The Analysis of Periodic Point Processes

Stephen D. Casey

American University
Washington, DC USA
scasey@american.edu

The 36th ASA Quality & Productivity Research Conference
Talk to Honor Prof. Benjamin Kedem
June 11th, 2019

Acknowledgments

- Author's research partially supported by U.S. Army Research Office Scientific Services Program, administered by Battelle (TCN 06150, Contract DAAD19-02-D-0001) and Air Force Office of Scientific Research Grant Number FA9550-12-1-0430.
- Simulations for the MEA algorithm is joint work with Brian Sadler of the Army Research Laboratories.
- Simulations for the EQUIMEA algorithm is joint work with Kevin Duke of American University. Special thanks to Kevin for allowing us to experimentally verify the EQUIMEA.
- It is a honor to present these results at a session acknowledging a truly wonderful mentor – Prof. Benjamin Kedem.

Motivation: Signal and Image Signatures

- Radar or Sonar.
- Bit synchronization in communications.
- Unreliable measurements in a fading communications channel.
- Biomedical signatures.
- Compute the "jump times" of a pseudorandomly occurring change in the carrier frequency of a "frequency hopping" radio, where the change rate is governed by a shift register output. In this case it is desired to find the underlying fundamental period τ.

Data Models

Assumption – noisy signal data is set of **event times**
"Time of Arrival" (TOA's)

$$s(t) + \eta(t)$$

with (large) gaps in the data.

Questions – $s(t)$ periodic? period $\tau =$? Are there multiple periods $\tau_k =$? If so, what are they? How do we deinterleave the signals?

Single Periodic Generator

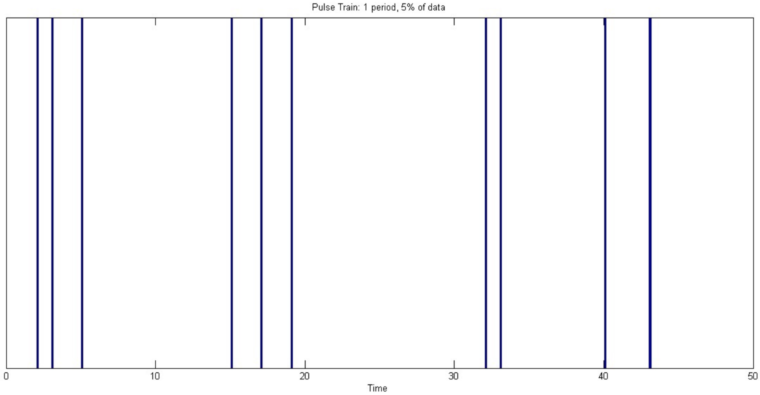


Figure: One Period – Original Data

Two Periodic Generators

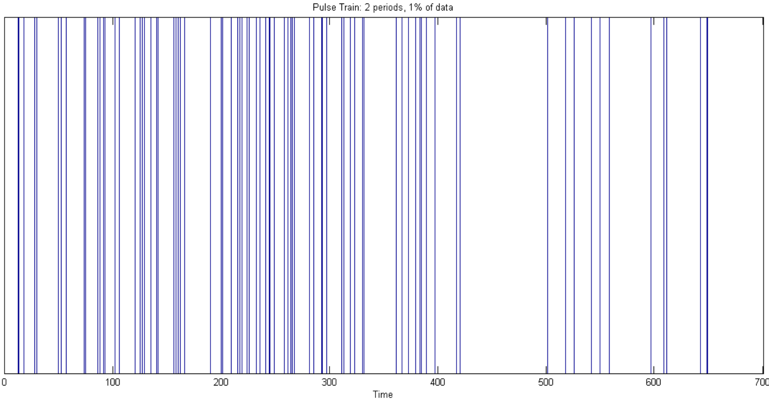


Figure: Two Periods – Original Data

Three Periodic Generators

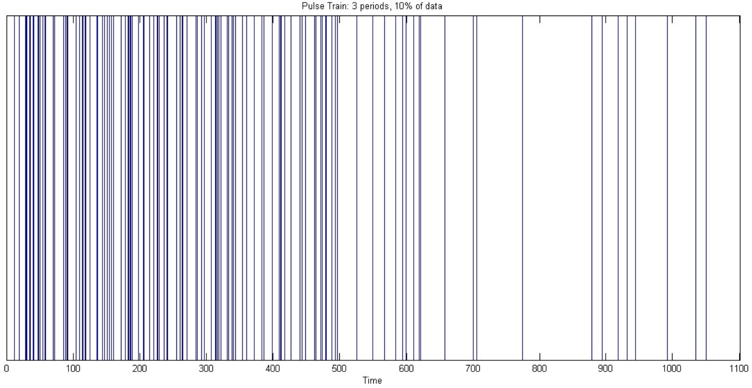
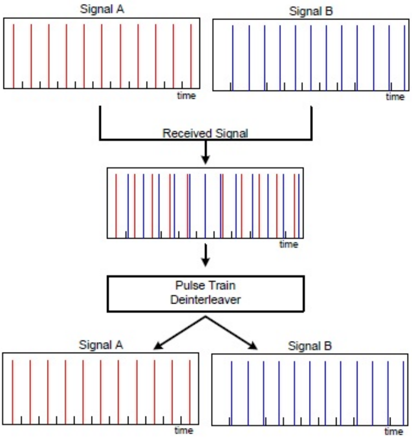


Figure: Three Periods – Original Data

Deinterleaving Multiple Signals



Mathematical Models – Single Period

Finite set of real numbers

$$S = \{s_j\}_{j=1}^{n}\,, \text{ with } s_j = k_j\tau + \varphi + \eta_j\,,$$

where

- τ (the period) is a fixed positive real number to be determined
- k_j's are non-repeating positive integers (natural numbers)
- φ (the phase) is a real random variable uniformly distributed over the interval $[0, \tau)$
- η_j's (the noise) are zero-mean independent identically distributed (iid) error terms. We assume that the η_j's have a symmetric probability density function (pdf), and that

$$|\eta_j| \le \eta_0 \le \frac{\tau}{2} \text{ for all } j\,,$$

where η_0 is an *a priori* noise bound.

Approaches to the Analysis

- The data can be thought of as a set of event times of a periodic process, which generates a zero-one time series or delta train with additive jitter noise $\eta(t)$ –

$$s(t) = \sum_{j=1}^{n} \delta(t - ((k_j\tau + \varphi) + \eta(t)))\,.$$

- Another model – Let $f(t) = \sin(\frac{\pi}{\tau}(t - \varphi))$ and $S = \{$occurrence time of noisy zero-crossings of f with missing observations$\}$.
- The k_j's determine the best procedure for analyzing this data.
- Given a sequence of consecutive k_j's, use least squares.
- Fourier analytic methods, e.g., Wiener's periodogram, work with some missing observations, but when the percentage of missing observations is too large ($> 50\%$), they break down.
- Number theoretic methods can work with very sparse data sets ($> 90\%$ missing observations). Trade-off – low noise – number theory vs. higher noise – combine Fourier with number theory.

The Structure of Randomness over $\mathbb{Z}$

Theorem

Given n $(n \geq 2)$ *"randomly chosen" positive integers* $\{k_1, \ldots, k_n\}$,

$$P\{\gcd(k_1, \ldots, k_n) = 1\} \longrightarrow 1^- \text{ quickly! as } n \longrightarrow \infty \,.$$

Theorem

Given n $(n \geq 2)$ *"randomly chosen" positive integers* $\{k_1, \ldots, k_n\}$,

$$P\{\gcd(k_1, \ldots, k_n) = 1\} = [\zeta(n)]^{-1} \,.$$

An Algorithm for Finding τ

$$S = \{s_j\}_{j=1}^{n}, \text{ with } s_j = k_j\tau + \varphi + \eta_j$$

Let $\widehat{\tau}$ denote the value the algorithm gives for τ, and let "$\longleftarrow$" denote *replacement*, e.g., "$a \longleftarrow b$" means that the value of the variable a is to be replaced by the current value of the variable b.

Initialize: Sort the elements of S in descending order. Set $\texttt{iter} = 0$.

1.) [Adjoin 0 after first iteration.] If $\texttt{iter} > 0$, then $S \longleftarrow S \cup \{0\}$.

2.) [Form the new set with elements $(s_j - s_{j+1})$.] Set $s_j \longleftarrow (s_j - s_{j+1})$.

3.) [Sort.] Sort the elements in descending order.

4.) [Eliminate zero(s).] If $s_j = 0$, then $S \longleftarrow S \setminus \{s_j\}$.

5.) The algorithm terminates if S has only one element s_1. Declare $\widehat{\tau} = s_1$. If not, $\texttt{iter} \longleftarrow (\texttt{iter} + 1)$. Go to **1.)**.

Paper and Pencil Computation

Initial	Iter 1	Iter 2	Iter 3	Iter 4
$\phi + 11\tau$	3τ	2τ	τ	$\tau = \widehat{\tau}$
$\phi + 9\tau$	3τ	τ	τ	
$\phi + 6\tau$	2τ			
$\phi + 3\tau$				

Simulation Results

"To err is human. To really screw up, you need a computer."
The Murphy Institute

Assume $\tau = 1$.

- Estimates and their standard deviations are based on averaging over 100 Monte-Carlo runs
- n = number of data points, *iter* = average number of iterations required for convergence, and %*miss* = average number of missing observations
- Estimates of τ are labeled $\widehat{\tau}$, and $std(\widehat{\tau})$ is the experimental standard deviation
- Threshold value of $\eta_0 = 0.35\tau = 0.35$ was used

Simulation Results, Cont'd

1.) *Noise-free estimation.*

Results from simulating noise-free estimation of τ.

n	M	%*miss*	*iter*	τ	2τ	3τ	$> 3\tau$
10	10^1	81.69	3.3	100%	0	0	0
10	10^2	97.92	10.5	100	0	0	0
10	10^3	99.80	46.5	100	0	0	0
10	10^4	99.98	316.2	100	0	0	0
10	10^5	99.998	2638.7	100	0	0	0
4	10^2	97.84	15.2	82%	12	4	2
6	10^2	97.82	14.2	97	3	0	0
8	10^2	97.80	10.2	98	1	1	0
10	10^2	97.78	10.2	99	1	0	0
12	10^2	97.76	8.6	100	0	0	0
14	10^2	97.75	7.4	100	0	0	0

Simulation Results, Cont'd

2.) *Uniformly distributed noise.*

Results from estimation of τ from noisy measurements.

n	M	Δ	$\%miss$	$iter$	$\widehat{\tau}$	$std(\widehat{\tau})$
10	10^1	10^{-3}	81.37	4.35	0.9987	0.0005
10	10^2	10^{-3}	97.88	9.67	0.9980	0.0010
50	10^3	10^{-3}	99.80	16.0	0.9969	0.0028
10	10^1	10^{-2}	80.85	4.38	0.9888	0.0046
10	10^1	10^{-2}	81.94	4.45	0.9883	0.0051
10	10^1	10^{-1}	81.05	4.33	0.8857	0.0432

π, the Primes, and Probability

- Let $\mathbb{P} = \{p_1, p_2, p_3, \ldots\} = \{2, 3, 5, \ldots\}$ be the set of all prime numbers.

π, the Primes, and Probability

- Let $\mathbb{P} = \{p_1, p_2, p_3, \ldots\} = \{2, 3, 5, \ldots\}$ be the set of all prime numbers.
-
> *"God gave us the integers. The rest is the work of man."*
> KRONECKER

π, the Primes, and Probability

- Let $\mathbb{P} = \{p_1, p_2, p_3, \ldots\} = \{2, 3, 5, \ldots\}$ be the set of all prime numbers.
- *"God gave us the integers. The rest is the work of man."* KRONECKER
- *"... the Euler formulae (1736)*

$$\zeta(z) = \sum_{n=1}^{\infty} n^{-z} = \prod_{j=1}^{\infty} \frac{1}{1-(p_j)^{-z}}, \ \Re(z) > 1$$

was introduced to us at school, as a joke." LITTLEWOOD

π, the Primes, and Probability, Cont'd

- *"Euclid's algorithm is found in Book 7, Proposition 1 and 2 of his Elements (c.300 B.C.). We might call it the grand daddy of all algorithms, because it is the oldest nontrivial algorithm that has survived to the present day."* KNUTH

π, the Primes, and Probability, Cont'd

- > *"Euclid's algorithm is found in Book 7, Proposition 1 and 2 of his Elements (c.300 B.C.). We might call it the grand daddy of all algorithms, because it is the oldest nontrivial algorithm that has survived to the present day."* KNUTH

- The Euclidean algorithm is a division process for the integers $\mathbb{Z}$. The algorithm is based on the property that, given two positive integers a and b, $a > b$, there exist two positive integers q and r such that $a = q \cdot b + r$, $0 \leq r < b$. If $r = 0$, we say that b divides a.

π, the Primes, and Probability, Cont'd

- > *"Euclid's algorithm is found in Book 7, Proposition 1 and 2 of his Elements (c.300 B.C.). We might call it the grand daddy of all algorithms, because it is the oldest nontrivial algorithm that has survived to the present day."* KNUTH

  - The Euclidean algorithm is a division process for the integers $\mathbb{Z}$. The algorithm is based on the property that, given two positive integers a and b, $a > b$, there exist two positive integers q and r such that $a = q \cdot b + r$, $0 \leq r < b$. If $r = 0$, we say that b divides a.
  - The Euclidean algorithm yields the greatest common divisor of two (or more) elements of $\mathbb{Z}$. The *greatest common divisor* of two integers a and b, denoted by $\gcd(a, b)$, is the the largest integer that evenly divides both integers.

π, the Primes, and Probability, Cont'd

Theorem (Fundamental Theorem of Arithmetic)

Every positive integer can be written uniquely as the product of primes, with the prime factors in the product written in the order of nondecreasing size.

- If $\gcd(a, b) = 1$, we say that the numbers are relatively prime. This means that a and b share no common prime factors in their prime factorization.
- $\gcd(k_1, \ldots, k_n)$ is the greatest common divisor of the set $\{k_j\}$, i.e., the product of the powers of all prime factors p that divide each k_j.
- Examples
    - $\gcd(3, 7) = 1$
    - $\gcd(3, 6) = 3$
    - $\gcd(35, 21) = 7$
    - $\gcd(35, 21, 15) = 1$

π, the Primes, and Probability, Cont'd

Theorem

Given n $(n \geq 2)$ "randomly chosen" positive integers $\{k_1, \ldots, k_n\}$,

$$P\{\gcd(k_1, \ldots, k_n) = 1\} = [\zeta(n)]^{-1}.$$

π, the Primes, and Probability, Cont'd

- Heuristic argument for this "theorem." Given randomly distributed positive integers, by the Law of Large Numbers, about 1/2 of them are even, 1/3 of them are multiples of three, and $1/p$ are a multiple of some prime p. Thus, given n independently chosen positive integers,

$$\begin{aligned} P\{p|k_1, p|k_2, \ldots, \text{and } p|k_n\} &= \\ \text{(Independence)} & \\ P\{p|k_1\} \cdot P\{p|k_2\} \cdot \ldots \cdot P\{p|k_n\} &= \\ 1/(p) \cdot 1/(p) \cdot \ldots \cdot 1/(p) &= \\ 1/(p)^n. & \end{aligned}$$

Therefore,

$$P\{p \not| k_1, p \not| k_2, \ldots, \text{and } p \not| k_n\} = 1 - 1/(p)^n \, .$$

π, the Primes, and Probability, Cont'd

- By the Fundamental Theorem of Arithmetic, every integer has a unique representation as a product of primes. Combining that theorem with the definition of gcd, we get

$$P\{\gcd(k_1, \ldots, k_n) = 1\} = \prod_{j=1}^{\infty} 1 - 1/(p_j)^n \,,$$

where p_j is the j^{th} prime.

π, the Primes, and Probability, Cont'd

- By the Fundamental Theorem of Arithmetic, every integer has a unique representation as a product of primes. Combining that theorem with the definition of gcd, we get

$$P\{\gcd(k_1, \ldots, k_n) = 1\} = \prod_{j=1}^{\infty} 1 - 1/(p_j)^n \,,$$

  where p_j is the j^{th} prime.
- But, by Euler's formula,

$$\zeta(z) = \prod_{j=1}^{\infty} \frac{1}{1 - (p_j)^{-z}} \,,\ \Re(z) > 1 \,.$$

  Therefore,

$$P\{\gcd(k_1, \ldots, k_n) = 1\} = 1/(\zeta(n)) \,.$$

π, the Primes, and Probability, Cont'd

This argument breaks down on the first line. Any uniform distribution on the positive integers would have to be identically zero. The merit in the argument lies in the fact that it gives an indication of how the zeta function plays a role in the problem.

Let $\mathrm{card}\{\cdot\}$ denote cardinality of the set $\{\cdot\}$, and let $\{1,\ldots,\ell\}^n$ denote the sublattice of positive integers in $\mathbb{R}^n$ with coordinates c such that $1 \leq c \leq \ell$. Therefore, $N_n(\ell) = \mathrm{card}\{(k_1,\ldots,k_n) \in \{1,\ldots,\ell\}^n : \gcd(k_1,\ldots,k_n) = 1\}$ is the number of relatively prime elements in $\{1,\ldots,\ell\}^n$.

Theorem (MEA Theorem, C (1998), ...)

Let $N_n(\ell) = \mathrm{card}\{(k_1,\ldots,k_n) \in \{1,\ldots,\ell\}^n : \gcd(k_1,\ldots,k_n) = 1\}$. *For* $n \geq 2$, *we have that*

$$\lim_{\ell\to\infty} \frac{N_n(\ell)}{\ell^n} = [\zeta(n)]^{-1}\,.$$

π, the Primes, and Probability, Cont'd

Brief Discussion of Proof : Let $\lfloor x \rfloor$ denote the floor function of x, namely

$$\lfloor x \rfloor = \max_{k \leq x} \{k : k \in \mathbb{Z}\}.$$

$$N_n(\ell) = \ell^n - \sum_{p_i} \left(\left\lfloor \frac{\ell}{p_i} \right\rfloor \right)^n + \sum_{p_i < p_j} \left(\left\lfloor \frac{\ell}{p_i \cdot p_j} \right\rfloor \right)^n - \sum_{p_i < p_j < p_k} \left(\left\lfloor \frac{\ell}{p_i \cdot p_j \cdot p_k} \right\rfloor \right)^n + \cdots$$

Convergence is demonstrated by a sequence of careful estimates, use of Möbius Inversion, and more careful estimates.

π, the Primes, and Probability, Cont'd

The counting formula is seen as follows. Choose a prime number p_i. The number of integers in $\{1, \ldots, \ell\}$ such that p_i divides an element of that set is $\left\lfloor \frac{\ell}{p_i} \right\rfloor$. (Note that is possible to have $p_i > \ell$, because $\left\lfloor \frac{\ell}{p_i} \right\rfloor = 0$.) Therefore, the number of n-tuples $(k_1, \ldots, k_n)$ contained in the lattice $\{1, \ldots, \ell\}^n$ such that p_i divides every integer in the n-tuple is

$$\left(\left\lfloor \frac{\ell}{p_i} \right\rfloor \right)^n .$$

Next, if $p_i \cdot p_j$ divides an integer k, then $p_i | k$ and $p_j | k$. Therefore, the number of n-tuples $(k_1, \ldots, k_n)$ contained in the lattice $\{1, \ldots, \ell\}^n$ such that p_i or p_j or their product divide every integer in the n-tuple is

$$\left(\left\lfloor \frac{\ell}{p_i} \right\rfloor \right)^n + \left(\left\lfloor \frac{\ell}{p_j} \right\rfloor \right)^n - \left(\left\lfloor \frac{\ell}{p_i \cdot p_j} \right\rfloor \right)^n ,$$

where the last term is subtracted so that we do not count the same numbers twice (in a simple application of the inclusion-exclusion principle).

π, the Primes, and Probability, Cont'd

Each term is convergent –

$$\frac{1}{\ell^n} \sum_{p_i < \ldots < p_k} \left(\left\lfloor \frac{\ell}{p_i \cdot \cdots \cdot p_k} \right\rfloor \right)^n \le \frac{1}{\ell^n} \sum_{p_i < \ldots < p_k \le \ell} \left(\frac{\ell}{p_i \cdot p_j \cdot \cdots \cdot p_k} \right)^n$$

$$= \sum_{p_i < \ldots < p_k \le \ell} \left(\frac{1}{p_i \cdot \cdots \cdot p_k} \right)^n = \left(\sum_{p \le \ell} \frac{1}{p^n} \right)^k$$

$$\le \left(\sum_{p \text{ prime}} \frac{1}{p^n} \right)^k \le \left(\sum_{j=2}^{\infty} \frac{1}{j^n} \right)^k .$$

Since $n \ge 2$, this series is convergent.

π, the Primes, and Probability, Cont'd

Now, let

$$M_k = \left(\sum_{j=2}^{\infty} \frac{1}{j^n} \right)^k, \text{ for } k = 2, 3, \ldots .$$

By noting that since $n \geq 2$ and the sum is over $j \in \mathbb{N} \setminus \{1\}$, we get

$$0 < \sum_j \frac{1}{j^n} \leq \left(\frac{\pi^2}{6} - 1 \right) < 1 .$$

Since the k^{th} term in the expansion of $N_n(\ell)/\ell^n$ is dominated by M_k and since

$$\sum_{k=0}^{\infty} M_k \leq \sum_{k=0}^{\infty} \left(\frac{\pi^2}{6} - 1 \right)^k = \frac{6}{(12 - \pi^2)}$$

is convergent, the series converges absolutely.

π, the Primes, and Probability, Cont'd

Euler showed that

$$1 - \sum_{p_i} \frac{1}{p_i^n} + \sum_{p_i<p_j} \frac{1}{(p_i \cdot p_j)^n} - \sum_{p_i<p_j<p_k} \frac{1}{(p_i \cdot p_j \cdot p_k)^n} + \cdots$$
$$= \sum_{m} \frac{\mu(m)}{m^n} = [\zeta(n)]^{-1}.$$

where the last sum is over $m \in \mathbb{N}$. For $n \geq 2$, this series is absolutely convergent. □

π, the Primes, and Probability, Cont'd

Theorem

Let $\omega \in (1, \infty)$. *Then* $\lim_{\omega \to \infty} [\zeta(\omega)]^{-1} = 1$, *converging to 1 from below faster than* $1/(1 - 2^{1-\omega})$.

Proof : Since $\zeta(\omega) = \sum_{n=1}^{\infty} n^{-\omega}$ and $\omega > 1$,

$$
\begin{aligned}
1 \leq \zeta(\omega) &= 1 + \frac{1}{2^\omega} + \frac{1}{3^\omega} + \frac{1}{4^\omega} + \frac{1}{5^\omega} + \cdots \\
&\leq 1 + \frac{1}{2^\omega} + \frac{1}{2^\omega} + \underbrace{\frac{1}{4^\omega} + \cdots + \frac{1}{4^\omega}}_{4-\text{times}} + \underbrace{\frac{1}{8^\omega} + \cdots + \frac{1}{8^\omega}}_{8-\text{times}} + \cdots \\
&= \sum_{k=0}^{\infty} \left(\frac{2}{2^\omega} \right)^k = \frac{1}{1 - \frac{2}{2^\omega}} = \frac{1}{1 - 2^{1-\omega}} .
\end{aligned}
$$

As $\omega \longrightarrow \infty$, $(1 - 2^{1-\omega}) \longrightarrow 1^+$. Thus, $[\zeta(\omega)]^{-1} \longrightarrow 1^-$ as $\omega \longrightarrow \infty$. □

The Modified Euclidean Algorithm (MEA)

$$S = \{s_j\}_{j=1}^{n}, \text{ with } s_j = k_j\tau + \varphi + \eta_j$$

Let $\widehat{\tau}$ denote the value the algorithm gives for τ, and let "$\longleftarrow$" denote *replacement*.

Initialize: Sort the elements of S in descending order. Set $\texttt{iter} = 0$.

1.) [Adjoin 0 after first iteration.] If $\texttt{iter} > 0$, then $S \longleftarrow S \cup \{0\}$.

2.) [Form the new set with elements $(s_j - s_{j+1})$.] Set $s_j \longleftarrow (s_j - s_{j+1})$.

3.) [Sort.] Sort the elements in descending order.

4.) [Eliminate zero(s).] If $s_j = 0$, then $S \longleftarrow S \setminus \{s_j\}$.

5.) The algorithm terminates if S has only one element s_1. Declare $\widehat{\tau} = s_1$. If not, $\texttt{iter} \longleftarrow (\texttt{iter} + 1)$. Go to **1.)**.

The Modified Euclidean Algorithm (MEA), Cont'd

- Euclidean algorithm for $\{k_j\}_{j=1}^n \subset \mathbb{N}$, $\tau > 0$ –

Lemma

$$\gcd(k_1\tau, \ldots, k_n\tau) = \tau \gcd(k_1, \ldots, k_n)\,.$$

The Modified Euclidean Algorithm (MEA), Cont'd

- Euclidean algorithm for $\{k_j\}_{j=1}^n \subset \mathbb{N}$, $\tau > 0$ –

Lemma

$$\gcd(k_1\tau, \ldots, k_n\tau) = \tau \gcd(k_1, \ldots, k_n)\,.$$

- What if "integers are noisy?"

The Modified Euclidean Algorithm (MEA), Cont'd

- Euclidean algorithm for $\{k_j\}_{j=1}^n \subset \mathbb{N}$, $\tau > 0$ –

Lemma

$$\gcd(k_1\tau, \ldots, k_n\tau) = \tau \gcd(k_1, \ldots, k_n)\,.$$

- What if "integers are noisy?"
- Remainder terms could be noise, and thus could be non-zero numbers arbitrarily close to zero. Subsequent steps in the procedure may involve dividing by such numbers, which would result in arbitrarily large numbers. The standard algorithm is unstable under perturbation by noise.

The Modified Euclidean Algorithm (MEA), Cont'd

- Euclidean algorithm for $\{k_j\}_{j=1}^n \subset \mathbb{N}$, $\tau > 0$ –

Lemma

$$\gcd(k_1\tau, \ldots, k_n\tau) = \tau \gcd(k_1, \ldots, k_n)\,.$$

- What if "integers are noisy?"
- Remainder terms could be noise, and thus could be non-zero numbers arbitrarily close to zero. Subsequent steps in the procedure may involve dividing by such numbers, which would result in arbitrarily large numbers. The standard algorithm is unstable under perturbation by noise.
- **Solution :** Replace division with subtraction, and threshold/average/filter/transform to eliminate noise.

The Modified Euclidean Algorithm (MEA), Cont'd

Lemma (The Key Lemma)

The MEA preserves the gcd, *i.e.,*

$$\gcd(k_1, \ldots, k_n) = \gcd((k_1 - k_2), (k_2 - k_3), \ldots, (k_{n-1} - k_n), k_n) .$$

The Modified Euclidean Algorithm (MEA), Cont'd

- Combining the MEA Theorem with the Lemmas above gives the theoretical underpinnings of the Modified Euclidean Algorithm.

Corollary

Let $n \geq 2$. Given a randomly chosen n-tuple of positive integers $(k_1, \ldots, k_n) \in \{1, \ldots, \ell\}^n$,

$$\gcd(k_1\tau, \ldots, k_n\tau) \longrightarrow \tau\,,$$

with probability $[\zeta(n)]^{-1}$ as $\ell \longrightarrow \infty$.

The Modified Euclidean Algorithm (MEA), Cont'd

- Combining the MEA Theorem with the Lemmas above gives the theoretical underpinnings of the Modified Euclidean Algorithm.

Corollary

Let $n \geq 2$. Given a randomly chosen n-tuple of positive integers $(k_1, \ldots, k_n) \in \{1, \ldots, \ell\}^n$,

$$\gcd(k_1\tau, \ldots, k_n\tau) \longrightarrow \tau\,,$$

with probability $[\zeta(n)]^{-1}$ as $\ell \longrightarrow \infty$.

- Moreover, the estimate

$$(1 - 2^{1-\omega})^{-1} \leq [\zeta(\omega)]^{-1} \leq 1$$

shows that the algorithm very likely produces this value in the noise-free case or with minimal noise with as few as 10 data elements.

Mathematical Models – Multiple Periods

Our data model is the union of M copies of $S = \{s_{i,j}\}_{j=1}^{n_i}$ with $s_j = k_j\tau + \varphi + \eta_j$,, each with different periods or "generators" $\Gamma = \{\tau_i\}$, k_{ij}'s and phases. Let $\tau_M = \max_i\{\tau_i\}$ and $\tau_m = \min_i\{\tau_i\}$. Then our data is

$$\mathcal{S} = \bigcup_{i=1}^{M} \left\{ \varphi_i + k_{ij}\tau_i + \eta_{ij} \right\}_{j=1}^{n_i},$$

- where n_i is the number of elements from the i^{th} generator
- the different periods or "generators" are $\Gamma = \{\tau_i\}$
- $\{k_{ij}\}$ is a linearly increasing sequence of natural numbers with missing observations
- φ_i (the phases) are random variables uniformly distributed in $[0, \tau_i)$
- η_{ij}'s are zero-mean iid Gaussian with standard deviation $3\sigma_{ij} < \tau/2$
- We think of the data as events from M periodic processes, and represent it, after reindexing, as $\mathcal{S} = \{\alpha_l\}_{l=1}^{N}$, where $N = \sum_i n_i$.

The Structure of Randomness over $[0, T]$

Theorem (Weyl's Equidistribution Theorem)

Let ϕ be an irrational number, $j \in \mathbb{N}$. Let

$$\langle j\phi \rangle = j\phi - \lfloor j\phi \rfloor \ .$$

Then given a, b, $0 \leq a < b < 1$,

$$\frac{1}{n}\mathrm{card}\left\{1 \leq j \leq n : \langle j\phi \rangle \in [a, b]\right\} \longrightarrow (b - a)$$

as $n \longrightarrow \infty$.

The Structure of Randomness over $[0, T)$

Assuming only minimal knowledge of the range of $\{\tau_i\}$, namely bounds T_L, T_U such that $0 < T_L \leq \tau_i \leq T_U$, we phase wrap the data by the mapping

$$\Phi_\rho(\alpha_l) = \left\langle \frac{\alpha_l}{\rho} \right\rangle = \frac{\alpha_l}{\rho} - \left\lfloor \frac{\alpha_l}{\rho} \right\rfloor,$$

where $\rho \in [T_L, T_U]$, and $\lfloor \cdot \rfloor$ is the floor function. Thus $\langle \cdot \rangle$ is the fractional part, and so $\Phi_\rho(\alpha_l) \in [0, 1)$.

Definition

A sequence of real random variables $\{x_j\} \subset [0, 1)$ is essentially uniformly distributed in the sense of Weyl if given a, b, $0 \leq a < b < 1$,

$$\frac{1}{n}\mathrm{card}\left\{1 \leq j \leq n : x_j \in [a, b]\right\} \longrightarrow (b - a)$$

as $n \longrightarrow \infty$ almost surely.

Applying Weyl's Theorem

We assume that for each i, $\{k_{ij}\}$ is a linearly increasing infinite sequence of natural numbers with missing observations such that

$$k_{ij} \longrightarrow \infty \text{ as } j \longrightarrow \infty\,.$$

Weyl's Theorem applies asymptotically.

Theorem (C (2019))

For almost every choice of ρ (in the sense of Lebesgue measure) $\Phi_\rho(\alpha_l)$ is essentially uniformly distributed in the sense of Weyl.

Applying Weyl's Theorem, Cont'd

- **Moreover, the set of ρ's for which this is not true are rational multiples of $\{\tau_i\}$. Additionally, since $\mathbb{Q}$ is countable (and thus measure zero), these occur (in the sense of $\{\tau_i\}$) with probability zero. Therefore, except for those values, $\Phi_\rho(\alpha_{ij})$ is essentially uniformly distributed in $[T_L, T_U)$.** The values at which $\Phi_\rho(\alpha_{ij}) = 0$ almost surely are $\rho \in \{\tau_i/n : n \in \mathbb{N}\}$. These values of ρ cluster at zero, but spread out for lower values of n.
- We phase wrap the data by computing modulus of the spectrum, i.e., compute

$$|S_{iter}(\tau)| = \left|\sum_{j=1}^{N} e^{(2\pi i s(j)/\tau)}\right|.$$

- The values of

$$|S_{iter}(\tau)|$$

will have peaks at the periods τ_j and their harmonics $(\tau_j)/k$.

The EQUIMEA Algorithm – One Period

The EQUIMEA Algorithm – One Period

$$S = \{s_j\}_{j=1}^{n}\ , \text{ with } s_j = k_j\tau + \varphi + \eta_j$$

Initialize: Sort the elements of S in descending order. Form the new set with elements $(s_j - s_{j+1})$. Set $s_j \longleftarrow (s_j - s_{j+1})$. (Note, this eliminates the phase φ.) Let $\widehat{\tau}$ denote the value the algorithm gives for τ, and let "$\longleftarrow$" denote *replacement*.

The EQUIMEA Algorithm – One Period

The EQUIMEA Algorithm – One Period

1.) [Adjoin 0 after first iteration.] $S_{iter} \longleftarrow S \cup \{0\}$.

2.) [Sort.] Sort the elements of S_{iter} in descending order.

3.) [Compute all differences.] Set $S_{iter} = \bigcup(s_j - s_k)$ for all j, k with $s_j > s_k$.

4.) [Eliminate zero(s).] If $s_j = 0$, then $S_{iter} \longleftarrow S_{iter} \setminus \{s_j\}$.

5.) [Adjoin previous iteration.] Form $S_{iter} \longleftarrow S_{iter} \cup S_{iter-1}$.

6.) [Compute spectrum.] Compute

$$|S_{iter}(\tau)| = \left| \sum_{j=1}^{N} e^{(2\pi i s(j)/\tau)} \right|.$$

7.) [Threshold.] Choose the largest peak. Label it as τ_{iter}

8.) The algorithm terminates if $|\tau_{iter} - \tau_{iter-1}| < \mathrm{Error}$. Declare $\widehat{\tau} = \tau_{iter}$. If not, $\mathtt{iter} \longleftarrow (\mathtt{iter} + 1)$. Go to **1.)**.

The EQUIMEA Algorithm – One Period, Cont'd

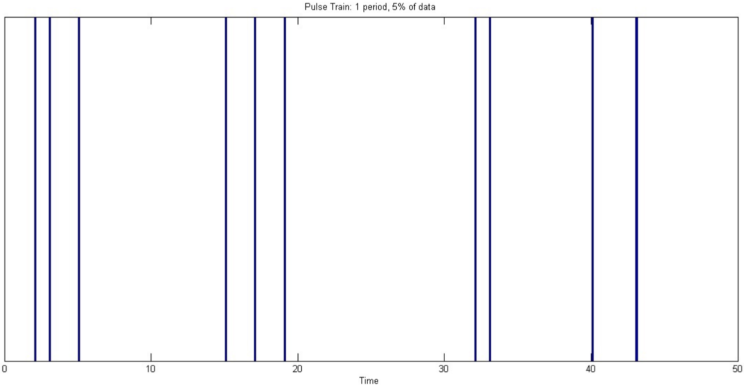


Figure: EQUIMEA One Period Tau – Original Data

The EQUIMEA Algorithm – One Period, Cont'd

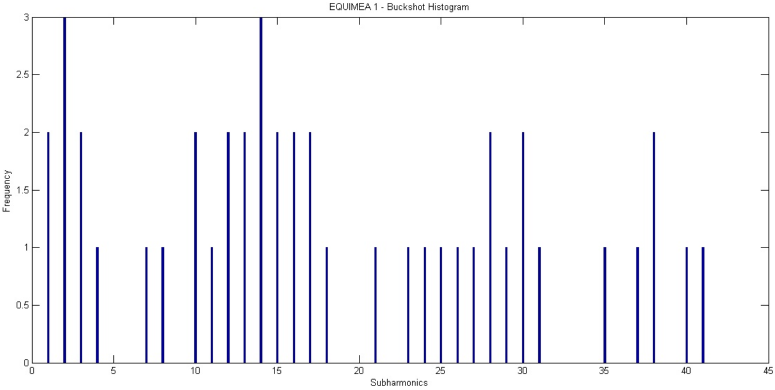


Figure: EQUIMEA One Period Tau – One Iteration

The EQUIMEA Algorithm – One Period, Cont'd

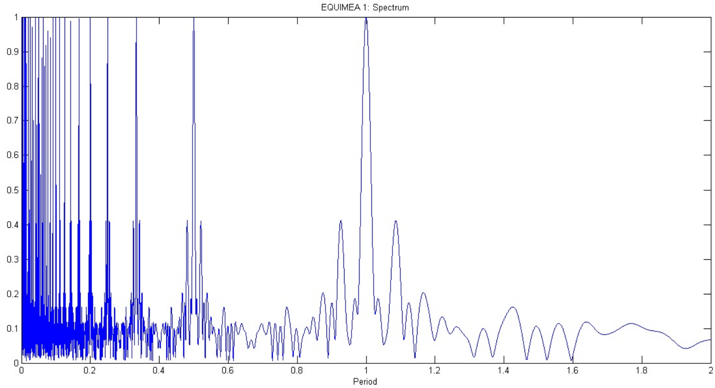


Figure: EQUIMEA One Period Tau – One Iteration – Spectrum

The EQUIMEA Algorithm – One Period, Cont'd

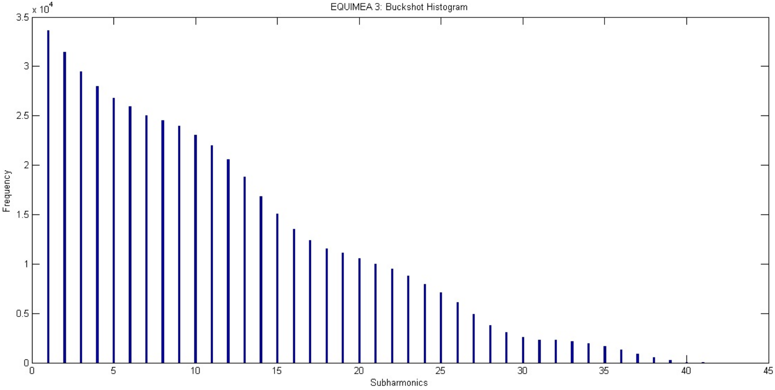


Figure: EQUIMEA One Period Tau – Third Iteration

The EQUIMEA Algorithm – One Period, Cont'd

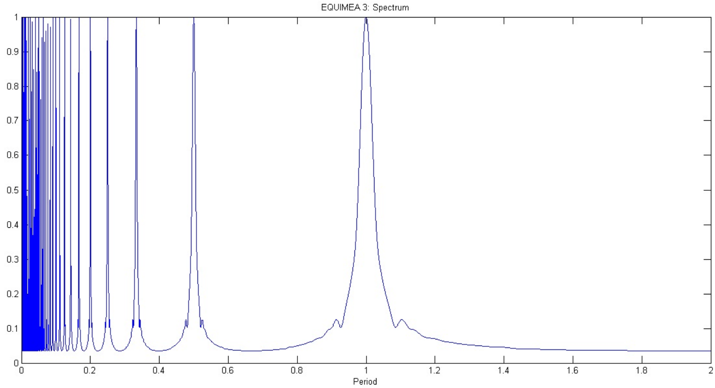
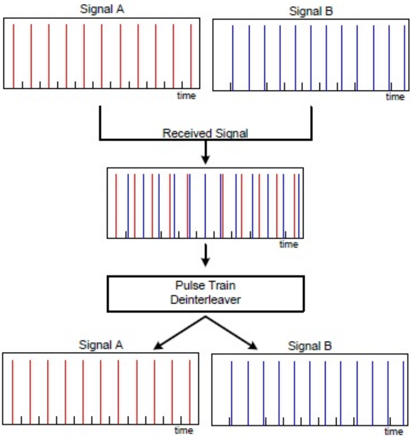


Figure: EQUIMEA One Period Tau – Third Iteration – Spectrum

Deinterleaving Multiple Signals



The EQUIMEA Algorithm – Multiple Periods

The EQUIMEA Algorithm – Multiple Periods

Our data model is the union of M copies of $S = \{s_{i,j}\}_{j=1}^{n_i}$ with $s_j = k_j\tau + \varphi + \eta_j$,, each with different periods or "generators" $\Gamma = \{\tau_i\}$, k_{ij}'s and phases. Let $\tau_M = \max_i\{\tau_i\}$ and $\tau_m = \min_i\{\tau_i\}$. Then our data is

$$\mathcal{S} = \bigcup_{i=1}^{M} \left\{\varphi_i + k_{ij}\tau_i + \eta_{ij}\right\}_{j=1}^{n_i},$$

Let $\widehat{\tau}$ denote the value the algorithm gives for τ, and let "$\longleftarrow$" denote *replacement*.

After reindexing, $\mathcal{S} = \{\alpha_l\}_{l=1}^{N}$, where $N = \sum_i n_i$.

Initialize: Sort the elements of S in descending order. Form the new set with elements $(s_l - s_{l+1})$. Set $s_l \longleftarrow (s_l - s_{l+1})$. (Note, this eliminates the phase φ.) Set $\texttt{iter} = 1$, $i = 1$, and $\texttt{Error}$. Go to **1.)**

The EQUIMEA Algorithm – Multiple Periods

1.) [Adjoin 0 after first iteration.] $S_{iter} \longleftarrow S \cup \{0\}$.

2.) [Sort.] Sort the elements of S_{iter} in descending order.

3.) [Compute all differences.] Set $S_{iter} = \bigcup (s_j - s_k)$ with $s_j > s_k$.

4.) [Eliminate zero(s).] If $s_j = 0$, then $S_{iter} \longleftarrow S_{iter} \setminus \{s_j\}$.

5.) [Adjoin previous iteration.] Form $S_{iter} \longleftarrow S_{iter} \cup S_{iter-1}$.

6.) [Compute spectrum.] Compute $|S_{iter}(\tau)| = \left| \sum_{j=1}^{N} e^{(2\pi i s(j)/\tau)} \right|$.

7.) [Threshold.] Choose the largest peak. Label it as τ_{iter}

8.) If $|\tau_{iter} - \tau_{iter-1}| < \mathrm{Error}$. Declare $\widehat{\tau_i} = \tau_{iter}$. If not, $\mathtt{iter} \longleftarrow (\mathtt{iter} + 1)$. Go to **1.)**.

9.) Given τ_i, frequency notch $|S_{iter}(\tau)|$ for $\widehat{\tau_i}/m$, $m \in \mathbb{N}$. Let $i \longleftarrow i + 1$.

10.) [Compute spectrum.] Compute $|S_{iter}(\tau)| = \left| \sum_{j=1}^{N} e^{(2\pi i s(j)/\tau)} \right|$.

11.) [Threshold.] Choose the largest peak. Label it as τ_{i+1}. Algorithm terminates when there are no peaks. Else, go to **9.)**.

The EQUIMEA Algorithm – Two Periods

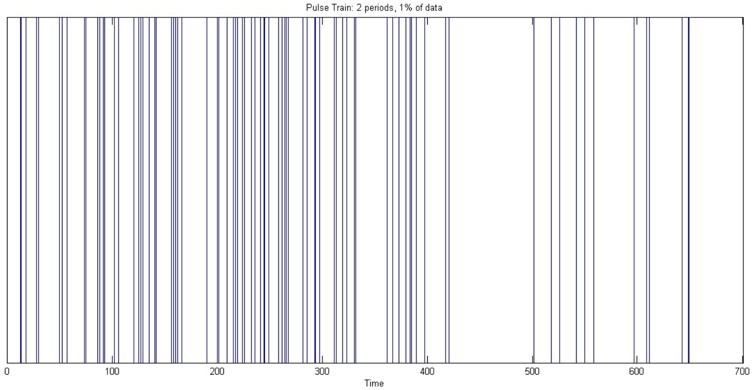


Figure: Two Periods – OriginalData

The EQUIMEA Algorithm – Two Periods, Cont'd

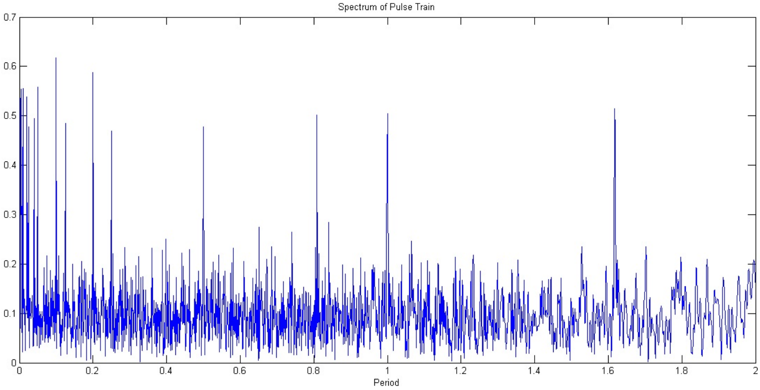


Figure: Spectrum of Two Period Data

The EQUIMEA Algorithm – Two Periods, Cont'd

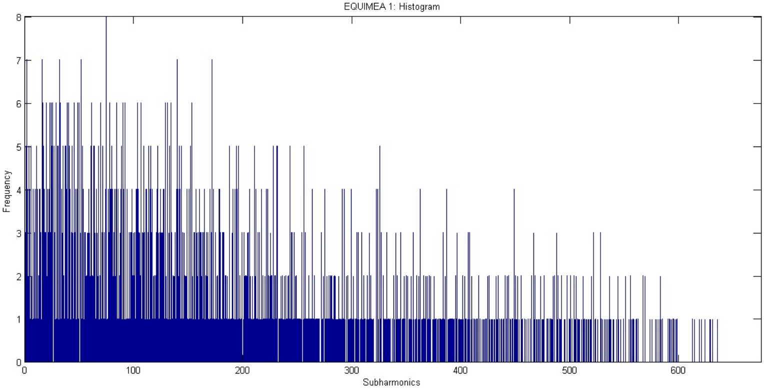


Figure: EQUIMEA – Two Periods – Iter1

The EQUIMEA Algorithm – Two Periods, Cont'd

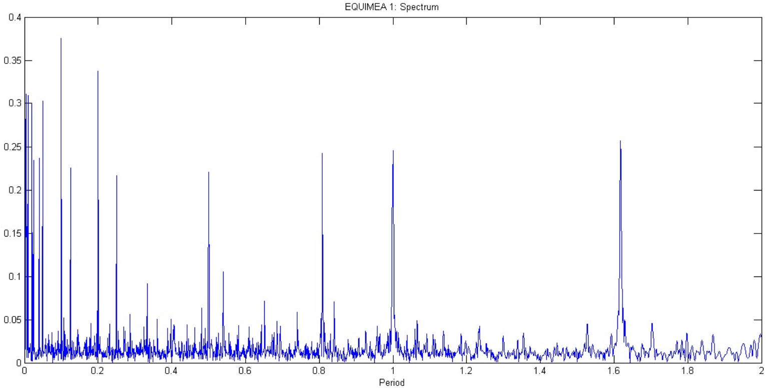


Figure: EQUIMEA – Two Periods – Iter1 – Spectrum

The EQUIMEA Algorithm – Two Periods, Cont'd

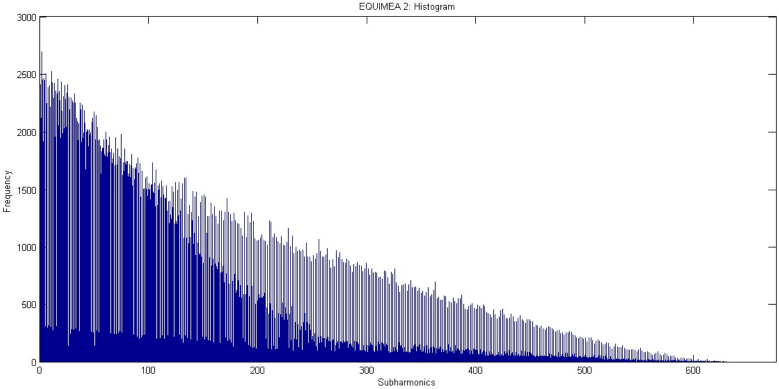


Figure: EQUIMEA – Two Periods – Iter2

The EQUIMEA Algorithm – Two Periods, Cont'd

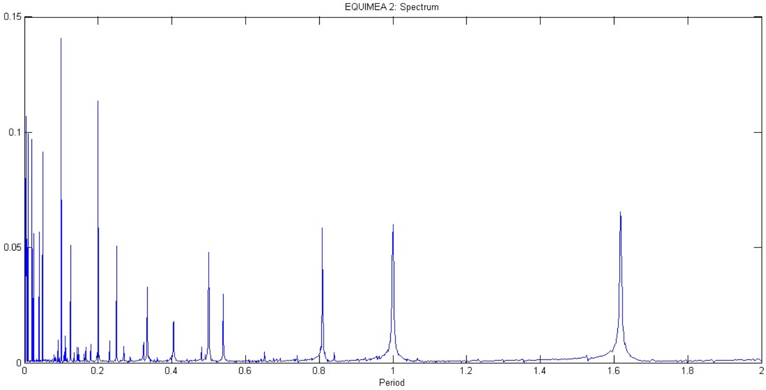


Figure: EQUIMEA – Two Periods – Iter2 – Spectrum

The EQUIMEA Algorithm – Three Periods

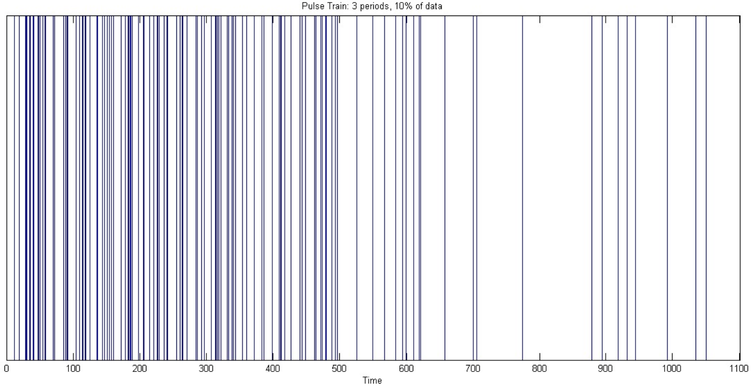


Figure: Three Periods – OriginalData

The EQUIMEA Algorithm – Three Periods, Cont'd

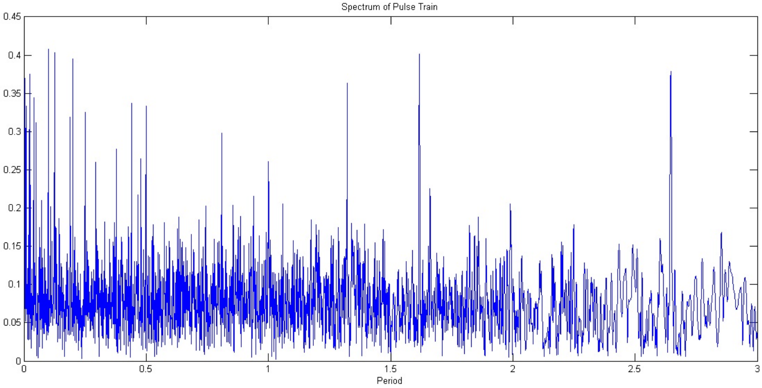


Figure: Spectrum of Three Period Data

The EQUIMEA Algorithm – Three Periods, Cont'd

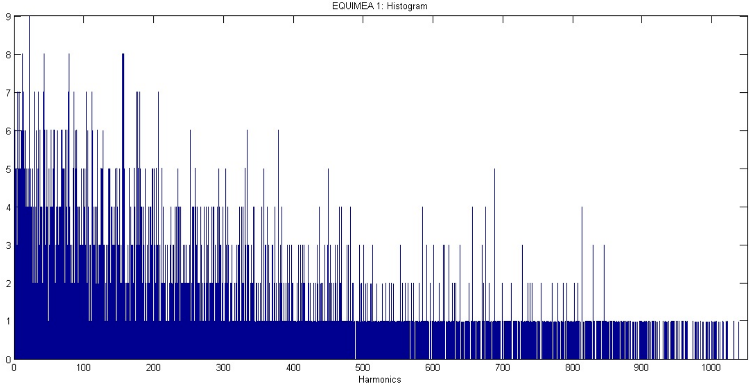


Figure: EQUIMEA – Three Periods – Iter1

The EQUIMEA Algorithm – Three Periods, Cont'd

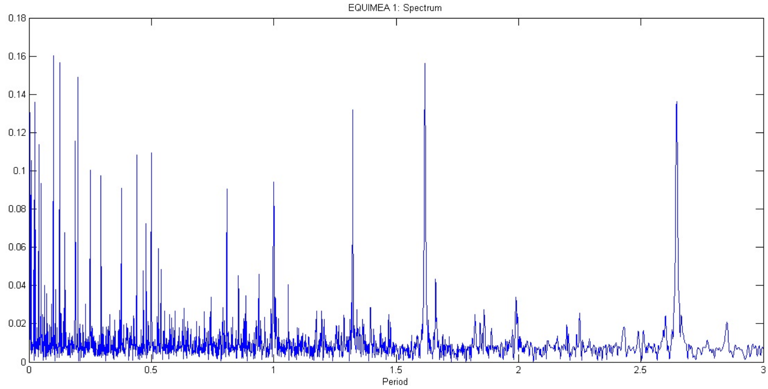


Figure: EQUIMEA – Three Periods – Iter1 – Spectrum

The EQUIMEA Algorithm – Three Periods, Cont'd

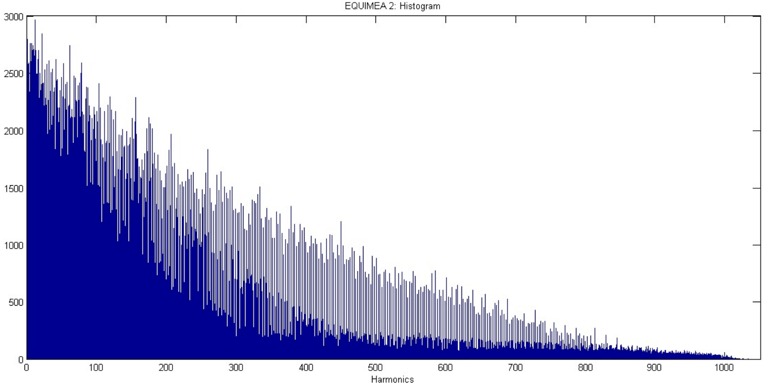


Figure: EQUIMEA – Three Periods – Iter2

The EQUIMEA Algorithm – Three Periods, Cont'd

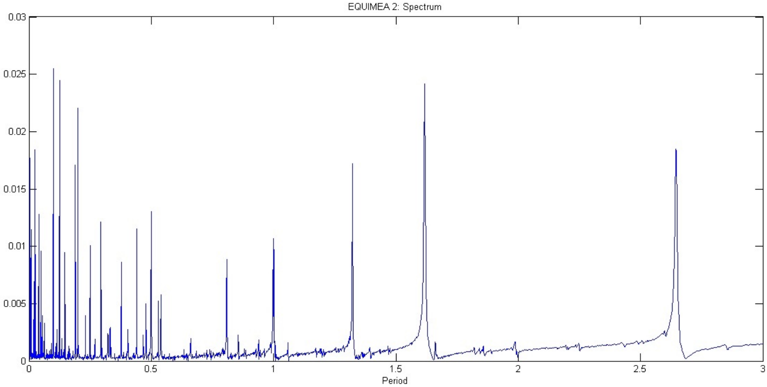


Figure: EQUIMEA – Three Periods – Iter2 – Spectrum

Prof. Kedem

- Let me close as I began, with sincere thanks to Prof. Kedem.
- Thank you for the gift of your wonderful lectures, which gave me the insight to attack problem solving via an interdisciplinary approach.

References

- S.D. Casey and B.M. Sadler, "Pi, the primes, periodicities and probability," *The American Mathematical Monthly*, Vol. 120, No. 7, pp. 594–608 (2013).
- S.D. Casey, "Sampling issues in Fourier analytic vs. number theoretic methods in parameter estimation," *31st Annual Asilomar Conference on Signals, Systems and Computers*, Vol. 1, pp, 453–457 (1998) (invited).
- S.D. Casey and B.M. Sadler, "Modifications of the Euclidean algorithm for isolating periodicities from a sparse set of noisy measurements," *IEEE Transactions on Signal Processing*, Vol. 44, No. 9, pp. 2260–2272 (1996) .
- D.E. Knuth, *The Art of Computer Programming, Volume 2: Seminumerical Algorithms (Second Edition)*, Addison-Wesley, Reading, Massachusetts (1981).
- K. Nishiguchi and M. Kobayashi, "Improved algorithm for estimating pulse repetition intervals," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 36, No. 2, pp. 407–421 (2000).
- B.M. Sadler and S.D. Casey, "PRI analysis from sparse data via a modified Euclidean algorithm," *29$^{\rm th}$ Annual Asilomar Conf. on Sig., Syst., and Computers* (invited) (1995).
- B.M. Sadler and S.D. Casey, "On pulse interval analysis with outliers and missing observations," *IEEE Transactions on Signal Processing*, Vol. 46, No. 11, pp. 2990–3003 (1998).
- B.M. Sadler and S.D. Casey, "Sinusoidal frequency estimation via sparse zero crossings," *Jour. Franklin Inst.*, Vol. 337, pp. 131–145 (2000).
- N.D. Sidiropoulos, A. Swami, and B.M. Sadler, "Quasi-ML period estimation from incomplete timing data," *IEEE Transactions on Signal Processing* **53** no. 2 (2005) 733–739.