x}; \theta_1, \theta_n)}{\partial \theta_1} \bigg|_{\theta_1 = \theta_0} + \tilde{\kappa}_2 f(\mathbf{x}; \theta_0, \theta_n). \quad (3.15) \]

where the constants \( \tilde{\kappa}_1 \) and \( \tilde{\kappa}_1 \) are chosen such that the constraints in (3.14) (or, equivalently, the constraints in (3.10)) are satisfied. That is, \( \tilde{\kappa}_1 \) and \( \tilde{\kappa}_1 \) are chosen such that \( \beta_{\Phi}(\theta_0, \theta_n) = \alpha \) and
\[ \frac{\partial \Phi}{\partial \theta_l}(\theta_l, \theta_n) \bigg|_{\theta_l=\theta_0} = 0 \] are satisfied for the test \( \Phi \) defined by (3.15). In addition, it can be verified that

\[ \frac{\partial f(x; \theta_l, \theta_n)}{\partial \theta_l} \bigg|_{\theta_l=\theta_0} = \frac{\partial \log f(x; \theta_l, \theta_n)}{\partial \theta_l} \bigg|_{\theta_l=\theta_0} f(x; \theta_l, \theta_n) \bigg|_{\theta_l=\theta_0} \]  

(3.16)

and

\[ \frac{\partial^2 f(x; \theta_l, \theta_n)}{\partial \theta_l^2} \bigg|_{\theta_l=\theta_0} = \left( \frac{\partial^2 \log f(x; \theta_l, \theta_n)}{\partial \theta_l^2} \right) \bigg|_{\theta_l=\theta_0} f(x; \theta_l, \theta_n) \bigg|_{\theta_l=\theta_0} + \frac{\partial f(x; \theta_l, \theta_n)}{\partial \theta_l} \bigg|_{\theta_l=\theta_0} \frac{\partial \log f(x; \theta_l, \theta_n)}{\partial \theta_l} \bigg|_{\theta_l=\theta_0}. \]  

(3.17)

By substituting (3.16) into (3.17), one obtains

\[ \frac{\partial^2 f(x; \theta_l, \theta_n)}{\partial \theta_l^2} \bigg|_{\theta_l=\theta_0} = \left( \frac{\partial^2 \log f(x; \theta_l, \theta_n)}{\partial \theta_l^2} \right) \bigg|_{\theta_l=\theta_0} f(x; \theta_l, \theta_n) \bigg|_{\theta_l=\theta_0} + \left( \frac{\partial \log f(x; \theta_l, \theta_n)}{\partial \theta_l} \bigg|_{\theta_l=\theta_0} \right)^2 f(x; \theta_l, \theta_n) \bigg|_{\theta_l=\theta_0}. \]  

(3.18)

Then, by substituting (3.16) and (3.18) in (3.15), we get that the LMPU test that solved (3.14) is the LMPU test for a known parameter vector, \( \theta_n \), given in (12) in [2].

4. Code Files

The source code for the LMPU and GLMPU functions for the special case of testing frequency deviations is given below. The MATLAB scripts are used to generate the simulations in [2], as follows:

- **LMPU_FUN.m** - calculates the LMPU test from (21) in [2] for the detection of a small deviation in the frequency of sinusoidal signals, where the amplitudes and noise variance are unknown.
- **GLMPU_FUN.m** - calculates the GLMPU test from (23) in [2] for the detection of a small deviation in the frequency of sinusoidal signals, where the amplitudes and noise variance are unknown.
function Tlmpu = LMPU_FUN(x, sigma, Am, w0)
  % Inputs:
  % x - A matrix composed in the following way:
  % x = [x_1, x_2, ..., x_M], where x_m is a complex
  % observation vector in dimension N measured by ...
  % the mth sensor
  % for all m = 1, ..., M. The measurement model is:
  % x_m = Am * s(w) + noise_m.
  % complex circularly symmetric
  % sigma_m - Vector of the variances of an independent ...
  % zero-mean Gaussian noise sequence in dimension M.
  % The Gaussian noise sequence created using the matlab
  % command: noise_m = wgn(N, 1, ...
  % pow2db(sigma_m(m)), 'complex');
  % Approximately held: sigma_m = var(noise_m).
  % Am - Vector of complex amplitudes in dimension M. The
  % amplitude mth obtain by Am(m) for all m = 1, ...
  % ..., M.
  % w0 - The known nominal system frequency under normal
  % conditions.
  % Output:
  % Tlmpu - The value of LMPU test given x, sigma_m, Am, w0.
  % Constants of the model
  persistent n
  if isempty(n)
    n=(0:1:length(x(:, 1))-1).';
  end % if isempty(n)
  persistent N
  if isempty(N)
    N=length(n);
  end % if isempty(N)
  persistent gama
  if isempty(gama)
    gama=2*pi/N;
  end % if isempty(gama)
  persistent M
  if isempty(M)
    M=length(Am);
  end % if isempty(M)
  persistent D_N
  if isempty(D_N)
    D_N=diag(n);
  end % if isempty(D_N)
  persistent s
  if isempty(s)
    s=exp(1i*gama*n);
  end % if isempty(s)
  % LMPU algorithm
  sum1=0;
  sum2=0;
  for k=1:M
    sum1=sum1-((gama/(w0*sigma(k)))*imag(Am(k)*x(:,k)')*D_N*s);
    sum2=sum2-((gama/w0)^2/sigma(k))*real(Am(k)*x(:,k)')*D_N*D_N*s;
  end % for k=1:M
  Tlmpu = sum1^2 + sum2;
end % function Tlmpu = LMPU_FUN(x, sigma, Am, w0)
function Tglmpu = GLMPU_FUN(x, w0)

% Inputs:
% x - A matrix composed in the following way:
% x = [x_1, x_2, ..., x_M], where x_m is a complex
% observation vector in dimension N measured by ...
% the mth sensor
% for all m = 1, ..., M. The measurement model is:
% x_m = A_m * s(w) + noise_m.
% w0 - The known nominal system frequency under normal
% conditions.
% Output:
% Tglmpu - The value of GLMPU test given x, and w0.
% Constants of the model
persistent n
if isempty(n)
    n=(0:1:length(x(:,1))-1).';
end % if isempty(n)
persistent N
if isempty(N)
    N=length(n);
end % if isempty(N)
persistent gama
if isempty(gama)
    gama=2*pi/N;
end % if isempty(gama)
persistent M
if isempty(M)
    M=size(x,2);
end % if isempty(M)
persistent D_N
if isempty(D_N)
    D_N=diag(n);
end % if isempty(D_N)
persistent s
if isempty(s)
    s=exp(1i*gama*n);
end % if isempty(s)
persistent Ps
if isempty(Ps)
    Ps=(1/N)*s*(s')
end % if isempty(Ps)
persistent Psc
if isempty(Psc)
    Psc=eye(size(Ps))-Ps
end % if isempty(Psc)

% GLMPU algorithm
sum1=0;
sum2=0;
for k=1:M
    sig_m=real((x(:,k)')*Psc*x(:,k));
    sum1=sum1-(gama/(w0*sig_m))*imag((x(:,k)')*D_N*s*(s')*x(:,k));
    sum2=sum2-((gama/w0)^2*sig_m-1)*real((x(:,k)')*D_N*D_N*s*(s')*x(:,k));
end % for k=1:M
Tglmpu=sum1^2+sum2;
end % function Tglmpu = GLMPU_FUN(x, w0)
Ethics Statement

The authors declare that this work has been done according to the ethical requirements for publication in Data in Brief.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships which have, or could be perceived to have, influenced the work reported in this article.

Acknowledgments

This research was partially supported by The Israel Science Foundation (grant no. 1173/16) and by the Israeli Ministry of National Infrastructure, Energy and Water Resources.

References

[1] E.L. Lehmann, J.P. Romano, Testing Statistical Hypotheses, Springer Science & Business Media, 2006.
[2] E. Levy, T. Routtenberg, Low-complexity detection of small frequency deviations by the generalized LMPU test, Signal Process. 180 (2021), doi:10.1016/j.sigpro.2020.107851.