# A MODIFIED EUCLIDEAN ALGORITHM FOR ISOLATING PERIODICITIES FROM A SPARSE SET OF NOISY MEASUREMENTS

*Stephen D. Casey*

Department of Mathematics and Statistics
The American University
Washington, DC 20016–8050

*Brian M. Sadler*

Army Research Laboratory
Adelphi, MD 20783

## ABSTRACT

A modified Euclidean algorithm is presented for determining the period from a sparse set of noisy measurements. The set may arise from measuring the occurrence time of noisy zero-crossings of a sinusoid with very many missing observations. The procedure is computationally simple, stable with respect to noise, and converges quickly. Its use is justified by a theorem that shows that, for a set of randomly chosen positive integers, the probability that they do not all share a common prime factor approaches one quickly as the cardinality of the set increases. Simulations are presented to demonstrate the proposed algorithm.

## 1. INTRODUCTION: THE PROBLEM

We are given the finite set of real numbers

$$S = \{s_j\}_{j=1}^{n}, \text{ with } s_j = k_j\tau + \epsilon_j, \qquad (1)$$

where $\tau$ is a fixed positive real number to be determined, the $k_j$'s are non-repeating positive integers, and the $\epsilon_j$'s are zero-mean indep. ident. dist. (iid) error terms. We assume that the $\epsilon_j$'s have a symmetric probability density function (pdf), and that $|\epsilon_j| < \frac{\tau}{2}$ for all $j$. Without loss of generality, we will also assume the set $S$ is ordered so that the $k_j$'s and thus the $s_j$'s are monotonically increasing (or decreasing). We may then interpret the $s_j$'s as occurrence times, with time gaps or jumps determined by the $k_j$'s. For example, letting $k_1 = 1$, $k_2 = 3$, $k_3 = 5, \ldots$, corresponds to missing every other occurrence of a periodic event with period $\tau$. In general, the solution is the unique maximum value of $\tau$ such that the $k_j$ are all integers. Note that for any solution $\tau$ that fits the form of $S$, the numbers $\frac{\tau}{2}, \frac{\tau}{3}, \ldots$ are also possible solutions.

The set (1) may occur in situations with unreliable measurements, such as fading communications channels or biomedical applications. A useful interpretation of our problem is to consider the set $S$ as containing measurements of the time of occurrence of noisy zero-crossings of a sinusoid, with (perhaps very many) missing observations. Given $x(t) = \cos[2\pi ft + \phi(t)]$ then zero-crossings are observed at odd multiples of $\tau = 1/4f$, with measurement error (phase jitter) introduced by $\phi(t)$. The $k_j$'s in (1) represent the occurrence of available measurements, and the $\epsilon_j$'s are associated with $\phi(t)$.

We solve for $\tau$ by using a modified Euclidean algorithm to find the greatest common divisor of the set $S$. The use of this algorithm is justified by a theorem that shows that, given a set of uniformly distributed positive integers distributed over an arbitrarily large lattice, the probability that they all do not share a common prime factor approaches one quickly as the cardinality of the set increases. In the noise-free case our algorithm is equivalent to the Euclidean algorithm and converges with very high probability given only 10 measurements, independent of the number of missing observations.

Our problem is similar to that of spectrum analysis using zero crossings. However, approaches based on zero-crossing counts (e.g., see Kedem [4]) are not applicable in our framework. Kay and Sudhaker proposed the use of the occurrence time of zero-crossings to obtain DFT coefficients [3]. This method requires addition of an auxiliary signal before detection of the zero-crossings and is therefore not applicable here. We also note that multiplicative (AM) missing observation models are not applicable due to their assumption of uniformly spaced amplitude samples (e.g., Parzen [7]).

## 2. A MODIFIED EUCLIDEAN ALGORITHM FOR FINDING $\tau$

Given the set $S$ as in (1), we develop a modified Euclidean algorithm to find $\tau$. The Euclidean algorithm is a division process for the integers $\mathbf{Z}$, which yields the greatest common divisor of two (or more) elements of $\mathbf{Z}$. We assume throughout the paper that $\{k_j\} \subset \mathbf{N}$, the set of positive integers (or natural numbers), and

0-7803-2431-5/95 $4.00 

that the $k_j$'s are sorted in descending order (unless otherwise noted). The symbol $\gcd(k_1, \ldots, k_n)$ 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 $k_j$. Note that this is not the pairwise gcd of the set $\{k_j\}$. If $\gcd(k_1, \ldots, k_n) = 1$, the set $\{k_j\}$ is called *mutually relatively prime*. If, however, $\gcd(k_i, k_j) = 1$ for all $i \neq j$, the set $\{k_j\}$ is called *pairwise relatively prime*. If a set is pairwise relatively prime, it is mutually relatively prime. However, the converse is not true (for example, consider the set $\{35, 21, 15\}$). The computation of the gcd of a set of more than two integers uses the fact that $\gcd(k_1, \ldots, k_n) = \gcd(k_1, \ldots, k_{n-2}, (\gcd(k_{n-1}, k_n)))$ (see Leveque [6]).

The standard Euclidean algorithm involves repeated division with remainder, terminating when the remainder is zero. In our problem, we are dealing with numbers that are essentially "noisy integers." 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. Thus, the standard algorithm is unstable under perturbation by noise. However, the algorithm may be modified so that the process of subtraction replaces division.

The modified algorithm is based on the following lemma, proven in Casey and Sadler [1]. We assume $\tau > 0$.

**Lemma 2.1**

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

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

This lemma allows a reformulation of the Euclidean algorithm, using subtraction rather than division. This idea works for the set $S$ as follows. First, sort the elements of $S$ in descending order, so that $s_1 \geq s_2 \geq \cdots \geq s_n$. Then, form a new set by subtracting adjacent pairs of these numbers, given by $s_j - s_{j+1}$. The result is a set of the same general form as $S$. Because of the $\epsilon_j$ perturbations we establish a threshold $\epsilon_0$ and, after the first sort and subtract, we declare all numbers in the interval $[0, \epsilon_0]$ to be zero and eliminate them from the set. Choice of $\epsilon_0$ is dictated by the distribution of the $\epsilon_j$'s, with $0 < \epsilon_0 < \frac{\tau}{2}$. We then adjoin the previous non-zero minimum to the set. The algorithm is continued by iterating this process of sorting, subtracting, and eliminating the elements in $[0, \epsilon_0]$, adjoining the previous non-zero minimum at each new iteration, and terminating when all elements are in $[0, \epsilon_0]$, i.e., equal to "zero." The maximal non-zero element from the previous iteration is equal to $\gcd(k_1, \ldots, k_n) \cdot \tau \pm$ error term.

The algorithm for finding $\tau$ given the set $S$ is summarized as follows. We assume for this and all other algorithms that the original data set is in descending order, i.e., $s_j \geq s_{j+1}$.

**Modified Euclidean Algorithm**

1. Save $s_m = \min(S)$ for adjoining in step 3.
2. Form the new set with elements $s_j - s_{j+1}$.
3. Adjoin $s_m$ from step 1.
4. Sort in descending order.
5. Eliminate elements in $[0, \epsilon_0]$ from end of the set.
6. Algorithm is done if left with empty set. Declare $\hat{\tau} = s_1$ from previous iteration. If not done, go to 1.

Note that given the set $S$ and an estimate $\hat{\tau}$, we can estimate $k_j$ using $\hat{k}_j = \text{round}\,(s_j/\hat{\tau})$, where round$(\cdot)$ denotes rounding to the nearest integer.

The success of the algorithm depends on the fact that $\gcd(k_1, \ldots, k_n) \longrightarrow 1$ with probability 1 as $n \longrightarrow \infty$ (see Casey and Sadler [1]). Moreover, in [1], we have shown that this convergence is very fast, implying that the proposed modified Euclidean algorithm yields $\tau$ and the sequence $\{k_j\}_{j=1}^n$ for small ($n \approx 10$) to moderate ($n \approx 100$) values of $n$, depending on the distribution of the $\epsilon_j$'s. In order to state our results, we need to establish some background on Riemann's Zeta function, which is defined on the complex half space $\{z \in \mathbf{C} : \Re(z) > 1\}$ by $\zeta(z) = \sum_{n=1}^{\infty} n^{-z}$. Euler demonstrated the connection of $\zeta$ with number theory by showing, in 1736, that

$$\zeta(z) = \prod_{j=1}^{\infty} \frac{1}{1 - (p_j)^{-z}}\,, \ \Re(z) > 1\,, \qquad (2)$$

where $\mathbf{P} = \{p_1, p_2, p_3, \ldots\} = \{2, 3, 5, \ldots\}$ is the set of all prime numbers.

**Theorem 2.1** *Given $n$ ($n \geq 2$) randomly chosen positive integers $\{k_1, \ldots, k_n\}$,*

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

The result is classical for the case $n = 2$, and was proven by Dirichlet in 1849 (see Knuth [5], pp. 324, 337, 595, and Schroeder [9], pp. 48–50). Our proof follows directly from the following theorem. We let card$\{\cdot\}$ denote set cardinality, and let $\{1, \ldots, \ell\}^n$ denote the sublattice of positive integers in $\mathbf{R}^n$ with coordinates $c$ such that $1 \leq c \leq \ell$.

**Theorem 2.2** *Let*

$$N_n(\ell) = \text{card}\{(k_1, \ldots, k_n) \in \{1, \ldots, \ell\}^n : \gcd(k_1, \ldots, k_n) = 1\}\,,$$

*For $n \geq 2$, we have that*

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

The values for $\zeta(2k)$ can be computed explicitly, using Cauchy Residue Theory (see [2]). The values $\zeta(2k+1)$ can be estimated numerically. It can be shown that that $[\zeta(n)]^{-1} \longrightarrow 1$ quickly as $n$ increases (see [1]).

## 3. FURTHER MODIFICATIONS

In this section we consider the effect of the noise perturbations on the modified Euclidean algorithm, and show how the noise effects can be reduced. In the modified Euclidean algorithm we have replaced division with repeated subtraction in order to gain stability with respect to noise. Error analysis of this approach is complicated by the fact that the algorithm is iterative and involves order statistics.

Suppose the pdf of the $\epsilon_j$'s is given by $f_\epsilon(\epsilon)$, and consider the set of differences obtained in the first iteration, given by

$$y_j = s_j - s_{j+1} = (k_j - k_{j+1})\tau + (\epsilon_j - \epsilon_{j+1}). \qquad (3)$$

Invoking the zero-mean iid assumption on the $\epsilon_j$'s the pdf of $(\epsilon_j - \epsilon_{j+1})$ is given by the convolution $f_\epsilon(\epsilon) * f_\epsilon(\epsilon)$. So, for example, if $f_\epsilon(\epsilon) \sim \mathcal{U}[-\frac{\Delta}{2}, \frac{\Delta}{2}]$ ($\epsilon$ is uniformly distributed with parameter $\Delta$) then $f_{y_j}(y) = \text{tri}[y - (k_j - k_{j+1})\tau]$, the triangle function centered at $(k_j - k_{j+1})\tau$. Two points can now be made. First, after the first iteration the differencing operation has removed the independence of the error terms. Second, the ordering operation makes the nature of the dependence in subsequent iterations difficult to determine. Analysis of order statistics very often rests on an iid assumption, e.g., see [8].

In general, beyond the first iteration the pdf of the subsequent error terms becomes asymmetric, even when starting with iid $\epsilon_j$'s with symmetric pdf $f_\epsilon(\epsilon)$. This occurs due to the reordering before differencing at each iteration. The result is that using the modified Euclidean algorithm without averaging leads to negatively biased estimates of $\tau$ after the first iteration due to the skewness of the pdf of the errors. However, after the first iteration the error is still symmetrically distributed. As we see from (3), the $y_j$'s will be clustered about integer multiples of $\tau$, given by $(k_j - k_{j+1})\tau$ for each $j$. The data has concentrated into "steps" with symmetrically distributed error about each step. This suggests averaging the data around each step to remove the noise effects as much as possible. A threshold $\epsilon_0$ is chosen to partition the steps. In practice the threshold $\epsilon_0$ may be chosen adaptively based on the spread of the data about $\tau$. This leads to the following algorithm.

**Modified Euclidean Algorithm (w/ Averaging)**

1. Save $s_m = \min(S)$ for adjoining in step 3.
2. Form the new set with elements $s_j - s_{j+1}$.
3. Adjoin $s_m$ from step 1.
4. Sort in descending order.
5. Average the data in each interval $[k\tau - \epsilon_0, k\tau + \epsilon_0]$, for $k = 1, 2, \ldots$.
6. Eliminate elements in $[0, \epsilon_0]$ from end of the set.
7. Algorithm is done if left with empty set. Declare $\hat{\tau} = s_1$ from previous iteration. If not done, go to 1.

This approach produces significant data reduction at each iteration and therefore greatly increases the speed of convergence. The averaging algorithm is also effective for extremely sparse data sets, as we demonstrate by example in the next section.

If the data is such that, after the first iteration, there is a relatively large cluster around the first step, then we can readily estimate $\tau$ by finding the first step, averaging only over these data points in the interval $[\tau - \epsilon_0, \tau + \epsilon_0]$, and declaring this to be $\hat{\tau}$. Under our assumptions this is an unbiased estimate. Accurate estimation of $\tau$ from a single iteration assumes that $n$ is large enough, and the spread small enough, to yield sufficient data in the neighborhood of $\tau$. This is a function of the distribution of the $k_j$'s. For example, suppose the stepsize between observations is uniformly distributed on the discrete interval $[1, M]$ ($M$ integer). Then, for large $n$, after the first iteration we expect the data to cluster into $M$ bins with $n/M$ points in each bin.

**Modified Euclidean Algorithm (One Iteration)**

1. Given the set $S$, form the new set with elements $s_j - s_{j+1}$.
2. Sort the new set in descending order.
3. Eliminate elements in $[0, \epsilon_0]$ from end of the set.
4. The estimate $\hat{\tau}$ is the average over the data in $[\tau - \epsilon_0, \tau + \epsilon_0]$.

## 4. SIMULATION RESULTS

In this section we present simulation results for the proposed algorithms. All estimates and their standard deviations are based on averaging Estimates of $\tau$ are labeled $\hat{\tau}$ with $std(\hat{\tau})$ the experimental standard deviation and $n$ the number of data points. Without loss of generality, we take $\tau = 1$ in all experiments, and set the threshold value of $\epsilon_0 = 0.35\tau = 0.35$. The value of *iter* is the average number of iterations required for convergence, and *%miss* denotes the average number of missing observations expressed as a percentage of the total possible number of observations. Noise free simu-

Table 1: Example 1 results, repeating example 2 with missing observations modeled by a Bernoulli process.

| $n$ | $\lambda$ | $\Delta$ | *%miss* | *iter* | $\hat{\tau}$ | $std(\hat{\tau})$ |
|---|---|---|---|---|---|---|
| 10 | 0.80 | 0.001 | 76.64 | 5.98 | 0.9988 | 0.0010 |
| 10 | 0.80 | 0.01 | 76.81 | 6.46 | 0.9885 | 0.0054 |
| 10 | 0.80 | 0.02 | 76.54 | 5.66 | 0.9763 | 0.0109 |

lations confirmed the expected behavior of the modified Euclidean algorithm, with $n = 10$ observations generally sufficient for recovery of $\tau$ with an arbitrarily large percentage of observations missing.

*Example 1: Bernoulli model for missing observations.* We implemented the modified Euclidean algorithm of section 2, modeling the missing observations using a Bernoulli process, given by

$$\begin{aligned} P(\text{missing observation}) &= \lambda \\ P(\text{observation occurring}) &= 1-\lambda. \end{aligned} \qquad (4)$$

The probability of an observation occurring is assumed independent between observations. The gaps in the data are modeled as uniformly distributed. The $\epsilon_j$'s have uniform distribution, given by $f_\epsilon(\epsilon) \sim \mathcal{U}[-\frac{\Delta}{2}, \frac{\Delta}{2}]$. Results are shown in Table 1. The value $\lambda = 0.8$ was used, corresponding to $\approx$ 77% observations missing, and the noise parameter $\Delta$ was varied.

*Example 2: Single iteration algorithm.* In this example we implement the single iteration algorithm of section 4, with results shown in Table 2. yielding $\approx$ 80% missing observations. Here *#data* is the average number of data points occurring in $[\tau - \epsilon_0, \tau + \epsilon_0]$ after the single iteration, and *std* is the experimental standard deviation. Note the quality of the estimates of $\tau$ despite large error terms.

*Example 3: Averaging algorithm with* 98% *missing observations.* As a final example we implement the averaging algorithm of section 4 with very sparse data, with results shown in Table 3. This example demonstrates the effectiveness of averaging (results should be compared with example 2 where the lack of averaging lead to significant bias in the estimate of $\tau$). The jumps in the $k_j$'s were uniformly distributed on $[1, M]$ with $M = 100$, corresponding to $\approx$ 98% missing observations. A threshold of $\epsilon_0 = 0.5$ was used to determine the occurrence of a step. Note the rapid convergence in 3 to 4 iterations, with accurate estimation of $\tau$ from 100 data points with $\pm$10% phase jitter.

Table 2: Example 2 results, single iteration algorithm with Bernoulli missing observation model ($\lambda = 0.8$, yielding $\approx$ 80% missing observations).

| $n$ | $\Delta$ | *#data* | *std* | $\hat{\tau}$ | $std(\hat{\tau})$ |
|---|---|---|---|---|---|
| 100 | $10^{-3}$ | 18.6 | 3.5 | 1.0000 | 0.00008 |
| 100 | $10^{-2}$ | 18.3 | 4.0 | 0.9999 | 0.0008 |
| 100 | $10^{-1}$ | 18.8 | 4.1 | 1.0000 | 0.0081 |
| 100 | $2 \times 10^{-1}$ | 19.1 | 4.3 | 1.0031 | 0.0157 |
| 200 | $2 \times 10^{-1}$ | 38.5 | 5.6 | 0.9978 | 0.0134 |
| 200 | $3 \times 10^{-1}$ | 37.8 | 6.6 | 0.9956 | 0.0197 |

Table 3: Example 3 results, multiple iterations with sublevel averaging, with $\approx$ 98% missing observations.

| $n$ | $\Delta$ | *iter* | $std(iter)$ | $\hat{\tau}$ | $std(\hat{\tau})$ |
|---|---|---|---|---|---|
| 100 | $10^{-3}$ | 3.4 | 0.48 | 1.0000 | 0.00008 |
| 100 | $10^{-2}$ | 3.3 | 0.45 | 1.0001 | 0.00064 |
| 100 | $10^{-1}$ | 3.3 | 0.45 | 1.0005 | 0.0060 |
| 100 | $2 \times 10^{-1}$ | 3.3 | 0.46 | 0.9955 | 0.037 |
| 200 | $2 \times 10^{-1}$ | 3.1 | 0.30 | 0.9993 | 0.0072 |

## 5. REFERENCES


[1] Casey, S. D., and Sadler, B. M., "A modified Euclidean algorithm for isolating periodicities from a sparse set of noisy measurements," *IEEE Trans. SP*, submitted 1994.

[2] Ireland, K., and Rosen, M., *A Classical Introduction to Modern Number Theory*, Springer-Verlag, New York (1982).

[3] Kay, S. M., and Sudhaker, R., "A Zero Crossing-Based Spectrum Analyzer," *IEEE Trans. ASSP*, **ASSP-34**, 1, 96–104, 1986.

[4] Kedem, B., *Time Series Analysis by Higher Order Crossings*, IEEE Press, New York (1994).

[5] Knuth, D., *The Art of Computer Programming - Vol. 2 (Second Edition)*, Addison-Wesley, Reading, Massachusetts (1981).

[6] Leveque, W. J., *Topics in Number Theory, Vols. 1 and 2*, Addison-Wesley, Reading, Massachusetts (1956).

[7] Parzen, E., "On Spectral Analysis with Missing Observations and Amplitude Modulation," *Sankhya*, Ser. A, **25**, 383–392, 1963.

[8] Sarhan, A. E., and Greenberg, B. G., eds., *Contributions to Order Statistics*, John Wiley, New York (1962).

[9] Schroeder, M. R., *Number Theory in Science and Communication (Second Edition)*, Springer-Verlag, Berlin (1986).