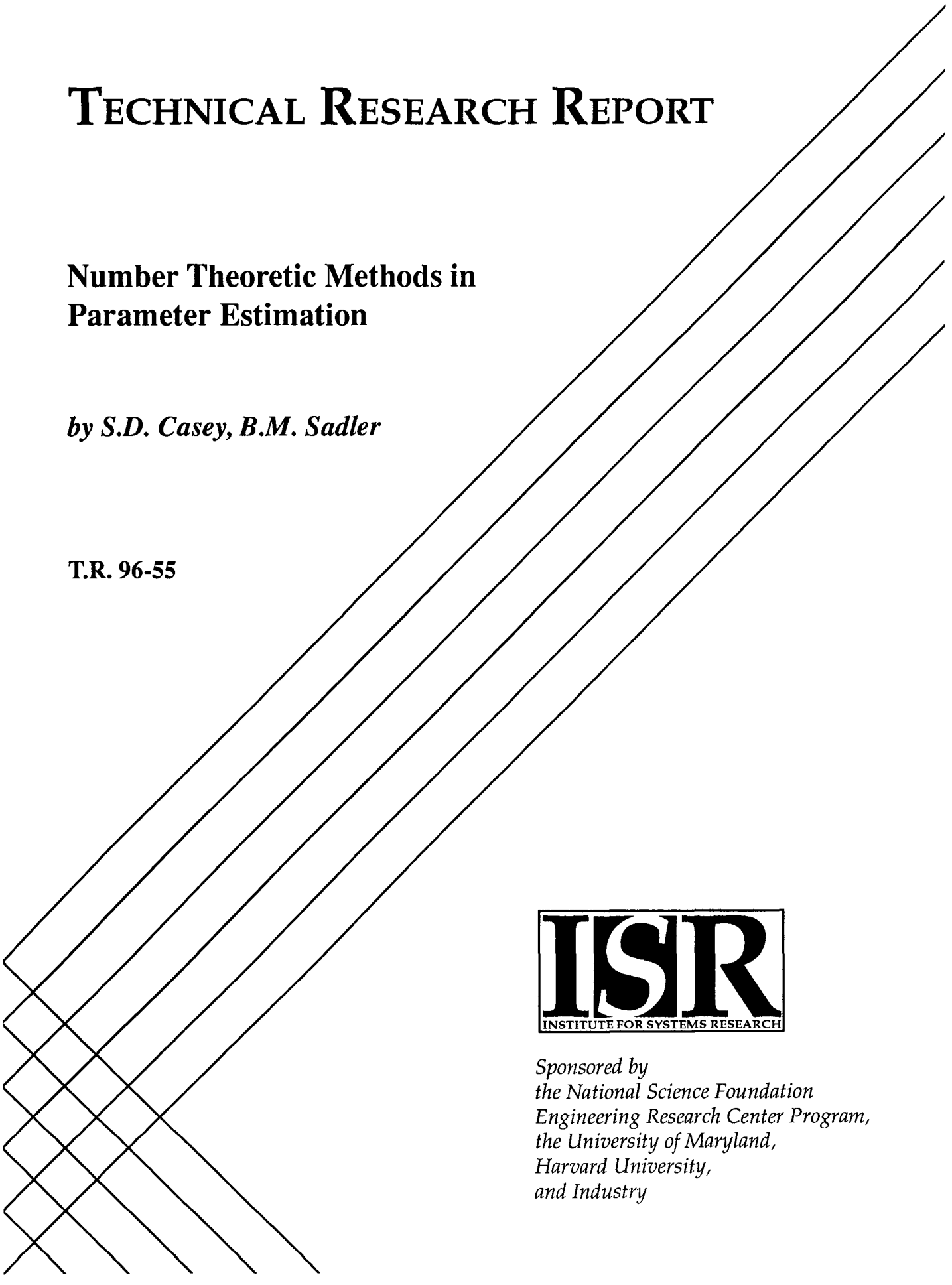
TECHNICAL RESEARCH REPORT

Number Theoretic Methods in Parameter Estimation

by S.D. Casey, B.M. Sadler

T.R. 96-55



Sponsored by
the National Science Foundation
Engineering Research Center Program,
the University of Maryland,
Harvard University,
and Industry

Number Theoretic Methods in Parameter Estimation

Stephen D. Casey[1][2]
Institute for Systems Research
University of Maryland
College Park, MD 20742-3311
e-mail: scasey@isr.umd.edu

Brian M. Sadler
Army Research Laboratory
2800 Powder Mill Road
Adelphi, MD 20783
e-mail: sadler@arl.mil

[1]Research partially supported by Army Research Laboratory contract # DAAL01-96-P-1311 and Air Force Office of Scientific Research Grant F49620-94-1-0196.

[2]On sabbatical leave from the Department of Mathematics and Statistics, The American University, 4400 Massachusetts Ave., N.W., Washington, DC 20016-8050 *(e-mail: scasey@american.edu)*.

Abstract

Number theory has proven to be an effective tool in harmonic analysis, used to extend existing theories (e.g., sampling theory, fast transform computations) and develop new approaches to problems (e.g., interpolation). Number theoretic methods have also been successfully applied to the analysis of periodic point processes, leading to computationally straightforward algorithms for several parameter estimation problems.

We first present modifications of the Euclidean algorithm which determine the period from a sparse set of noisy measurements. The elements of the set are the noisy occurrence times of a periodic event with (perhaps very many) missing measurements. The approach is justified by a theorem which 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. A robust version is developed that is stable despite the presence of arbitrary outliers. We then use these algorithms in the analysis of periodic pulse trains, getting an estimate of the underlying period. This estimate, while not maximum likelihood, is used as initialization in a three-step algorithm that achieves the Cramer-Rao bound for moderate noise levels, as shown by comparing Monte Carlo results with the Cramer-Rao bounds. We close by discussing our work on deinterleaving. Here we discuss a variation on Weyl's Equidistribution Theorem, which works for noisy measurements. We then use periodogram-like operators in a multistep procedure to isolate fundamental periods.

1 Introduction

Problems in harmonic analysis and synthesis are intertwined with their applications in signal and image processing. Some recent advances in this analysis have used number theory to extend existing theories (e.g., sampling theory, fast transform computations) and develop new approaches to problems (e.g., interpolation). Number theoretic methods have also been successfully applied to the analysis of periodic point processes. The purpose of this note is to discuss several recent developments in which number theory has been used to develop algorithms for several classes of parameter estimation problems. These results will be presented at the 1996 IEEE SSAP Workshop [4].

We first present modifications of the Euclidean algorithm which determine the period from a sparse set of noisy measurements [2, 3]. The elements of the set are the noisy occurrence times of a periodic event with (perhaps very many) missing measurements. The proposed algorithms are computationally straightforward and converge quickly. A robust version is developed that is stable despite the presence of arbitrary outliers. The Euclidean algorithm approach is justified by a theorem which 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. The theorem is in essence a probabilistic interpretation of the Riemann Zeta Function. In the noise-free case this implies convergence with only ten data samples, independent of the percentage of missing measurements. In the case of noisy data simulation results show, for example, good estimation of the period from one hundred data samples with fifty percent of the measurements missing and twenty five percent of the data samples being arbitrary outliers.

We then use these algorithms in the analysis of periodic pulse trains, getting an estimate of the underlying period [4, 18]. This estimate, while not maximum likelihood, is used as initialization in a three-step algorithm that achieves the Cramer-Rao bound for moderate noise levels, as shown by comparing Monte Carlo results with the Cramer-Rao bounds. An approach using multiple independent data records is also developed that overcomes high levels of contamination.

We close by discussing our work on the deinterleaving of multiple periodic pulse trains [5]. Here we give a variation on Weyl's Equidistribution Theorem, which shows that noisy phase-wrapped data is equidistributed on $[0, 1)$ almost surely. We then use periodogram-like operators in a multistep procedure to isolate fundamental periods.

2 Modified Euclidean Algorithms

Our first problem begins with a set of noisy occurrence times of a periodic event with (perhaps very many) missing measurements. We have developed modifications of the Euclidean algorithm for determining the period from this set [2, 3]. This problem arises in radar pulse repetition interval (PRI) analysis, in bit synchronization in communications, in biomedical applications, and other scenarios. We assume our data is a finite set of real numbers

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

where τ (the period) is a fixed positive real number, the k_j's are non-repeating positive integers, ϕ (the phase) is a real random variable uniformly distributed over the interval $[0, \tau)$, and the η_j's 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 $|\eta_j| < \frac{\tau}{2}$ for all j. We develop an algorithm for isolating the period of the process from this set, which we shall assume is (perhaps very) sparse. In the noise-free case our basic algorithm, given below, is equivalent to the Euclidean algorithm and converges with very high probability given only $n = 10$ data samples, independent of the number of missing measurements. We assume that the original data set is in descending order, i.e., $s_j \geq s_{j+1}$. For this first algorithm, we assume an *a priori* threshold η_0, where $0 < \eta_0 < \tau$. Variations on this algorithm include a data-adaptive threshold (see Section 2.3).

Modified Euclidean Algorithm

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

Initialize : Sort the elements of S in descending order.

1.) After the first iteration, append zero.

2.) Form the new set with elements $s_j - s_{j+1}$.

3.) Sort in descending order.

4.) Eliminate elements in $[0, \eta_0]$ from end of the set.

5.) Algorithm is done if left with a single element. Declare $\widehat{\tau} = s_1$. If not done, go to **(1.)**.

Noise-free simulation examples demonstrate successful estimation of τ for $n = 10$ with 99.99% of the possible measurements missing. In fact, with only 10 data samples, it is possible to have the percentage of missing measurements arbitrarily close to 100%. There is, of course, a cost, in that the number of iterations the algorithm needs to converge increases with the percentage of missing measurements. In the presence of noise and false data (outliers), there is a tradeoff between the number of data samples, the amount of noise, and the percentage of outliers. The algorithm will perform well given low noise for $n = 10$, but will degrade as noise is increased. However, given more data, it is possible to reduce noise effects and speed up convergence by binning the data, and averaging across bins. Binning can be effectively implemented by using an adaptive threshhold with a gradient operator, allowing convergence in a single iteration in many cases. Simulation results show, for example, good estimation of the period from one hundred data samples with fifty percent of the measurements missing and twenty five percent of the data samples being arbitrary outliers [2, 3].

2.1 Simulation Results

We assume that $\tau = 1$ for all simulations, which makes it easier to evaluate our results. All estimates and their standard deviations are based on averaging over 100 Monte-Carlo runs. The number of data points is given by n. Estimates of τ are labeled $\hat{\tau}$, and $std(\hat{\tau})$ is the experimental standard deviation. The threshold value of $\eta_0 = 0.35\tau = 0.35$ was used throughout. (The value η_0 was set data-adaptively in later experiments.) 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.

1.) *Noise-free estimation.*
In this example we examine the effects of changing n and the percentage of missing observations on the modified Euclidean algorithm of Section 2. The data are noise-free, i.e., $\eta_j = 0$ for all j. In this case the algorithm converges to the exact value of $\tau = 1$ with standard deviation equal to zero, or in some cases for small n to some multiple of τ. The jumps in the k_j's were modeled as uniformly distributed on the (discrete) interval $[1, M]$. Results are shown in Table 1 where $\%miss$ denotes the experimentally determined average percentage of missing observations, and *iter* is the average number of iterations required to converge.

The top half of Table 1 illustrates the effect of changing M, and therefore changing the percentage of missing observations. Given insufficient data the algorithm will converge to a multiple of τ. Columns labeled τ, 2τ, etc., indicate the percentage of runs that converged to these values. The algorithm is able to choose τ correctly based on $n = 10$ data samples, even with 99.998% of the possible observations missing. Convergence in the noise-free case depends on n but is independent of M, as implied by the analysis of Section 2.2.

The bottom half of Table 1 illustrates the effect of changing n for M fixed. Reliable results are achieved for $n \geq 10$.

Table 1: Results from Simulation 1, noise-free estimation of τ with the modified Euclidean algorithm.

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	97.81	14.2	97	3	0	0
8	10^2	97.96	10.2	98	1	1	0
10	10^2	97.95	10.2	99	1	0	0
12	10^2	97.95	8.6	100	0	0	0
14	10^2	97.97	7.4	100	0	0	0

2.) *Uniformly distributed noise.*
We repeat Simulation 1 with the addition of noise. The gaps in the data are modeled as uniformly distributed as in Simulation 1. The η_j's have uniform distribution, given by $f_\eta(\eta) \sim \mathcal{U}[-\frac{\Delta}{2}, \frac{\Delta}{2}]$. Increasing M generally requires more data to maintain the same accuracy in $\widehat{\tau}$, and results in larger $std(\widehat{\tau})$. The bottom half of Table 2 shows the effect of increasing noise with n and M fixed.

Table 2: Simulation 2 results, estimation of τ from noisy measurements using the algorithm.

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

2.2 Theoretical Basis

Our algorithm is based on several theoretical results, which we now present. We also present short proofs and a sketch of a long proof, as these are of independent interest.

The Euclidean algorithm is a division process for the set of integers $\mathbf{Z}$. The algorithm is based on the property that, given two positive integers a and b, $a > b$, there exists 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, and denote this by $b|a$. This property of the set of integers, combined with the fact that if $a, b \in \mathbf{Z} \setminus \{0\}$ then $a \cdot b \neq 0$ ($\mathbf{Z}$ has no zero divisors), make $\mathbf{Z}$ a unique factorization domain. Thus in $\mathbf{Z}$ every non-zero element may be written as the product of powers of irreducible integers, or primes. The Euclidean algorithm also holds in more general algebraic structures called Euclidean domains.

The Euclidean algorithm yields the greatest common divisor of two (or more) elements of $\mathbf{Z}$. The *greatest common divisor* of two integers a and b, denoted by $\gcd(a, b)$, is the product of the powers of all prime factors p that divide both a and b. We may represent the algorithm applied to a, b, $a > b$, as follows:

$$\begin{aligned} a &= b \cdot q_1 + r_1 &&: \quad 0 < r_1 < b \\ b &= r_1 \cdot q_2 + r_2 &&: \quad 0 < r_2 < r_1 \\ &\vdots &&\quad \vdots \\ r_{k-2} &= r_{k-1} \cdot q_k + r_k &&: \quad 0 < r_k < r_{k-1} \\ r_{k-1} &= r_k \cdot q_k \,. \end{aligned}$$

The procedure terminates when $r_{k+1} = 0$. This gives $\gcd(a, b) = r_k$.

This procedure can be extended to work on S. 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 following proposition. There is also a natural extension of the gcd to multiples of a fixed $\tau > 0$.

Proposition 2.1

$$(i.) \quad \gcd(k_1\tau, \ldots, k_n\tau) = \tau \gcd(k_1, \ldots, k_n) \,, \tag{2}$$

$$(ii.) \quad \gcd(k_1, \ldots, k_n) = \gcd(k_1, \ldots, k_{n-2}, (\gcd(k_{n-1}, k_n))) \,. \tag{3}$$

Proof : See Leveque [14], page 16. □

The standard Euclidean algorithm, as shown above, involves repeated division. 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 iterations in the procedure may involve dividing by such small values, which would result in

arbitrarily large numbers. Thus, the standard algorithm is unstable under perturbation by noise. However, the algorithm may be changed so that the process of subtraction replaces division by making use of the following proposition. So that $(k_j - k_{j+1}) \in \mathbf{N}$, we assume that the k_j's are sorted in descending order.

Proposition 2.2

$$\gcd(k_1, \ldots, k_n) = \gcd((k_1 - k_2), (k_2 - k_3), \ldots, (k_{n-1} - k_n), k_n)\,. \tag{4}$$

Proof : Let α be a positive integer such that $\alpha | k_j$ for $j = 1, \ldots, n$. Then, $\alpha | (k_j - k_{j+1})$ for $j = 1, \ldots, n-1$, and $\alpha | k_n$.

Conversely, assume β is a positive integer such that $\beta | (k_j - k_{j+1})$ for $j = 1, \ldots, n-1$, and $\beta | k_n$. Therefore, there exists positive integers c and d such that $c\beta = k_n$ and $d\beta = (k_{n-1} - k_n)$. Thus, $d\beta + k_n = (d+c)\beta = k_{n-1}$, and so $\beta | k_{n-1}$. By complete induction, $\beta | k_j$ for $j = 1, \ldots, n$.

Therefore, since the sets $\{k_j\}$ and $\{(k_j - k_{j+1})\} \cup \{k_n\}$ have the same divisors, their gcd's are equal. □

We will also need the following related result. The proof is very similar.

Proposition 2.3

$$\gcd((k_1 - k_2), (k_2 - k_3), \ldots, (k_{n-1} - k_n)) = \gcd((k_1 - k_n), (k_2 - k_n), \ldots, (k_{n-1} - k_n))\,. \tag{5}$$

Proof : See [2]. □

We are now ready to give the justification of our modified algorithm. We assume the s_j's are sorted in *descending* order, i.e., $s_1 \geq s_2 \geq \cdots \geq s_n$. This allows a more straightforward visualization of our algorithm. We form a new set by subtracting adjacent pairs of these numbers, given by $s_j - s_{j+1}$. After this first operation, the phase information has been subtracted out, and the resulting set has the simpler form

$$S' = \{s'_j\}_{j=1}^{n-1}\,, \text{ with } s'_j = K_j \tau + \eta'_j\,,$$

where $K_j = k_j - k_{j+1}$ and $\eta'_j = \eta_j - \eta_{j+1}$. In subsequent iterations of the algorithm, the data will maintain this same general form. Because of the η_j perturbations we establish a threshold η_0 and, after the subtraction of adjacent pairs, we declare all numbers in the interval $[0, \eta_0]$ to be zero and eliminate them from the set. Choice of η_0 is dictated by the distribution of the η_j's, with $0 < \eta_0 < \frac{\tau}{2}$. We then append zero, sort, subtract adjacent pairs, and threshold. By appending zero, we adjoin the previous non-zero minimum to the set. The algorithm is continued by iterating this process of appending zero, sorting, subtracting, and eliminating the elements in $[0, \eta_0]$. It terminates when all but one of the elements are in $[0, \eta_0]$, i.e.,"equal to zero." By the propositions above, this element is equal to $\gcd(K_1, \ldots, K_{n-1}) \cdot \tau \pm$ error term.

We will see in the following discussion that $\gcd(K_1, \ldots, K_{n-1}) \longrightarrow 1$ with probability 1 as $n \longrightarrow \infty$. Moreover, we will see that this convergence is very fast, which shows that the

proposed modified Euclidean algorithm yields τ with a high probability for small ($n \approx 10$) to moderate ($n \approx 100$) values of n, depending on the distribution of the η_j's and the percentage of outliers. Let $\mathbf{P} = \{p_1, p_2, p_3, \ldots\} = \{2, 3, 5, \ldots\}$ be the set of all prime numbers. Let

$$\zeta(z) = \sum_{n=1}^{\infty} n^{-z}\,,\ \Re(z) > 1 \tag{6}$$

denote the Riemann zeta function. In 1736, Euler demonstrated the connection of the zeta function with number theory by proving that

$$\zeta(z) = \sum_{n=1}^{\infty} n^{-z} = \prod_{j=1}^{\infty} \frac{1}{1-(p_j)^{-z}}\,,\ \Re(z) > 1\,. \tag{7}$$

We show the following.

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

$$P\{\gcd(k_1, \ldots, k_n) = 1\} = [\zeta(n)]^{-1}\,. \tag{8}$$

Theorem 2.1 follows directly from the following. We let $\{1, \ldots, \ell\}^n$ denote the sublattice of $\mathbf{N}^n$ with coordinates c such that $1 \leq c \leq \ell$.

Theorem 2.2 *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

$$\lim_{\ell \to \infty} \frac{N_n(\ell)}{\ell^n} = [\zeta(n)]^{-1}\,. \tag{9}$$

We eliminate the phase information in the data by the subtraction of adjacent elements in the set. But then, rather than working with $\{k_1, \ldots, k_n\}$, we are working with $\{(k_1 - k_2), \ldots, (k_{n-1} - k_n)\}$.

Corollary 2.1 *Let $\{k_1, \ldots, k_n\}$ be n ($n \geq 3$) randomly chosen positive integers, with $k_j > k_{j+1}$, and let $K_j = k_j - k_{j+1}$ for $j = 1, \ldots, n-1$. Then*

$$P\{\gcd(K_1, \ldots, K_{n-1}) = 1\} = [\zeta(n-1)]^{-1}\,.$$

Sketch of Proof of Theorem 2.2 : Let $\lfloor x \rfloor = \max_{k \leq x}\{k : k \in \mathbf{Z}\}$. *Claim :*

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

Proof of Claim : 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 $\lfloor \frac{\ell}{p_i} \rfloor$. (Thus, we can have $p_i > \ell$, because $\lfloor \frac{\ell}{p_i} \rfloor = 0$.)

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 (independence)

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

If $p_i \cdot p_j | k$, 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 and p_j both divide every integer in the n-tuple is

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

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).

Inductively, we can see that the number of n-tuples $(k_1, \ldots, k_n)$ contained in the lattice $\{1, \ldots, \ell\}^n$ such that $p_i, p_j, p_k, \ldots$ all divide every integer in the n-tuple is given by the inclusion-exclusion principle as

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

But this counts the complement of $N_n(\ell)$ in the lattice $\{1, \ldots, \ell\}^n$. Therefore,

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

Now,

$$\begin{aligned}
& \frac{1}{\ell^n} \sum_{p_i < p_j < \ldots < p_k} \left(\lfloor \frac{\ell}{p_i \cdot p_j \cdot \ldots \cdot p_k} \rfloor\right)^n \\
\leq\ & \frac{1}{\ell^n} \sum_{p_i < p_j < \ldots < p_k \leq \ell} \left(\frac{\ell}{p_i \cdot p_j \cdot \ldots \cdot p_k}\right)^n \\
=\ & \left(\sum_{p \leq \ell} \frac{1}{p^n}\right)^k \\
\leq\ & \left(\sum_{p \text{ prime}} \frac{1}{p^n}\right)^k \\
\leq\ & \left(\sum_{j} \frac{1}{j^n}\right)^k \quad (j \in \mathbf{N} \setminus \{1\}) .
\end{aligned}$$

Since $n \geq 2$, this series is convergent. Therefore, each term in the expansion of $\frac{N_n(\ell)}{\ell^n}$ is convergent.

Recall the Möbius function μ :

$$\begin{aligned}\mu(1) &= 1\,,\\ \mu(m) &= \left\{\begin{array}{ll} 0 & \text{if } m \text{ is divisible by the square of a prime}\,,\\ (-1)^r & \text{if } m = p_1\cdot p_2\cdot\ldots\cdot p_r,\ \text{where } p_1,p_2,\ldots,p_r \text{ are all distinct primes}\,.\end{array}\right\}\end{aligned}$$

Euler showed that

$$\begin{aligned}&1-\sum_{p_i}\frac{1}{p_i^n}+\sum_{p_i<p_j}\frac{1}{(p_i\cdot p_j)^n}-\sum_{p_i<p_j<p_k}\frac{1}{(p_i\cdot p_j\cdot p_k)^n}+\ldots\\ =\ &\sum_m\frac{\mu(m)}{m^n}=[\zeta(n)]^{-1}\,.\end{aligned}$$

where the last sum is over $m\in\mathbf{N}$. For $n\geq 2$, this series is absolutely convergent. This last equality follows because for $j,k,m,n\in\mathbf{N}$,

$$\sum_m\frac{1}{m^n}\sum_j\frac{\mu(j)}{j^n}=\sum_{m,j}\frac{\mu(j)}{(mj)^n}=\sum_k\frac{1}{k^n}\sum_{d|k}\mu(d)=1\,,$$

where we have used the fact that both series in the first term converge uniformly and thus can be rearranged in any order.

Let

$$M_k=\left(\sum_j\frac{1}{j^n}\right)^k\,,$$

with sum over $j\in\mathbf{N}\setminus\{1\}$. Since $n\geq 2$ and the sum is over $j\in\mathbf{N}\setminus\{1\}$,

$$0<\sum_j\frac{1}{j^n}<1\,.$$

Since the k^{th} term in the expansion of $\frac{N_n(\ell)}{\ell^n}$ is dominated by M_k, and $\sum_k M_k$ is convergent, we may apply the Weierstrass M test, and evaluate term-by-term. We use the fact that $x-1\leq\lfloor x\rfloor\leq x$ for all x, and so $\lim_{x\to\infty}\frac{\lfloor x\rfloor}{x}=1$. Therefore, we have

$$\begin{aligned}&\lim_{\ell\to\infty}\frac{N_n(\ell)}{\ell^n}\\ =\ &\lim_{\ell\to\infty}\frac{1}{\ell^n}\left(\ell^n-\sum_{p_i}\left(\lfloor\frac{\ell}{p_i}\rfloor\right)^n+\sum_{p_i<p_j}\left(\lfloor\frac{\ell}{p_i\cdot p_j}\rfloor\right)^n-\sum_{p_i<p_j<p_k}\left(\lfloor\frac{\ell}{p_i\cdot p_j\cdot p_k}\rfloor\right)^n+\ldots\right)\\ =\ &1-\sum_{p_i}\frac{1}{p_i^n}+\sum_{p_i<p_j}\frac{1}{(p_i\cdot p_j)^n}-\sum_{p_i<p_j<p_k}\frac{1}{(p_i\cdot p_j\cdot p_k)^n}+\ldots\\ =\ &\sum_m\frac{\mu(m)}{m^n}\\ =\ &[\zeta(n)]^{-1}\,,\end{aligned}$$

where the last sum is over $m \in \mathbf{N}$. $\square$

Euler derived the following remarkable formula -

$$2\zeta(2k) = (-1)^{k+1}\frac{(2\pi)^{2k}}{(2k)!}(2k^{th} \text{ Bernoulli Number}) . \tag{10}$$

This allows us to generate the following table. We can now see why the algorithm works with as few as 10 data points.

Table 3: Some values of the Zeta function $\zeta(n)$ and $1/\zeta(n)$.

n	$\zeta(n)$	$1/\zeta(n)$
2	1.64493	0.6079
4	1.08232	0.9239
6	1.01734	0.9830
8	1.00407	0.9959
10	1.00099	0.9990
12	1.00024	0.9998
16	1.00001	1.0000

We can also estimate the convergence rate for all the integers.

Proposition 2.4 *Let $\omega \in (1, \infty)$. Then*

$$\lim_{\omega \to \infty} [\zeta(\omega)]^{-1} = 1 ,$$

converging to 1 from below faster than $(1 - 2^{1-\omega})$.

Proof : Since

$$\zeta(\omega) = \sum_{n=1}^{\infty} n^{-\omega}$$

and $\omega > 1$,

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

As $\omega \longrightarrow \infty$, $1/(1 - 2^{1-\omega}) \longrightarrow 1^+$. Thus, by the Squeezing Theorem,

$$[\zeta(\omega)]^{-1} \longrightarrow 1^- \text{ as } \omega \longrightarrow \infty . \quad \square$$

2.3 Variations of the Algorithm

In this section we consider the effects of noise and outliers on the modified Euclidean algorithm described above. The modified algorithm replaces division, as required in the standard Euclidean algorithm, with repeated subtraction in order to gain stability with respect to noise. Error analysis of this approach is complicated by the facts that the algorithm is iterative and that the algorithm sorts the data. Therefore, this analysis involves order statistics.

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

$$y_j = s_j - s_{j+1} = (k_j - k_{j+1})\tau + (\eta_j - \eta_{j+1}). \tag{11}$$

Invoking the zero-mean iid assumption on the η_j's, the pdf of $(\eta_j - \eta_{j+1})$ is given by the convolution $f_\eta(\eta) * f_\eta(\eta)$. So, for example, if $f_\eta(\eta) \sim \mathcal{U}[-\frac{\Delta}{2}, \frac{\Delta}{2}]$ (η is uniformly distributed with parameter Δ) 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 Reiss [17]. Without the iid assumption, this analysis leads into many open questions (see Reiss [17]).

In general, beyond the first iteration the pdf of the subsequent error terms becomes asymmetric, even when starting with iid η_j's with symmetric pdf $f_\eta(\eta)$. This occurs due to the reordering before differencing at each iteration, and because after the first iteration the errors are no longer iid. The result is that using the modified Euclidean algorithm can lead to negatively biased estimates of τ after the first iteration due to the skewness of the pdf of the errors. As we will see this can be corrected for by averaging.

In order to illustrate the behavior of the algorithm consider the following example. Let the set S of equation (1) be generated as follows. Let $\tau = 1$, $n = 100$ data samples, the jumps in the k_j's be randomly selected from a discrete uniform distribution on the interval $[1, 10]$, and the noise be iid and uniformly distributed as $f_\eta(\eta) \sim \mathcal{U}[-0.1, 0.1]$. A data set S was generated according to these parameters and used as input to our algorithm. Consider the results after one iteration, in which the data has been differenced and sorted into descending order, as plotted in Figure 1. The data are clustered into "steps" around integer multiples of $\tau = 1$, as we expect from (11). That the steps are all of the same approximate length is due to the uniform distribution in the jumps of the k_j's in the original data set S. Other distributions will result in different proportions. From (11) and the assumptions on the noise we know that the data has a mean that is an integer multiple of τ given by $(k_j - k_{j+1})\tau$, with noise symmetrically distributed around this mean. This suggests isolating each step and averaging the data within each step to reduce noise effects.

A straightforward method for clustering the data is to employ a gradient operator to determine when a step has occurred. After the first iteration (as in Figure 1) the gradient is estimated, with large gradient values indicating a step or "edge" in the data. We have employed a simple estimator by convolving with an impulse response given by $[-1, 0, 1]$. This operator is well known in signal and image processing. A data-adaptive gradient threshold

g_0 is selected as 10% of the maximum gradient value, and data points above this threshold are assumed to correspond to the step edges. After the steps have been isolated the step heights are easily found, and the minimum step height, call it $\tilde{\tau}$, is taken as a coarse estimate of τ. Referring to Figure 1, all of the step heights are approximately equal to τ, again due to the original distribution of the jumps in the k_j's used in generating S. We then use $\tilde{\tau}$ to set two thresholds. The first is $\eta_0 = 0.35\tilde{\tau}$, used to define the neighborhood of zero in which data will be eliminated during each iteration. The second we take to be $y_0 = 0.6\tilde{\tau}$, used to segment the steps at each iteration. The segmentation proceeds by searching for jumps in height greater than y_0, and averaging over each segment. The choices of 0.35 and 0.6 are based on extensive simulation experience, and can be more rigorously justified in specific cases. However, performance is reasonably robust to changes in these weights under the various scenarios considered. The averaging produces significant data reduction, and therefore greatly increases the speed of convergence. The gradient operator is applied only as part of the first iteration, the data reduces rapidly with each iteration and precludes use of the gradient operator except as part of the first iteration.

We summarize the foregoing in the following algorithm statement. Again we assume the data is initially sorted in descending order. Recall that appending zero in the first step appends the previous minimum.

Modified Euclidean Algorithm (With Averaging)

1.) After the first iteration, append zero.

2.) Form the new set with elements $s_j - s_{j+1}$.

3.) Sort in descending order.

4.) On the first iteration, apply gradient and obtain $\tilde{\tau}$, yielding $\eta_0 \cong 0.35\tilde{\tau}$ and $y_0 \cong 0.6\tilde{\tau}$ (see text).

5.) Average the data over each step, with steps determined by jumps of height y_0.

6.) Eliminate elements in $[0, \eta_0]$ from end of the set.

7.) Algorithm is done if left with a single element. Declare $\hat{\tau} = s_1$. If not done, go to **(1.)**.

Simulations show that this averaging makes the algorithm more robust. Results are shown in Table 4. The missing observations were modeled using a Bernoulli process with parameter λ, Δ is the noise parameter, and *iter* is the mean number of iterations required to converge. Note that λ corresponds to the expected percentage of missing observations.

The top half of Table 4 shows that estimates of τ degrade as the noise increases, as we would reasonably expect. The bottom half of Table 4 shows the effects of increasing λ, hence increasing the percentage of missing observations. The performance is essentially unchanged with $0.6 \leq \lambda \leq 0.9$. The last entry in Table 4 shows accurate estimation of τ from 100 data samples with $\pm 5\%$ phase jitter and 90% of the observations missing. It is possible to select $\lambda > 0.9$. However, estimation of $\tilde{\tau}$ becomes less reliable. For example, with $\lambda = 0.95$, 3 out of 100 trials resulted in poor estimates of $\tilde{\tau}$ that in turn resulted in poor estimates of τ, while the other 97 trials produced estimates close to $\tau = 1$.

Table 4: Estimation of τ from noisy measurements using the modified Euclidean algorithm (with averaging).

n	λ	Δ	$\widehat{\tau}$ (std)	*iter* (std)
100	.5	10^{-3}	1.0000 (.0001)	3.5 (.6)
100	.5	10^{-2}	1.0000 (.0014)	3.5 (.5)
100	.5	10^{-1}	1.0002 (.0169)	3.5 (.6)
100	.5	2×10^{-1}	1.0032 (.0212)	3.5 (.5)
100	.6	10^{-1}	0.9998 (.0158)	3.7 (.6)
100	.7	10^{-1}	0.9997 (.0200)	4.0 (.6)
100	.8	10^{-1}	1.0002 (.0202)	3.9 (.4)
100	.9	10^{-1}	1.0038 (.0213)	4.2 (.6)

If the data is such that, after the first iteration, there is a relatively large cluster around the step nearest to zero, then we can readily estimate τ by finding this step, averaging only over these data points, and declaring this to be $\widehat{\tau}$. This is the rightmost or lowest step after the first iteration (see Figure 1). Under our assumptions this mean is an unbiased estimate of τ. Accurate estimation of τ from a single iteration assumes that n is large enough, and the span of the original data set S small enough, to yield sufficient data in the neighborhood of τ. This is a function of the distribution of the k_j's.

In our example above the missing observations were modeled by taking the jumps in the k_j's as uniformly distributed on the discrete interval $[1, M]$, with $M = 10$. Thus, for large n, after the first iteration the data will cluster in M steps with an expected value of n/M samples in each step, as in Figure 1. As another model for missing observations we can employ an iid Bernoulli process to determine if an observation is missing or not, where

$$\begin{aligned} P(\text{missing observation}) &= \lambda \\ P(\text{observation occurring}) &= 1 - \lambda. \end{aligned} \tag{12}$$

For example, with $\lambda = 0.6$, we expect 60% of the observations to be missing. Given an element of S, $s_j = k_j\tau + \eta_j + \phi$, then the probability that $s_{j+1} = k_{j+1}\tau + \eta_{j+1} + \phi = (k_j + 1)\tau + \eta_{j+1} + \phi$ is an element of the set S is given by $1 - \lambda$. It follows that, after the first iteration, we expect $(1 - \lambda)(n - 1)$ data samples to be in the lowest step clustered around the true value of τ. Thus, if $n = 101$ and $\lambda = 0.6$, then after the first iteration we expect

$(1-0.6)100 = 40$ data samples to be clustered in the lowest step around τ; the average over these 40 data samples may then be taken as an estimate of τ.

The single iteration algorithm is a simple modification of the preceding multi-iteration version. After the (first) difference and sort operations the gradient is applied and the lowest step isolated. The average over this step is then taken as our estimate of τ. We summarize the single iteration form of the algorithm as follows.

Modified Euclidean Algorithm (Single 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.) Apply gradient operator and obtain $\tilde{\tau}$, with $y_0 \cong 0.6\tilde{\tau}$.

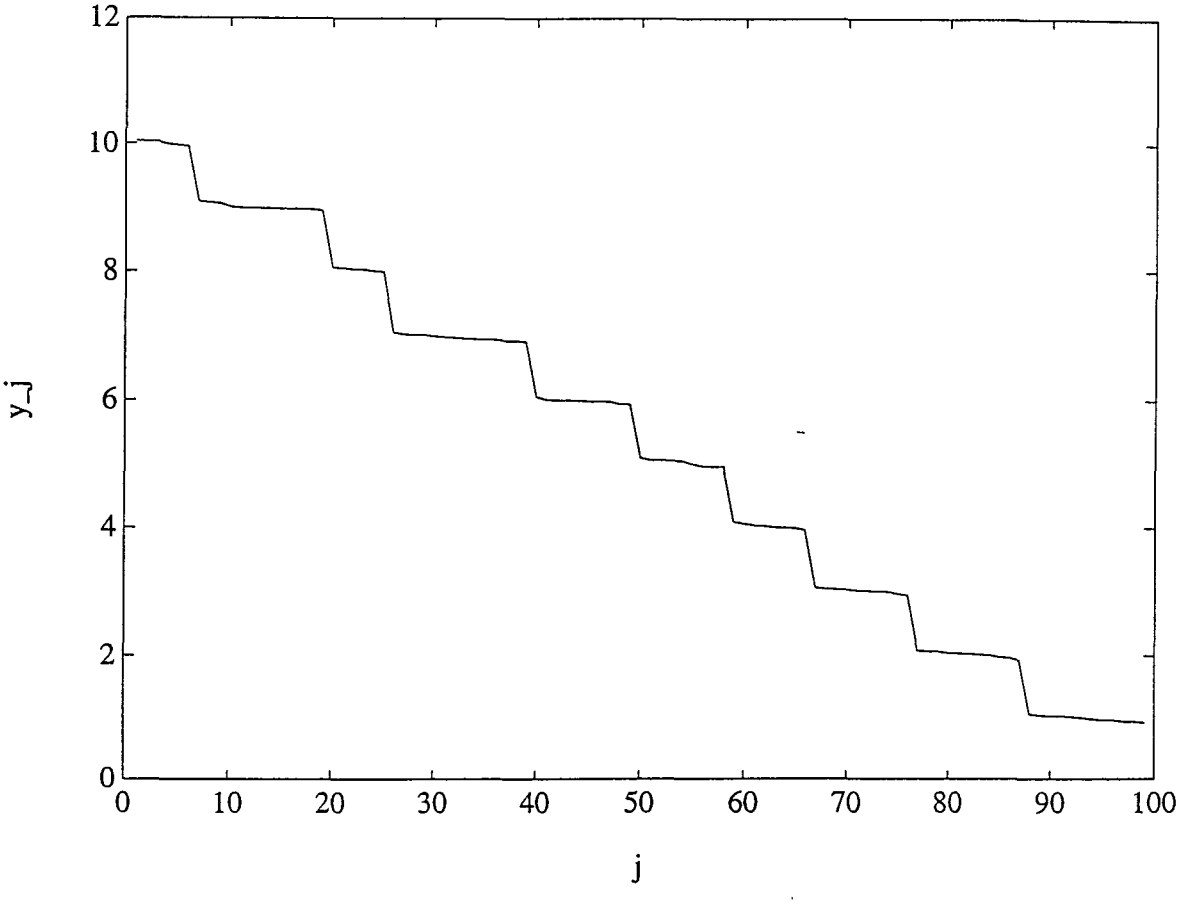
4.) Obtain $\hat{\tau}$ by averaging the data over the lowest step, isolating this step based on the lowest two jumps of height y_0.

Next we consider the effect of arbitrary outliers. These are, in general, quite harmful to the estimation of the gcd. This is easily seen in the noise-free case because the gcd of a contaminated set may be arbitrarily different from the gcd of the uncontaminated set. Testing of the algorithms presented thus far shows sensitivity to the presence of even a single outlier. This is because the outliers will not necessarily fall into the step-like clusters we expect.

A more robust version of the algorithm can be obtained with the introduction of a step-width threshold, x_0. Let us concentrate on the single iteration approach. After application of the gradient operator as before, we avoid false edges and clusters occurring below the true lowest step by requiring the step-width (the number of data samples in a particular step) to be greater than x_0.

The choice of x_0 is dictated by two considerations, the distribution of the outliers and the expected number of data samples in the lowest step after the first iteration. This represents a tradeoff. For example, consider again the Bernoulli model for missing observations. Suppose $\lambda = 0.25$ and $n = 101$. We therefore expect 75 data samples in the lowest step, allowing us to choose $x_0 = 50$, say. The single iteration algorithm employing x_0 will therefore search for the lowest step in the data that contains at least $x_0 = 50$ data samples. Depending on the distribution and number of outliers it may be extremely unlikely that a step will occur that simultaneously has a mean less than τ and has more than $x_0 = 50$ data values. The drawback to employing x_0 is that setting its value requires some *a priori* knowledge or guesswork and is not easily set adaptively. The advantage is that for reasonable scenarios its use makes the algorithm very robust to the presence of even large percentages of outliers, as illustrated in the simulation results of the next section. This robust single iteration algorithm is summarized as follows.

Figure 1



1. Plot of example data set after one iteration of the modified Euclidean algorithm of Section 2. The data is sorted in descending order into steps centered around multiples of τ ($\tau = 1$ in this example). The stepwidths are a function of the distribution of the k_j's in the original data set S. The lowest (rightmost) step is centered around the true value of $\tau = 1$.

Robust Single Iteration Algorithm

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.) Apply gradient operator and obtain $\tilde{\tau}$, with $y_0 \cong 0.6\tilde{\tau}$.

4.) Obtain $\hat{\tau}$ by averaging the data over the lowest step, isolating the step based on the lowest two jumps of height y_0 with step-width greater than x_0.

See [2] for simulation of these last two algorithms.

3 PRI Analysis

The parameter estimation techniques given above lead to an effective method for periodic pulse interval analysis (see [4, 18]). We assume time is highly resolved and ignore any time quantization error. We are primarily concerned with a single periodic pulse train with (perhaps very many) missing observations that may be contaminated with outliers. Our data model for this case, in terms of the arrival times t_j, is given by (1), with the additional assumption that η_j is zero-mean additive white Gaussian noise. Outliers are included as arbitrary arrival times. The problem, again, is to recover the period τ and possibly the phase ϕ. With Gaussian noise the minimum variance unbiased estimates for this linear regression problem take a least-squares form. However, this requires knowledge of the k_j's. We therefore propose a multi-step procedure that proceeds by (i) estimating τ directly, (ii) estimating the k_j's, and (iii) refining the estimate of τ using the estimated k_j's in the least-squares solution. This estimate is shown to perform well, achieving the Cramer-Rao bound in many cases, despite many missing observations and contaminated data. The direct estimate of τ (step (i)) is obtained using the modified Euclidean algorithms described above. While not maximum-likelihood (ML), the modified Euclidean algorithm performs well under difficult conditions.

We now give the maximum likelihood solution and Cramer-Rao bounds for estimating τ and ϕ. Our analysis has led us to work with the data set $\{t_{j+1} - t_j\}_{j=1}^{n-1}$, so as to avoid estimating ϕ (which can be unreliable). Given the sample data set S from (1) we may write

$$\begin{bmatrix} t_1 \\ t_2 \\ \vdots \\ t_n \end{bmatrix} = \begin{bmatrix} 1 & k_1 \\ 1 & k_2 \\ \vdots & \vdots \\ 1 & k_n \end{bmatrix} \begin{bmatrix} \phi \\ \tau \end{bmatrix} + \begin{bmatrix} \eta_1 \\ \eta_2 \\ \vdots \\ \eta_n \end{bmatrix}, \tag{13}$$

where $k_{j+1} > k_j$. In compact form this is

$$\mathbf{t} = X\boldsymbol{\beta} + \boldsymbol{\eta}, \tag{14}$$

where $\boldsymbol{\beta} = [\phi, \tau]^T$ and the definitions of $\mathbf{t}$, $\boldsymbol{\eta}$, and X follow from (13). We eliminate ϕ by forming the differences $y_j = t_{j+1} - t_j = (k_{j+1} - k_j)\tau + (\eta_{j+1} - \eta_j)$, yielding

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_{n-1} \end{bmatrix} = \begin{bmatrix} k_2 - k_1 \\ k_3 - k_2 \\ \vdots \\ k_n - k_{n-1} \end{bmatrix} \tau + \begin{bmatrix} \delta_1 \\ \delta_2 \\ \vdots \\ \delta_{n-1} \end{bmatrix}, \tag{15}$$

where $\delta_j = \eta_{j+1} - \eta_j$. Similar to (14) we may write (15) compactly as

$$\mathbf{y} = X_d \tau + \boldsymbol{\delta}. \tag{16}$$

Equations (14) and (16) are linear regression problems whose least squares solutions yield the minimum-variance unbiased estimate when the noise is zero-mean Gaussian, e.g., see Kay [10]. Generally, use of (16) is preferred for estimating τ, avoiding estimation of ϕ which has high variance. The solution to (16) corresponds to ML estimation and takes the form of a least squares estimate

$$\hat{\tau} = (X_d^T R_\delta^{-1} X_d)^{-1} X_d^T R_\delta^{-1} \mathbf{y}, \tag{17}$$

where $R_\delta = E[\boldsymbol{\delta}\boldsymbol{\delta}^T]$. We have assumed white noise so $R_\delta = \sigma_\eta^2 \tilde{R}_\delta$ where $\tilde{R}_\delta$ has 2's on the main diagonal, -1's on the first upper and lower diagonals, and zeros elsewhere. In general R_δ is full rank and its inverse can be expressed element-wise as $[R_\delta^{-1}]_{ij} = \min(i,j) - ij/n$, and is therefore easily computed. Although optimal, use of (17) requires knowledge of X_d. This is not a problem if there are no missing observations for then $k_j = j$ for $j = 1, 2, \ldots n$. However, when observations are arbitrarily missing then the k_j's are not known in general, and one is faced with more unknowns than equations in (16).

The pdf of the noise $\boldsymbol{\eta}$ in (14) is multivariate Gaussian, leading to the Cramer-Rao bound (CRB) for (17)

$$\text{var}\{\tau - \hat{\tau}\} \geq \sigma_\delta^2 (X_d^T \tilde{R}_\delta^{-1} X_d)^{-1}, \tag{18}$$

with $\sigma_\delta = 2\sigma_\eta$. Generally, the CRB is reduced for smaller σ_η^2. Also, for fixed n, it is reduced when the spread of the k_j's increases.

Now, if τ were known then X_d could be estimated using $(1/\tau)\,\mathbf{y}$. Ideally, this estimate is composed of positive integers, but imperfect knowledge of τ and the presence of noise will generally yield an estimate of X_d that has non-integer components. We therefore propose to estimate X_d via

$$\widehat{X}_d = \text{round}\left[\frac{1}{\hat{\tau}_{MEA}}\,\mathbf{y}\right], \tag{19}$$

where $\hat{\tau}_{MEA}$ is the estimate of τ obtained via the modified Euclidean algorithm, and

$$\text{round}[\cdot] = \lfloor \cdot + \frac{1}{2} \rfloor$$

is rounding to the nearest integer. A refined estimate of τ is then obtained by using $\widehat{X}_d$ in (17) yielding

$$\hat{\tau} = (\widehat{X}_d^T R_\delta^{-1} \widehat{X}_d)^{-1} \widehat{X}_d^T R_\delta^{-1} \mathbf{y}. \tag{20}$$

This result approaches the optimal minimum variance performance when $\widehat{X}_d$ is close to X_d. The refinement algorithm is summarized as follows.

Refined Estimation Algorithm

1.) Estimate τ via the modified Euclidean algorithm, calling this estimate $\widehat{\tau}_{MEA}$.

2.) Estimate $\widehat{X}_d$ via (19).

3.) Refine the estimate of τ using $\widehat{X}_d$ in (20), calling this estimate $\widehat{\tau}$.

Performance analysis of the estimate $\widehat{\tau}_{MEA}$ depends not only on the distribution of the noise η_j, but also on the distribution of the k_j's. We have completed this analysis for some specific cases in [4]. We also compare the estimates to Cramer-Rao bounds via Monte Carlo simulation, revealing the very good performance of the algorithm with many missing observations and contaminated data (see [4, 18]). We can also apply our estimation procedures to estimation of the frequency of a single sinusoid in Gaussian noise. We address the problem [19], using only very sparse noisy zero-crossings with the presence of outliers.

Table 5: Comparison of data from figure 2a.

% jitter	$\widehat{T}_{MEA}$ mean(std)	$\widehat{T}$ mean(std)	$\sqrt{\text{CRB}}$
2	1.0000 (0.0007)	1.0000 (1.826×10^{-5})	1.829×10^{-5}
4	1.0000 (0.0011)	1.0000 (3.587×10^{-5})	3.669×10^{-5}
6	1.0003 (0.0016)	1.0000 (5.056×10^{-5})	5.532×10^{-5}
8	1.0000 (0.0022)	1.0000 (7.427×10^{-5})	7.366×10^{-5}
10	1.0002 (0.0028)	1.0000 (9.275×10^{-5})	9.228×10^{-5}
15	0.9998 (0.0043)	1.0000 (1.438×10^{-4})	1.387×10^{-4}
20	1.0000 (0.0054)	1.0000 (1.907×10^{-4})	1.836×10^{-4}
25	0.9999 (0.0072)	1.0000 (2.210×10^{-4})	2.308×10^{-4}
30	1.0007 (0.0084)	1.0000 (1.583×10^{-3})	2.758×10^{-4}
35	0.9999 (0.0110)	0.9994 (3.401×10^{-3})	3.222×10^{-4}
40	1.0008 (0.0139)	0.9996 (6.216×10^{-3})	3.672×10^{-4}
45	1.0052 (0.0426)	0.9999 (3.285×10^{-2})	4.178×10^{-4}
50	1.0133 (0.0741)	1.0041 (5.830×10^{-2})	4.792×10^{-4}

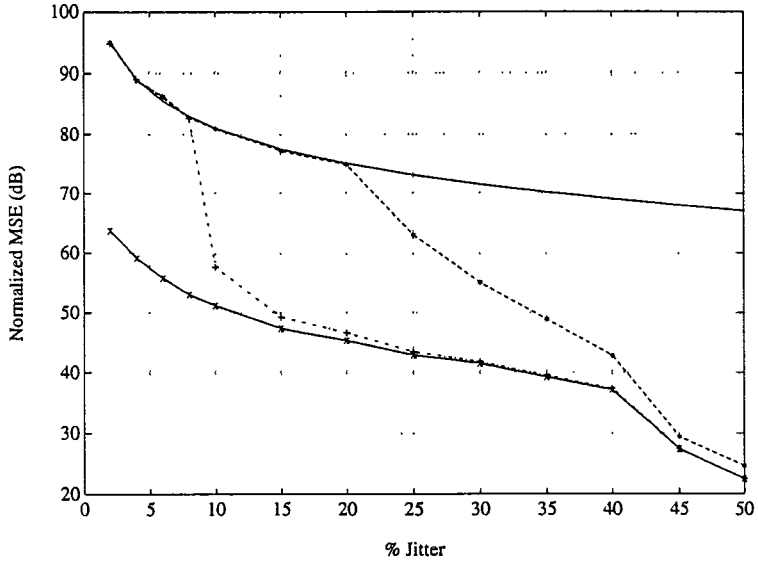


Figure 2a: Monte Carlo estimation results (without outliers, $\lambda = 0.25, N = 100, x_0 = 5$). Solid = CRB, x = $\widehat{T}_{MEA}$, dash (*) = $\widehat{T}$, and dash-dot (+) = $\widehat{T}$ incorporating $\widehat{\phi}_x$.

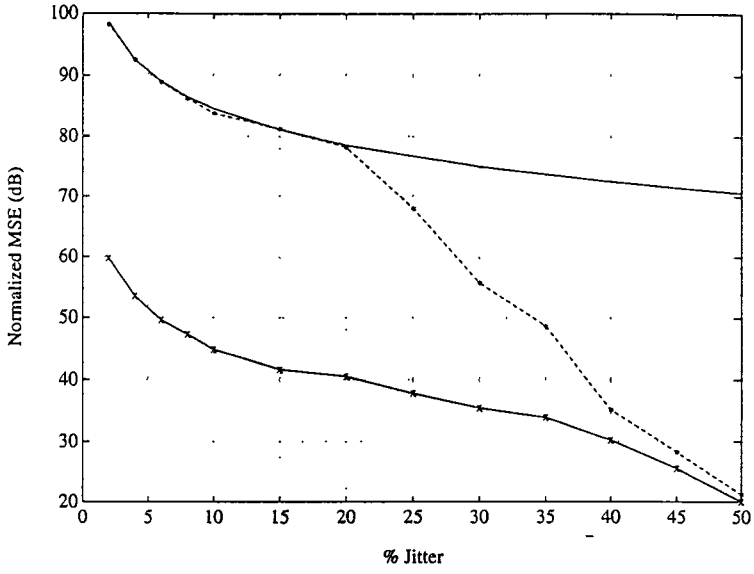


Figure 2: Monte Carlo estimation results (without outliers, $\lambda = 0.50, N = 100, x_0 = 5$). Solid = CRB, x = $\widehat{T}_{MEA}$, and dash (*) = $\widehat{T}$.

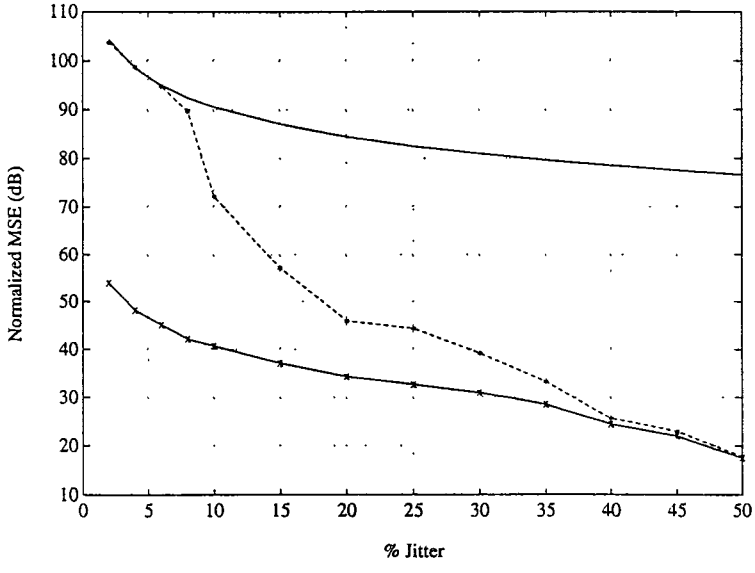


Figure 3: Monte Carlo estimation results (without outliers, $\lambda = 0.75, N = 100, x_0 = 5$). Solid = CRB, x = $\widehat{T}_{MEA}$, and dash (*) = $\widehat{T}$.

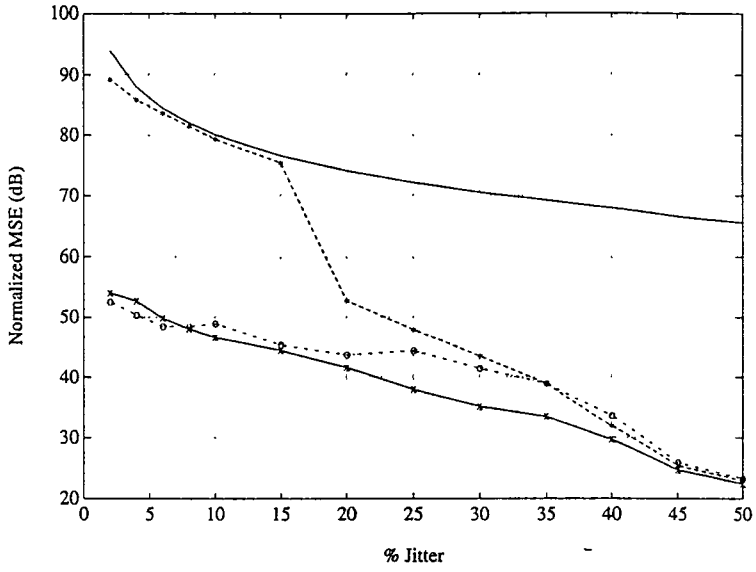


Figure 4: Monte Carlo estimation results (with 5% outliers, $\lambda = 0.25, N = 100, x_0 = 15$). Comparison of $\widehat{T}$ versus $\widehat{T}'$ (full versus selected data reuse in the refinement algorithm). Solid = CRB, x = $\widehat{T}_{MEA}$, dash-dot (o) = $\widehat{T}$, dash (*) = $\widehat{T}'$.

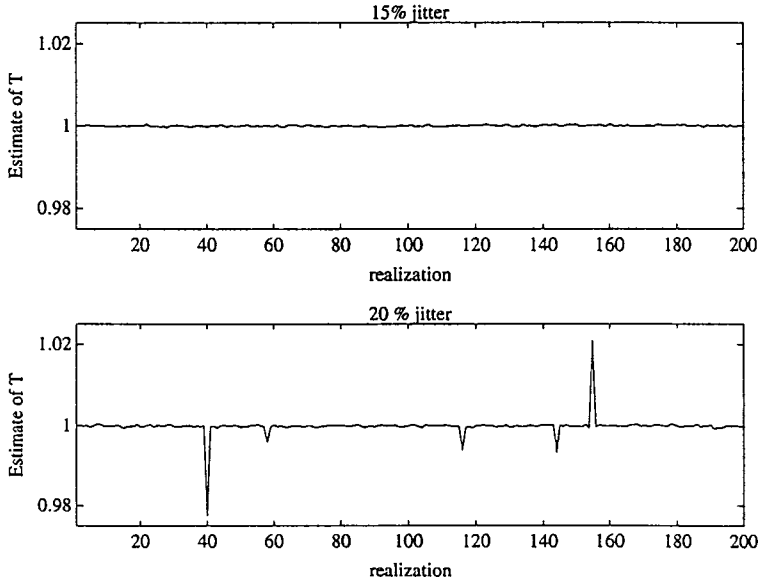


Figure 5: Realizations of $\widehat{T}'$ for different percentages of jitter near threshold, with experimental parameters matching those of figure 4.

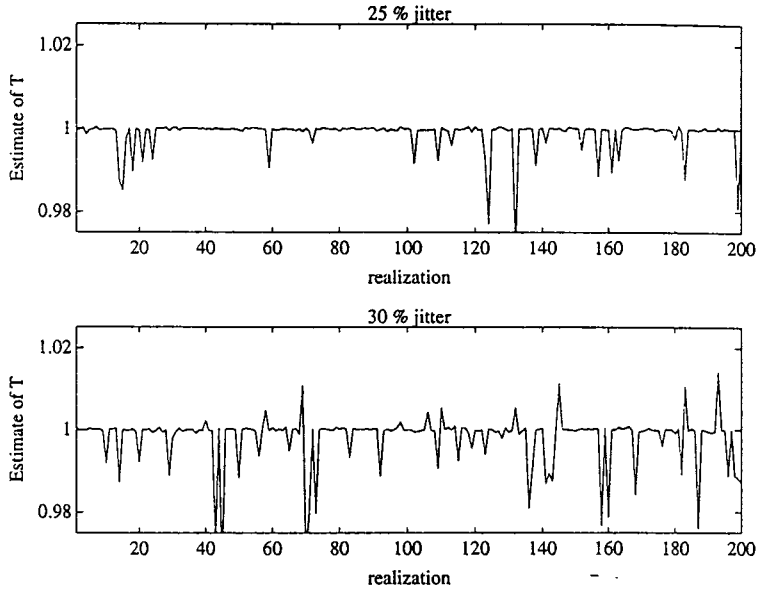


Figure 6: Realizations of $\widehat{T}'$ for different percentages of jitter above threshold, with experimental parameters matching those of figure 4.

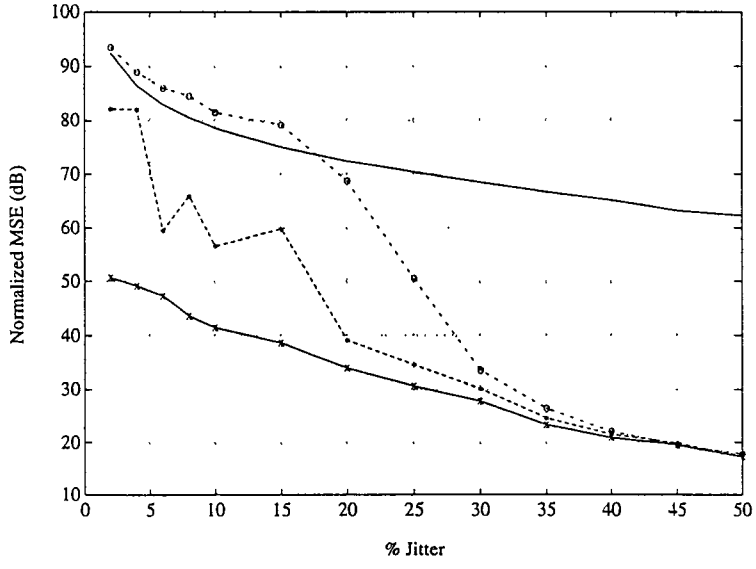


Figure 7: Multiple realization results. Solid = CRB, x = $\widehat{T}_{MEA}$, and dash (*) = $\widehat{T}'$, for single data records ($\lambda = 0.25$, $N = 100$, $x_0 = 10$, and 15% outliers). Also, dash-dot (o) = $\widehat{T}_M$ ($N_r = 5$ data records).

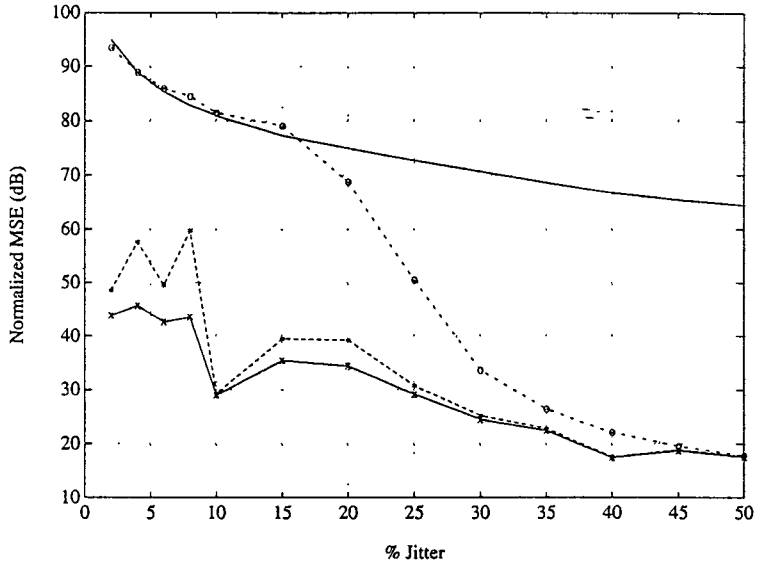


Figure 8: Multiple realization results. Solid = CRB, x = $\hat{T}_{MEA}$, and dash (*) = $\hat{T}'$, for single data records ($\lambda = 0.5$, $N = 100$, $x_0 = 10$, and 15% outliers). Also, dash-dot (o) = $\hat{T}_M$ ($N_r = 5$ data records).

4 Deinterleaving

We close by discussing our work on deinterleaving. Our data model is the union of M copies of (1), each with different periods or "generators" $\Gamma = \{\tau_i\}$, k_{ij}'s and phases. Let $\tau = \max_i\{\tau_i\}$. Then our data is

$$\Delta = \bigcup_{i=1}^{M} \{\phi_i + k_{ij}\tau_i + \eta_{ij}\}_{j=1}^{n_i} , \tag{21}$$

where n_i is the number of elements from the i^{th} generator, $\{k_{ij}\}$ is a linearly increasing sequence of natural numbers with missing observations, ϕ_i is a random variable uniformly distributed in $[0, \tau_i)$, and the 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 $\Delta = \{\alpha_l\}_{l=1}^{N}$. 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

$$\Phi_\rho(\alpha_l) = \left\langle \frac{\alpha_l}{\rho} \right\rangle = \frac{\alpha_l}{\rho} - \left\lfloor \frac{\alpha_l}{\rho} \right\rfloor , \tag{22}$$

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

$$\frac{1}{n}\text{card}\,\{1 \leq j \leq n : x_j \in [a, b]\} \longrightarrow (b - a) \tag{23}$$

as $n \longrightarrow \infty$ almost surely.

Weyl's Theorem is presented in [6]. For our variation, we assume that for each i, $\{k_{ij}\}$ is a linearly increasing infinite sequence of natural numbers with missing observations such that $k_{ij} \longrightarrow \infty$ as $j \longrightarrow \infty$. We must make this assumption because the result is only approximately true for a finite length sequence.

Theorem 4.1 *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.*

Moreover, the set of ρ's for which this is not true are rational multiples of $\{\tau_i\}$. Therefore, except for those values, $\Phi_\rho(\alpha_{ij})$ is essentially uniformly distributed in $[0, 1)$. Moreover, the values at which $\Phi_\rho(\alpha_{ij}) = 0$ almost surely are $\rho \in \{\tau_i/n : n \in \mathbf{N}\}$. These values of ρ cluster at zero, but spread out for lower values of n.

We then map the phase wrapped data by non-linear variations on the periodogram,

$$F(\alpha_l, \rho) = \frac{1}{N}\sum_l \cos^{2r-1}(2\pi\frac{\alpha_l}{\rho}) + i\frac{1}{N}\sum_l \sin^{2r-1}(2\pi\frac{\alpha_l}{\rho}) , \tag{24}$$

for $r = 2, 3, \ldots$. Now, the periodicity of sin and cos gives us that

$$\cos^{2r-1}(2\pi\Phi_\rho(\alpha_l)) = \cos^{2r-1}(2\pi\frac{\alpha_l}{\rho})$$

and

$$\sin^{2r-1}(2\pi\Phi_\rho(\alpha_l)) = \sin^{2r-1}(2\pi\frac{\alpha_l}{\rho})\,.$$

By Theorem 4.1, the random variables $\Phi_\rho(\alpha_l)$ are uniformly distributed on $[0, 1)$ for almost every choice of ρ. We can then compute the distributions of the real and imaginary parts of F. The "noise-like" behavior of $\Phi_\rho(\alpha_l)$ for *a.e.* ρ leads to a "flat" range for F. However, at $\rho \in \{\tau_i/n : n \in \mathbf{N}\}$, we have increasingly strong peaks as n decreases. In turn, this gives the following. Let i_0 denote the index of the most prolific generator, and $\Re$, $\Im$ denote the real and imaginary parts.

Theorem 4.2

$$\max_\rho (\Re F - |\Im F|) = \tau_{i_0}\,. \tag{25}$$

We then isolate the data generated by τ_{i_0} by convolution with a pulse train of width τ_{i_0}, and subtract it out. We then repeat the process, terminating when Δ equals the empty set.

References

[1] Bartlett, M. S., "The spectral analysis of point processes," *Journal of the Royal Statistical Society B*, Vol. 25, No. 2, pp. 264-280, 1963.

[2] Casey, S. D., and Sadler, B. M., "Modifications of the Euclidean algorithm for isolating periodicities from a sparse set of noisy measurements," *IEEE Trans. Sig. Proc.*, 31 pp., to appear 1996.

[3] Casey, S. D., and Sadler, B. M., "A Modification of the Euclidean algorithm for isolating periodicities from a sparse set of noisy measurements," *IEEE Int. Conf. Acoustics, Speech, and Sig. Proc. (ICASSP '95)*, Vol. 3, pp. 1764-1767, 1995.

[4] Casey, S. D., and Sadler, B. M., "Number theoretic methods in parameter estimation," *IEEE Conf. Stat. Sig. and Array Proc. (SSAP '96)*, 4 pp., 1996.

[5] Casey, S. D., "Variations on Weyl's equidistribution theorem with applications to deinterleaving," 28 pp., preprint.

[6] Dym, H., and McKean, H. P., *Fourier Series and Integrals*, Academic Press, Orlando, Florida, 1972.

[7] Hardy, G. H., *A Mathematician's Apology*, Cambridge University Press, Cambridge, 1940.

[8] Hardy, G. H., and Wright, E. M., *An Introduction to the Theory of Numbers*, Oxford University Press, New York, 1982.

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

[10] Kay, S. M., *Fundamentals of Statistical Signal Processing*, Prentice-Hall, New Jersey, 1993.

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

[12] Knuth, D. E., *The Art of Computer Programming, Volume 2: Seminumerical Algorithms (Second Edition)*, Addison-Wesley, Reading, Massachusetts, 1981.

[13] Knuth, D. E., *The Art of Computer Programming, Volume 3: Sorting and Searching*, Addison-Wesley, Reading, Massachusetts, 1973.

[14] Leveque, W. J., *Topics in Number Theory, Volumes 1 and 2*, Addison-Wesley, Reading, Massachusetts, 1956.

[15] Papoulis, A., *Signal Analysis*, McGraw-Hill, New York, 1977.

[16] Papoulis, A., *Probability, Random Variables, and Stochastic Processes (Third Edition)*, McGraw-Hill, New York, 1991.

[17] Reiss, R. -D., *Approximate Distributions of Order Statistics*, Springer-Verlag, New York, 1989.

[18] Sadler, B. M., and Casey, S. D., "PRI analysis from sparse data via a modified Euclidean algorithm," invited paper for the *29th Annual Asilomar Conference on Signals, Systems, and Computers*, 6 pp., 1995.

[19] Sadler, B. M., and Casey, S. D., "On periodic pulse interval analysis with outliers and missing observations," submitted to *IEEE Trans. Sig. Proc.*, 31 pp.

[20] Sadler, B. M., and Casey, S. D., "Frequency estimation via sparse zero crossings," accepted for *IEEE Int. Conf. Acoustics, Speech, and Sig. Proc. (ICASSP '96)*, 4 pp., 1996.

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

[22] Wiley, R. G., *Electronic Intelligence: The Analysis of Radar Signals (Second Edition)*, Artech House, Norwood, Massachusetts, 1993.