Med Biol Eng Comput (2009) 47:697–707
DOI 10.1007/s11517-009-0461-0

ORIGINAL ARTICLE

A novel recursive Fourier transform for nonuniform sampled signals: application to heart rate variability spectrum estimation

Alexander Holland · Mateo Aboy

Received: 4 May 2008 / Accepted: 30 January 2009 / Published online: 27 February 2009

Abstract We present a novel method to iteratively calculate discrete Fourier transforms for discrete time signals with sample time intervals that may be widely nonuniform. The proposed recursive Fourier transform (RFT) does not require interpolation of the samples to uniform time intervals, and each iterative transform update of N frequencies has computational order N. Because of the inherent non-uniformity in the time between successive heart beats, an application particularly well suited for this transform is power spectral density (PSD) estimation for heart rate variability. We compare RFT based spectrum estimation with Lomb–Scargle Transform (LST) based estimation. PSD estimation based on the LST also does not require uniform time samples, but the LST has a computational order greater than $N\log(N)$. We conducted an assessment study involving the analysis of quasi-stationary signals with various levels of randomly missing heart beats. Our results indicate that the RFT leads to comparable estimation performance to the LST with significantly less computational overhead and complexity for applications requiring iterative spectrum estimations.

Keywords Discrete Fourier transform · Spectrum · Power spectral density · Nonuniform sampling · Heart rate variability

A. Holland (✉)
Draeger Medical, Telford, PA, USA
e-mail: alexander.holland@draeger.com

A. Holland
Oregon Health Sciences Univeristy, Telford, PA, USA

M. Aboy
Oregon Institute of Technology, Beaverton, OR, USA
e-mail: mateoaboy@ieee.org

1 Introduction

We describe a recursive least squares Fourier transform (RFT) for nonuniform sampled signals. The RFT does not require interpolation of samples to uniform time intervals, and is robust to missed samples due to signal loss or dropout. Additionally, it can be used to perform power spectral density (PSD) estimation directly on nonuniform sampled signals. PSD estimation methods based on the Lomb–Scargle transform (LST) achieve these same characteristics using least squares estimation [10, 16]. Although fast LST methods reduce computation by using the fast Fourier transform (FFT) for internal computations [13], the LST remains primarily an off-line analysis method. Real-time usage of the LST on overlapped, sliding, weighted time windows (e.g. Hamming, Bartlett, etc.) is possible [21], but the best computational efficiency for each iteration, including iterations using a single new sample, remains of order $N\log(N)$, where N is the number of frequencies in the estimated discrete PSD.

Because of the inherent non-uniformity in the time between successive heart beats, an application particularly well suited for these methods is the estimation of the heart rate variability (HRV) spectrum or time–frequency plane that are common in HRV based diagnostic methods [8, 11, 14, 17]. The HRV spectrum is the PSD of the instantaneous heart rate signal, $h(n)$. $h(n)$ is the reciprocal of the variation in beat to beat time intervals, $r(n)$, or "RR intervals" between successive heart beats. "R" refers to the peak of the R wave in the electrocardiogram (ECG) "QRS complex." These RR intervals are themselves nonuniform spaced in time, and beats are often missing due to environmental noise or sometimes due to data networks losing packets carrying signal data.

FFT based spectrum estimation methods that first interpolate heart beat time interval data to uniformly spaced intervals have significant HRV spectrum estimation error that is avoided by the LST [9]. In addition to the LST, adaptive filter banks of finite impulse response (FIR) bandpass filters can be used to provide real-time estimates of the total energy in frequency bands of the HRV spectrum [19]. The filter bank linearly interpolates the coefficients of the FIR filters to adaptively adjust for the nonuniform sampling period. The method is more computationally efficient than the LST, but does not compute the detailed power spectrum and requires custom digital filter design to change the desired frequency bands.

The RFT is an on-line transform based on recursive least squares (RLS) that iteratively updates a previous estimate using each new data sample. Each update has computational order N, where N is the number of frequency coefficients, and can be used to perform PSD estimation directly on nonuniform sampled signals. Additionally, the RFT generalizes to estimate other types of transforms corresponding to other types of discrete orthogonal basis functions.

RLS has a long venerable history, particularly in signal processing fields using linear model fitting, such as adaptive linear filtering, linear predictive coding, or parametric spectrum estimation. General computationally efficient variations on RLS employ matrix factorizations or exploit matrix structure [3, 15]. QR matrix factorization, for example, reduces the computational order of the RLS recursion from N^3 to N^2. The Levinson–Durbin variation exploits the Toeplitz structure of the normal equations to achieve a reduction in computational order from N^3 to N^2. Closely related Schur recursions have the same affect, but are more amenable to parallel implementation. Fast adaptive transversal filtering methods reduce computational order from N^3 to N [2, 7, 15], and use of the FFT can reduce RLS computational order from N^3 to $N\log(N)$ [12].

In this paper, we present the mathematical derivation for the RFT, a specific type of efficient RLS iteration using discrete complex Fourier exponentials as basis functions. We show illustrative spectrum estimation application examples and the results of a HRV spectrum estimation performance study that compares performance using the RFT versus the LST.

2 Transform methods

2.1 Lomb–Scargle

For N discrete time samples $r(n) = r(t_n)$ with nonuniform sample times t_n, the Lomb–Scargle transform, $L_N(\omega)$, is defined as follows: [10, 16],

$$L_N(\omega) = \frac{1}{2\sigma^2}\frac{\left[\sum_i (r(i)-\bar{r})\cos\omega(t_i-\tau)\right]^2}{\sum_i \cos^2\omega(t_i-\tau)} + \frac{1}{2\sigma^2}\frac{\left[\sum_i (r(i)-\bar{r})\sin\omega(t_i-\tau)\right]^2}{\sum_i \sin^2\omega(t_i-\tau)} \tag{1}$$

where τ is defined by

$$\tan(2\omega\tau) = \frac{\sum_i \sin 2\omega t_i}{\sum_i \cos 2\omega t_i} \tag{2}$$

and

$$\bar{r} = \frac{1}{N}\sum_{i=0}^{N-1} r(i) \tag{3}$$

$$\sigma^2 = \frac{1}{N-1}\sum_{i=0}^{N-1}(r(i)-\bar{r})^2$$

$L_N(\omega)$ represents the PSD. The variable t_n is typically time, but may also represent a discrete value of any other continuous variable such as space. A fast algorithm with complexity on the same order as an FFT exists for this transform [13].

2.2 Fast recursive least squares

The vector/matrix equations,

$$\begin{aligned}\mathbf{r}(n) &= \begin{bmatrix} r(t_1) \\ r(t_2) \\ \vdots \\ r(t_n) \end{bmatrix} \\ &= \frac{1}{N}\begin{bmatrix} \mathbf{b}^T(1) \\ \mathbf{b}^T(2) \\ \vdots \\ \mathbf{b}^T(n) \end{bmatrix}\begin{bmatrix} w_0(n) \\ w_1(n) \\ \vdots \\ w_{N-1}(n) \end{bmatrix} \\ &= \frac{1}{N}\mathbf{B}(n)\mathbf{w}(n)\end{aligned} \tag{4}$$

represent a general inverse transform from N discrete transform coefficients $w_k(n)$ corresponding to discrete time samples, $r(t_n)$. $w_k(n)$ are coefficients that correspond to frequencies f_k, and

$$\mathbf{b}^T(n) = [\, b_0(n) \quad b_1(n) \quad \ldots \quad b_{N-1}(n)\,] \tag{5}$$

$$b_k(n) = \mathrm{e}^{j2\pi f_k t_n} \quad \text{and} \quad j = \sqrt{-1}$$

For the analysis below, assume equally spaced discrete frequencies,

$$f_k = \frac{kf_s}{N} \tag{6}$$

for $k = 0, 1, \ldots, N-1$, and unequally spaced, but quantized time samples, $t_n = n/f_s$, where $n \in \{0, 1, \ldots, \infty\}$ and $1/f_s$ is

the time measurement quantization. With frequency as the independent variable and time as constant, the rows of $\mathbf{B}(n)$ are the same basis functions as in the inverse discrete Fourier transform (DFT), and hence are orthogonal despite the unequally spaced time samples, as long as no two time samples are equal modulo N/f_s (in which case the rows would be equal), i.e.

$$\mathbf{b}^T(n) * \mathbf{b}^*(m) = \begin{cases} 0 & \text{for} \quad t_m f_s \neq t_n f_s \quad \text{mod}(N) \\ N & \text{for} \quad t_m f_s = t_n f_s \quad \text{mod}(N) \end{cases} \tag{7}$$

where $\mathbf{b}^*$ (m) is the complex conjugate of $\mathbf{b}(m)$, but does not indicate transpose.

The weighted least mean-square error solution to Eq. 4 is given by

$$\begin{aligned} \mathbf{w}(n) &= N[\mathbf{B}'(n)\mathbf{\Lambda}(n)\mathbf{B}(n)]^{-1}\mathbf{B}'(n)\mathbf{\Lambda}(n)\mathbf{r}(n) \\ &= N\mathbf{P}^{-1}(n)\mathbf{B}'(n)\mathbf{\Lambda}(n)\mathbf{r}(n) \end{aligned} \tag{8}$$

where

$$\mathbf{P}^{-1}(n) = [\mathbf{B}'(n)\mathbf{\Lambda}(n)\mathbf{B}(n)]^{-1} \tag{9}$$

and $\mathbf{B}'(n)$ is the complex conjugate of the transpose of $\mathbf{B}(n)$, i.e. $\mathbf{B}'(n) = \mathbf{B}^T(n)^*$, and $\mathbf{\Lambda}(n)$ is a symmetric, positive definite weighting matrix [18]. $\mathbf{\Lambda}(n)$ is often the inverse of the measurement error covariance matrix. Measurements with less noise provide a greater contribution to the estimate than those with more noise, minimizing the expected mean-square error. In the iterative computation of the transform frequencies, the weighting matrix discounts older measurements.

For uniformly spaced time samples $t_i = i/f_s$, $i = 0, 1, \ldots, N-1$ with $\mathbf{\Lambda}(n) = \mathbf{I}$ and $n = N$, Eq. 8 becomes the well-known forward DFT,

$$\mathbf{w}(n) = \mathbf{B}'(n)\mathbf{r}(n) \tag{10}$$

because the columns of $\mathbf{B}(n)$ are now also orthogonal in addition to the rows, and Eq. 4 becomes the inverse discrete Fourier transform (IDFT).

Adding the next time sample, $r(n+1) = r(t_{n+1})$, Eq. 8 becomes

$$\begin{aligned} &\mathbf{w}(n+1) \\ &= N\left([\mathbf{B}'(n)\mathbf{b}^*(n+1)]\mathbf{\Lambda}(n+1)\begin{bmatrix}\mathbf{B}(n)\\ \mathbf{B}^T(n+1)\end{bmatrix}\right)^{-1} \\ &\quad \times [\mathbf{B}'(n)\mathbf{b}^*(n+1)]\mathbf{\Lambda}(n+1)\begin{bmatrix}\mathbf{r}(n)\\ r(n+1)\end{bmatrix} \end{aligned} \tag{11}$$

where, for example, a weighting matrix such as

$$\mathbf{\Lambda}(n+1) = \begin{bmatrix}\lambda\mathbf{\Lambda}(n) & \underline{0}\\ \underline{0} & 1\end{bmatrix} \tag{12}$$

provides filtering to fade older samples relative to newer ones, where $0 < \lambda < 1$.

Algebraic manipulations of Eq. 11 and substitution from Eq. 8 yield the following recursive least squares update of the frequencies,

$$\begin{aligned} &\mathbf{w}_0 = (\text{any reasonable initial value, e.g. } 0) \\ &\mathbf{P}_0^{-1} = \mathbf{I}(\text{initial value} = \text{identity matrix}) \\ &\mathbf{K}(n+1) = \frac{\mathbf{I}}{\lambda} - \frac{\mathbf{P}^{-1}(n)\mathbf{b}^*(n+1)\mathbf{b}(n+1)^T}{\lambda^2 + \lambda\mathbf{b}^T(n+1)\mathbf{P}^{-1}(n)\mathbf{b}^*(n+1)} \\ &\mathbf{P}^{-1}(n+1) = \mathbf{K}(n+1)\mathbf{P}^{-1}(n) \\ &\mathbf{w}(n+1) = \lambda\mathbf{K}(n+1)\mathbf{w}(n) + N\mathbf{P}^{-1}(n+1)\mathbf{b}^*(n+1)r(n+1) \\ &\mathbf{w}(n+1) = \mathbf{w}(n) + \Big(Nr(n+1)\mathbf{P}^{-1}(n+1) \\ &\qquad - \frac{\mathbf{b}(n+1)^T\mathbf{w}(n)\mathbf{P}^{-1}(n)}{\lambda + \mathbf{b}^T(n+1)\mathbf{P}^{-1}(n)\mathbf{b}^*(n+1)}\Big)\mathbf{b}^*(n+1) \end{aligned} \tag{13}$$

Matrix by matrix multiplications imply that this iteration has complexity of order N^3. However, modifications described next eliminate most of these multiplications.

Using mathematical induction, one can show that $\mathbf{b}^*(i)$ is always an eigenvector of $\mathbf{P}^{-1}(n)$. The induction step yields the following iterative update for these eigenvalues,

$$\epsilon_k(n+1) = \begin{cases} \frac{\epsilon_k(n)}{\lambda} & \text{when } \mathbf{b}^T(n+1)\mathbf{b}^*(i) = 0 \\ \frac{\epsilon_k(n)}{\lambda + N\epsilon_k(n)} & \text{when } \mathbf{b}^T(n+1)\mathbf{b}^*(i) = N \end{cases} \tag{14}$$

where $k = t_i f_s \text{mod}(N)$.

Using this eigenvalue iteration, we eliminate explicit matrix arithmetic with $\mathbf{P}^{-1}(n)$ and the iterative update of $\mathbf{P}^{-1}(n)$ in Eq. 13. The recursive least squares method becomes

$$\begin{aligned} &\mathbf{w}(0) = (\text{any reasonable initial value, e.g. } 0) \\ &\epsilon_i(0) = 1 \text{ for } i = 0, 1, \ldots, N-1 \\ &\epsilon_i(n+1) = \frac{\epsilon_i(n)}{\lambda} \text{ for all } i \neq t_{n+1} f_s \text{mod}(N) \\ &\epsilon_k(n+1) = \frac{\epsilon_k(n)}{\lambda + N\epsilon_k(n)} \text{ for } k = t_{n+1} f_s \text{mod}(N) \\ &\mathbf{w}(n+1) = \mathbf{w}(n) + \epsilon_k(n+1) \\ &\quad N\big(r(n+1) - \mathbf{b}^T(n+1)\mathbf{w}(n)/N\big)\mathbf{b}^*(n+1) \end{aligned} \tag{15}$$

Iterative updating of $\mathbf{P}^{-1}(n)$ becomes iterative updating of its eigenvalues, which is very efficient because it involves updating scalars only. This iteration includes a single inner product of vectors of length N and scale/add of two vectors of length N, so the iterative update now has complexity of order N.

Because $\mathbf{b}^T(n+1)\mathbf{w}(n)/N$ is an inverse DFT calculated at one time point, the term $r(n+1) - \mathbf{b}^T(n+1)\mathbf{w}(n)/N$ in Eq. 15 is a prediction error that measures how well an inverse DFT evaluated at time $k = t_{n+1} f_s \text{mod}(N)$ predicts the latest input datum $r(n+1)$ using the present estimate

for the coefficients $\mathbf{w}(n)$. The update to $\mathbf{w}(n)$ is proportional to this error. The product of $\varepsilon_k(n)$ and the prediction error is a scalar weighting factor for $\mathbf{b}^*(n+1)$ that additively updates $\mathbf{w}(n)$. If the inverse DFT of $\mathbf{w}(n)$ predicts the new data sample well, or $\varepsilon_k(n)$ is small, then $\mathbf{w}(n)$ changes little.

The forgetting factor λ determines whether or not the eigenvalues ($\varepsilon_i(n)$ and $\varepsilon_k(n)$) in Eq. 15 grow without bound or converge to 0. If $\lambda = 1$, i.e. no forgetting of old data, each $\varepsilon_i(n)$ stays constant, while $\varepsilon_k(n+1) < \varepsilon_k(n) < 1/N$ for all values of $\varepsilon_k(n)$. If all values for k are equally likely, then all eigenvalues will eventually converge to 0. Because the estimate of $\mathbf{w}(n)$ weights all data samples equally and each new data sample becomes a smaller fraction of all data samples, it makes sense that the prediction error weight $\varepsilon_k(n+1)$ continually shrinks.

However, we would like the estimator to discount old data in order to adapt to quasi-stationary conditions when used for PSD estimation, so in general, $0 < \lambda < 1$. In this case, each $\varepsilon_i(n)$ in Eq. 15 grows larger with each iteration, while $\varepsilon_k(n)$ shrinks. If the forgetting factor is very small, $\varepsilon_k(n)$ will in general be large and $\epsilon_k(n+1) \approx 1/N$. This implies that the exact value of the forgetting factor is not important, and that we could assume $\varepsilon_k(n+1) = 1/N$ and remarkably remove the eigenvalue updates completely. The fast update method becomes

$$\begin{aligned}\mathbf{w}(0) &= \text{(any reasonable initial value, e.g. 0)}\\ \mathbf{w}(n+1) &= \mathbf{w}(n) + \left(r(n+1) - \mathbf{b}^T(n+1)\mathbf{w}(n)/N\right)\\ &\quad \times \mathbf{b}^*(n+1)\end{aligned} \tag{16}$$

Figure 1 shows a block diagram for this method, where equivalently the iteration is shown from $n-1$ to n. Because $\mathbf{b}^T(n)$ is always a selection from the same finite set of complex sinusoidal values, computational overhead can be reduced further by storing pre-computed values in memory. Because these sinusoids are the same as used in fast and discrete Fourier transforms, the same well-known methods that exploit sinusoidal function symmetries for storage efficiency will also work here (e.g. all values can be obtained trivially from a quarter cycle of values with a frequency resolution of $2\pi/N$). Each iterative update retrieves the values it needs from memory.

In the case of uniform sampling, the Fourier coefficients computed by this method converge to the DFT after N iteration steps. Assume uniform sampling for N contiguous samples spaced $1/f_s$ seconds apart, in which case $k = t_n f_s \bmod(N) = 0,\ 1,\ 2,\ 3, \ldots, N-1$. Let $\mathbf{w}_A$ represent the DFT of these N samples and $\mathbf{e}(n) = \mathbf{w}(n) - \mathbf{w}_A$ represent the error between the present estimate of the frequency coefficients and the DFT. Subtracting $\mathbf{w}_A$ from both sides of Eq. 16 and using the fact that $r(n+1) = \mathbf{b}^T(n+1)\mathbf{w}_A/N$ results in the following coefficient error expression,

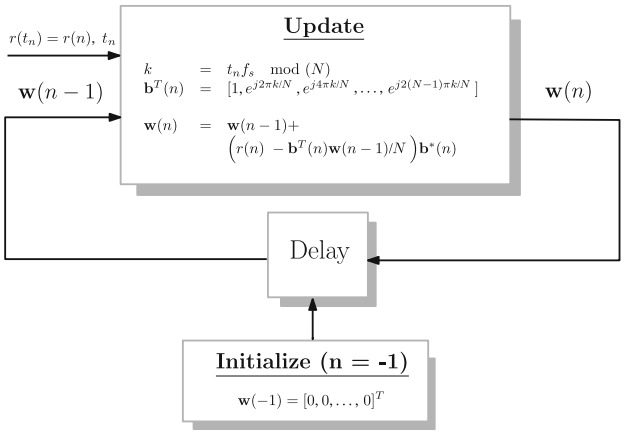

Fig. 1 Signal flow for the fast recursive least squares method of Eq. 16 shows an update of the present spectrum coefficients estimate $\mathbf{w}(n-1)$ by using a "prediction error", i.e. the difference between input data $r(t_n)$ and estimate of it using an inverse DFT of $\mathbf{w}(n-1)$. $r(t_n)$ is sampled nonuniform in time, but is measured with a sample time quantization of $1/f_s$. Think of f_s as the frequency of a digital counter whose count measures the time of each data sample. N is the number of equally spaced frequencies within the frequency band of 0 to $2\ \pi/f_s$

$$\begin{aligned}\mathbf{e}(n+1) &= \mathbf{e}(n) - \left(\mathbf{b}^T(n+1)\mathbf{e}(n)/N\right)\mathbf{b}^*(n+1)\\ \mathbf{e}(n+1) &= \left(\mathbf{I} - \frac{\mathbf{b}^*(n+1)\mathbf{b}^T(n+1)}{N}\right)\mathbf{e}(n)\end{aligned} \tag{17}$$

or

$$\mathbf{e}(n+1) = \prod_{i=0}^{n}\left(\mathbf{I} - \frac{\mathbf{b}^*(i+1)\mathbf{b}^T(i+1)}{N}\right)\mathbf{e}(0) \tag{18}$$

Then

$$\begin{aligned}\mathbf{e}(N) &= \prod_{i=0}^{N-1}\left(\mathbf{I} - \frac{\mathbf{b}^*(i+1)\mathbf{b}^T(i+1)}{N}\right)\mathbf{e}(0)\\ &= \left(\mathbf{I} - \frac{\mathbf{b}^*(1)\mathbf{b}^T(1)}{N}\right)\left(\mathbf{I} - \frac{\mathbf{b}^*(2)\mathbf{b}^T(2)}{N}\right)\\ &\quad \left(\mathbf{I} - \frac{\mathbf{b}^*(N)\mathbf{b}^T(N)}{N}\right)\mathbf{e}(0)\\ &= \left(\mathbf{I} - \frac{\mathbf{b}^*(1)\mathbf{b}^T(1)}{N} - \frac{\mathbf{b}^*(2)\mathbf{b}^T(2)}{N} - \frac{\mathbf{b}^*(N)\mathbf{b}^T(N)}{N}\right)\mathbf{e}(0)\end{aligned} \tag{19}$$

where the last equality follows from the fact that $\mathbf{b}(1)$, $\mathbf{b}(2)$,..., $\mathbf{b}(N)$ are orthogonal in the case of uniform sampling. Further simplification leads to

$$\begin{aligned}\mathbf{e}(N) &= \left(\mathbf{I} - \frac{\mathbf{B}'(N)\mathbf{B}(N)}{N}\right)\mathbf{e}(0)\\ &= (\mathbf{I} - \mathbf{I})\mathbf{e}(0) = 0\end{aligned} \tag{20}$$

where $\mathbf{B}'(N)$ is the DFT matrix of Eq. 10 that is well known to have orthogonal rows and columns. Therefore the error

is 0 after N iterations, no matter what the starting values are for the coefficients.

A priori information about $\mathbf{w}(n)$ may further reduce the amount of computation implied by Eq. 16. If we know that certain frequency bands in $r(n)$ are 0, for example, we can remove those frequencies from $\mathbf{b}(n)$ and the corresponding coefficients from $\mathbf{w}(n)$. N is the number of remaining non-zero frequencies or the remaining length of $\mathbf{w}(n)$. Section 3.1 shows an example resulting in significant computational savings.

3 Performance assessment

We assessed the performance of RFT based spectrum estimation by comparison with LST spectrum estimation. Because the true nature of field collected physiological signals is unknown (e.g. which part is true signal versus noise, what is the true interpretation, etc.), we primarily used simulated signals whose characteristics are known. However, we also illustrated usage of the method on real, non-uniformly heart rate signals. First, we present two illustrative spectrum estimation examples using simulated signals with known spectra. Next, we do a detailed comparison for estimation of the HRV spectrum using Monte Carlo methods to simulate quasi-stationary signals with various levels of randomly missing beats. Finally, we illustrate performance on real signals.

3.1 Spectrum estimation examples

The first example provides an illustration of spectrum estimation performance in a situation where the time interval between samples varies randomly as multiple of the basic time quantization, $1/f_s$. We used the following sinusoidal signal,

$$r(t) = \sum_{i=-(M-1)/2}^{(M-1)/2} a_i e^{j2\pi f_i t} + v(t) \tag{21}$$

with $M + 1$ frequencies, f_i, and corresponding amplitudes, a_i, which may be complex numbers. $v(t) \sim N(0,\ \sigma^2)$ is a normally distributed, zero mean additive random noise.

In our particular simulation, we chose $a_1 = a_{-1} = a_2 = a_{-2} = a_3 = a_{-3} = a_4 = a_{-4} = 1$, and $f_1 = 10$, $f_{-1} = -10$ $f_2 = 20$, $f_{-2} = -20$, $f_3 = 100$, $f_{-3} = 100$, $f_4 = 200$, $f_{-4} = 200$ Hz, and SD of $v(t)$ is $\sigma = 0.01$. All other parameters were set equal to zero. The sampling frequency $f_s = 512$, and we sample the signal at discrete times

$$t_n = \frac{3n + \lfloor z + 0.5 \rfloor}{f_s} \tag{22}$$

where z is a random number uniformly distributed between -1.5 and 1.5. The offset of 0.5 causes the "floor function",

$$\lfloor z \rfloor = \max\{n \in \mathbb{Z} \mid n \leq z\} \tag{23}$$

to round (vs. truncate). The sample time interval varied randomly from 1 to 5 times the basic time quantization, $1/f_s = 1/512$ s. This signal has two closely spaced low frequency peaks along with two additional more widely spaced higher frequency peaks. The 200 Hz peak is close to the 256 Hz Nyquist frequency implied by a sampling time quantization of 1/512 s.

The second example illustrates how to exploit a priori knowledge of $\mathbf{w}(n)$ in order to avoid significant amounts of computation. For example, if it is known that a certain portion of the coefficients $\mathbf{w}(n)$ are zero, they are removed from the vector $\mathbf{w}(n)$ in Eq. 16 (and hence are not stored or updated), and the corresponding components are removed from the complex exponential vectors.

A simple model for the "instantaneous heart rate" is

$$h(\mathrm{t}) = \mathrm{h_m} + \mathrm{a_l}\cos(2\pi \mathrm{f_l t}) + \mathrm{a_h}\cos(2\pi \mathrm{f_h t}) + \mathrm{v(t)} \tag{24}$$

which is the heart rate equal to the reciprocal of the time interval between successive heart beats. This time interval varies from beat to beat. Excluding the DC mean, two frequency peaks roughly model the spectral characteristics of the heart rate, as discussed in more detail in Sect. 3.2. We chose a sampling rate of $f_s = 500$, and $h_m = 60$ to be the mean number of beats per minute (bpm). Let $a_l = 2$ bpm, $a_h = 2.5$ bpm, $f_l = 0.095$ Hz, $f_h = 0.275$ Hz, and the SD of $v(t)$ be $\sigma = 0.2$. The time interval between heart beats is $r(t) = 1/h(t)$, and the sequence of time points for each heart beat is given by,

$$\begin{aligned} t_0 &= 0 \\ t_{n+1} &= t_n + \lfloor r(t_n) f_s + 0.5 \rfloor / f_s \quad \text{for } n = 0, 1, 2, \ldots, \end{aligned} \tag{25}$$

In this scenario, the time quantization $1/f_s$ is small and there are many hundreds of quantization intervals between samples. Each beat arrives about once every second, but the signal time quantization is 1/500 s controlled typically by the original electrocardiogram sampling rate of 500 samples/s.

A time resolution or quantization of $1/f_s = 1/500$ implies a very large number of coefficients, N, to resolve the low frequencies in this example. To achieve frequency resolution $f_s/N = 0.01$ Hz, for example, Eq. 16 would update $N = 50{,}000$ frequency coefficients, where most of these coefficients are known a priori to be 0. Instead of updating coefficients over the entire bandwidth, we exploit this a priori information by including in $\mathbf{w}(n)$ only those frequency coefficients for which there may be non-zero values. In this particular example, we updated only $N = 512$ frequencies that are equally spaced in the bands 0 to $f_b = 1$ Hz and 255–256 Hz (or equivalently via aliasing to $-f_b = -1$ to 0 Hz). We computed only 1/256 of the full Nyquist bandwidth implied by

f_s = 512, achieving an equivalent reduction in the amount of computation.

3.2 Heart rate variability spectrum estimation

This experiment illustrates how well the RFT performs in a well-known application where signal sample time intervals vary widely and missing signal samples are common. The autonomic nervous and endocrine systems regulate the cardiac cycle causing beat to beat variation in the heart rate. Therefore, it is natural to investigate HRV for assessing organ system and biological feedback interactions. In addition, there are some clinical studies that indicate HRV measurements are useful guidance for trauma triage [5, 6] and may be predictive of life threatening arrhythmias [20].

HRV metrics measure time and frequency domain characteristics of the variation between successive heart beats. Table 1 lists the most common frequency domain HRV measures [1]. The low frequency (LF), high frequency (HF) and LF/HF ratio metrics are of particular interest in our experiment, where the HRV spectrum is of the instantaneous heart rate, $h(n)$, in bpm. If the time units are in seconds, s, for example, the power spectrum units are s^2/Hz and total power in a frequency band is in units of s^2. One could also estimate the spectrum for the RR interval, $r(n) = 60/h(n)$ using exactly the same methods.

The study of Clifford and Tarassenko [4] concluded that the LST provides more accurate LF/HF estimates for RR interval data with ectopic beats than do FFT methods that first interpolate samples to uniform spacings. In order to compare our RFT estimates with results from this study, we adopted some of the same signal models, signal configurations, and experimental methods. This provides an independent verification of our LST implementation and also indirectly allows us to compare the RFT to FFT methods.

In particular, the heart rate model is that of Eq. 24, sampled using Eq. 25. We selected the model parameters to match those of the study [4] and are $f_s = 1{,}000$, and $h_m =$ 60 bpm, a_l = 2 bpm, $a_h = 2.5$ bpm. The mean value for low frequency and high frequency parameters are f_l = 0.095 Hz and f_h = 0.275 Hz respectively. In addition, f_l and f_h change during the simulation to model non-stationarity, incrementing as follows,

$$\begin{aligned} f_l(n) &= 0.077 + 0.00056n \quad \text{for } 0 \le n < 66 \\ f_h(n) &= 0.233 + 0.00130n \quad \text{for } 0 \le n < 66 \end{aligned} \tag{26}$$

Index n cycles up from 0 to 65 and then back down to 0, but not regularly. For each frequency, f_l and f_h, the following Gaussian functions $S_l(f_l)$ and $S_h(f_h)$ respectively determine how many samples to generate at a particular frequency value before incrementing (decrementing) n to specify the next frequency value.

$$\begin{aligned} S_l(f_l) &= \lfloor 8\, e^{-(f_l - 0.095)^2/0.0002} \rfloor \\ S_h(f_h) &= \lfloor 8\, e^{-(f_h - 0.275)^2/0.0010} \rfloor \end{aligned} \tag{27}$$

Figure 2 approximately replicates Fig. 1a in Clifford and Tarassenko [4] indicating that we achieve a comparable model of non-stationarity. The max of 65 for n leads to approximately 300 beats having this Gaussian distribution and hence a theoretical LF/HF power ratio of 2/2.5 = 0.64.

We used this quasi-stationary synthetic signal model to generate 300 heart beats, minus random beats simulated missing due to abnormal ectopic depolarization or noise, and ran 1,000 simulations for one randomly missing beat out of the 300 beats, 1,000 simulations for two randomly missing beats out of the 300 beats, 1,000 simulations for three randomly missing beats, continuing for up to 30 randomly missing beats. The results are reported using a statistical box plot, where each box marks the middle 50% of the 1,000 estimates made for each LF/HF value at a particular number of simulated missing beats (1–30). The line in the box is the median. Vertical whiskers indicate maximum and minimum data values, except for outlier data values marked with plus signs.

To illustrate behavior on real heart rate data, we computed the spectrum of ECG and ECG RR interval data obtained from "Physionet" (http://www.physionet.org). In particular, we computed the spectrum of ECG record 100 in the "MIT-BIH Arrhythmia" database. This is a 30-min record, where the original ECG signal from which RR interval data was extracted had a sample rate of 360 Hz. We also computed a time–frequency plane for long-term ECG RR interval record N1 in the "Exaggerated heart rate oscillations during two meditation techniques" database. The original ECG sample rate for this signal was 128 Hz.

Table 1 Typical HRV frequency domain measures

Spectral Power	Abbrev.	Frequency range (Hz)
All frequencies	Total HRV	0.000–0.500
Ultra-low frequency	ULF	0.000–0.003
Very low frequency	VLF	0.003–0.040
Low frequency	LF	0.040–0.150
High frequency	HF	0.150–0.400
Very high frequency	VHF	0.400–0.500
Low/high frequency ratio	LF/HF	NA

4 Results

Figure 3 compares spectrum estimates of the LST of Eq. 1 with the RFT of Eq. 16 for the first example signal of

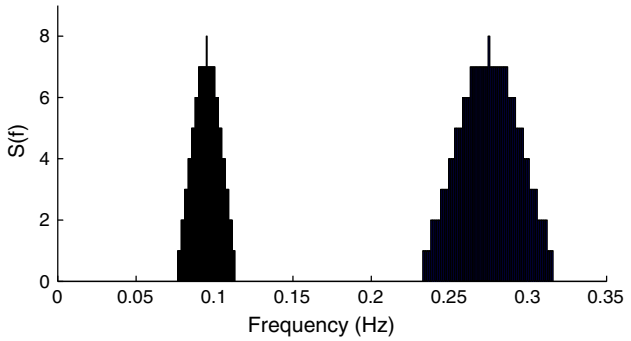

Fig. 2 In order to model non-stationarity in the instantaneous heart rate signal model of Eq. 24, the low frequency peak f_l and high frequency peak f_h increment and then decrement around a mean. $S_l(f_l)$ and $S_h(f_h)$ are integer valued Gaussian functions (see Eq. 27] that respectively determine how many samples to generate at each frequency value before incrementing (decrementing) the frequency to the next value

Eq. 21 with four sinusoidal frequencies and a sample time interval that varies randomly from 1 to 5 times the basic time quantization of 1/512 s. In this example case, both estimators show all four frequency peaks. However, the RFT is more consistent for this stationary signal example in the sense that signal to noise ratio improves as more data samples are used in computing the estimate.

Figure 4 compares spectrum estimates of the LST of Eq. 1 with the RFT of Eq. 16 for the signal of Eq. 24 where 99.6% of the frequency coefficients in the full signal band ($f_s/2$) are known to be 0 and therefore not stored or updated. Both the RFT and LST estimators perform comparably in having very similar signal peak to noise floor ratios. The estimates of both methods do not appear to improve when using more data samples.

Figure 5 shows the results of the Monte Carlo HRV simulation experiment described in Sect. 2. As in Clifford and Tarassenko [4], the LST method produces accurate estimates with mean value within 3–6% of the true value and with a few additinoal percent variance increasing as missing beats increase. The RFT method performs equally well as the LST (the RFT estimate has slightly less bias in this particular experiment), and hence by inference from results in Clifford and Tarassenko [4], performs better than FFT methods that first interpolate beats to create uniformly sampled data.

Figure 6 shows a power spectrum of ECG record 100 in the "MIT-BIH Arrhythmia" database from "Physionet" (http://www.physionet.org). The Physionet "RR Intervals, Heart Rate, and HRV Howto"uses this record to illustrate HRV spectrum computation, and compares the LST based spectrum with FFT and maximum entry (MEM) based methods on this record. The sharp peaks at 0.167, 0.28, and 0.42 Hz are artifacts from the analog tape recorder and playback used to record the data. The Physionet LST and FFT plots better distinguished this artifact of non-physiological origin than MEM, and the Physionet LST was the only method that clearly showed the respiratory sinus arrythmia peak. Both our LST and RFT spectrum plots have the same features and qualitative character as the Physionet LST plot.

Figure 7 shows a time frequency plane computation for long-term ECG record N1 of normal breathing taken from Physionet's "Exaggerated heart rate oscillations during two meditation techniques" database. We compute the power spectrum over a moving window of 512 beat samples to compute the spectrum at each new heart beat (i.e. we use the present and last 511 beats). The respiratory sinus arrhythmia peak around 0.3 Hz is visible in these plots. The original ECG signal from which RR interval data was extracted was sampled at 128 samples/s. A higher sample rate is more desirable for the RFT for better time resolution.

5 Discussion

We described a new method for calculating the DFT iteratively and our results demonstrate that the proposed method works for discrete time signals whose sample time intervals may be widely nonuniform. Specifically, we showed equivalent performance with Lomb–Scargle periodogram methods.

One advantage of the RFT is that it enables high quality spectrum estimation implementations in many embedded applications where other methods would be impractical or would not work nearly as well. As an example, suppose one desires to estimate the HRV spectrum, but the time series is missing a few heart beats in a sequence of RR intervals that are naturally nonuniform spaced, as noted above. Typically a running 5 min of data or longer is the spectral window for which HRV frequencies are estimated. In these situations it is best to display no answer for the period of missing beats rather than "manufacture" beats, but this might mean no answer for 5 min until a full set of beats are obtained. The LST and the RFT methods work despite missing beats because they bypass the necessity for re-sampling or interpolation. For similar computational performance to RFT of Eq. 16, the LST must either buffer a minute or two of data (a beat only comes about every second), causing a minute or so of time delay before a spectrum can be computed, or for typical configuration take almost two orders of magnitude more CPU bandwidth than the RFT to compute an estimate on every beat.

In many embedded instruments, a single CPU is typically asked to handle many functions, and the arithmetic ones tend to use the bulk of its computational bandwidth—e.g. multiple filters applied to multiple signal channels,

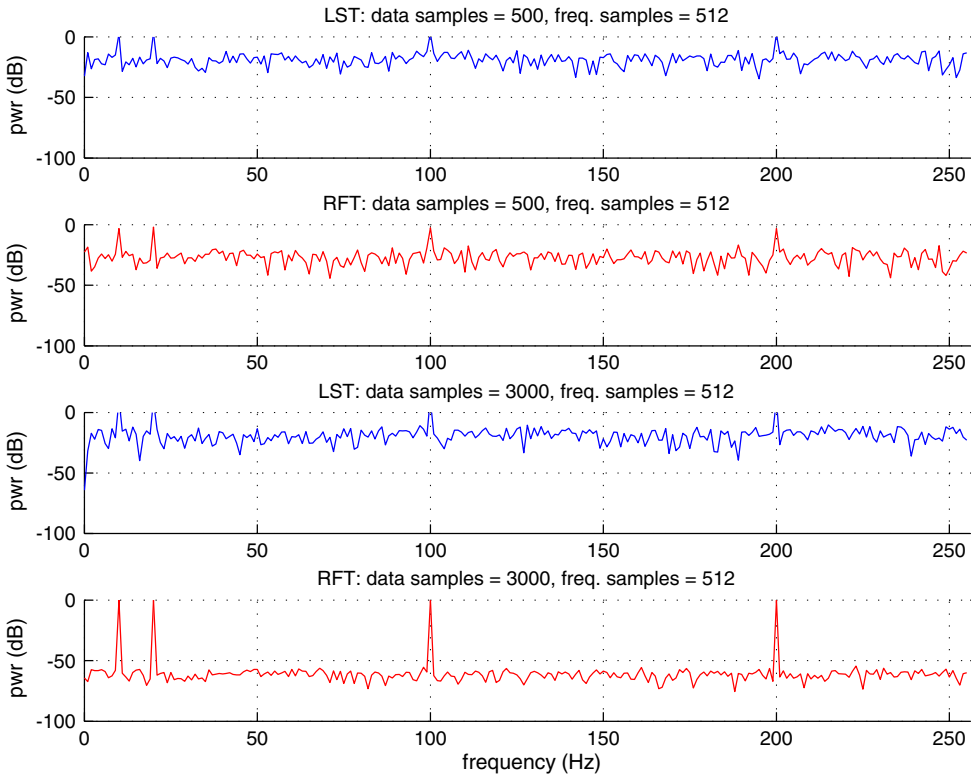

Fig. 3 Power spectrum estimate comparing the LST of Eq. 1 with the RFT of Eq. 16 at 512 equally spaced frequencies for a synthetic signal having four cosine frequencies and 1% noise. The sample time interval varies randomly from 1 to 5 times the basic time quantization of 1/512 s

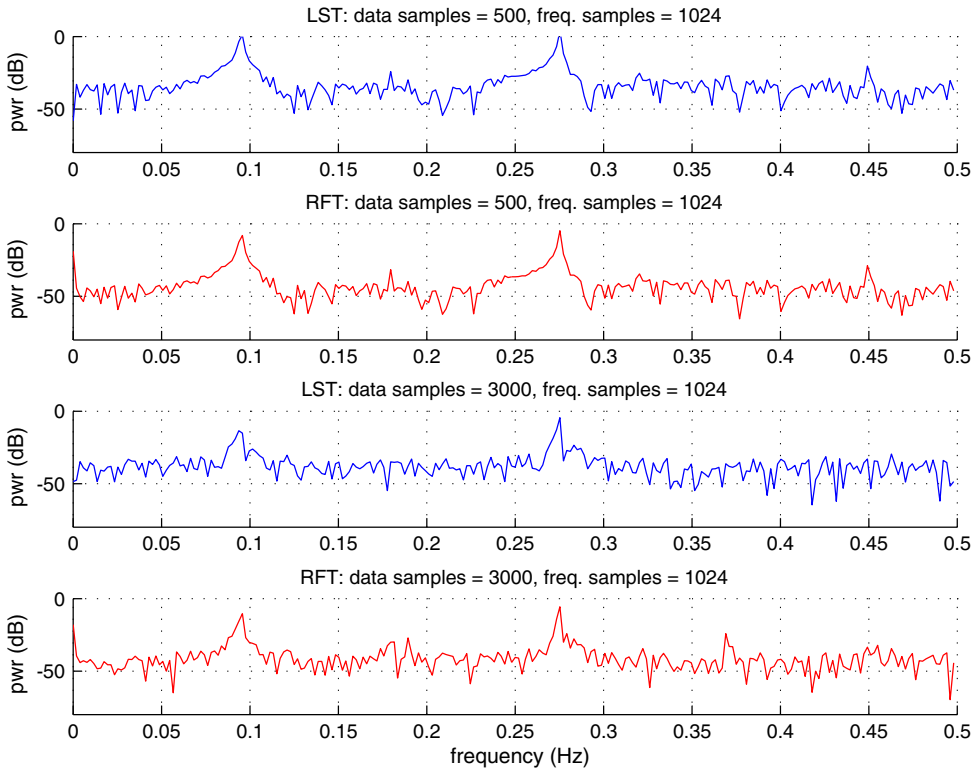

Fig. 4 These power spectrum estimates compare LST of Eq. 1 with the RFT of Eq. 16. The input is a synthetic signal with two sinusoids emulating the variability in heart rate and having 1% noise. The estimate is of 512 equally spaced frequencies over the frequency band 0 to 1 Hz (a small fraction of the full 0–256 Hz band implied by the time quantization of $1/f_s = 1/512$). Because samples are spaced on average 1-s apart, these estimates cover approximately 8.5 and 17-min time windows

signal feature detection algorithms, pattern recognition, etc. For typical practical configurations as described above, the Fast LST method from the classic "Numerical Recipes" book [13] requires more than $10 \times \log(N)$ times as many mulitply/adds for each spectrum update than does the RFT of Eq. 16. For a 512 point spectrum this is about

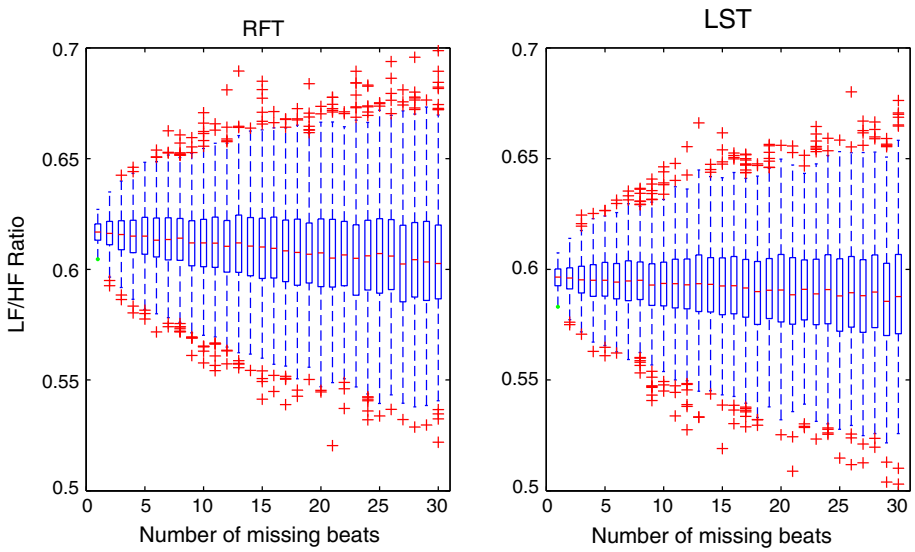

Fig. 5 These statistical box plots compare LST and RFT estimates of the ratio of heart rate low frequency power (LF) power to high frequency (HF) power as a function of number of randomly missing beats in the input data. Each estimation uses 300 consecutive heart beat intervals minus random beats simulated missing due to abnormal ectopic depolarization or noise. The input is a synthetic heart rate signal with non-stationary LF and HF peaks in its spectrum (see Eqs. 26, 27]. The LF/HF power ratio is a known constant 0.64. Each *box* marks the middle 50% of the 1,000 estimates made for each LF/HF value. The *line* in the box is the median. *Vertical whiskers* indicate maximum and minimum data values, except for outlier data values marked with plus signs

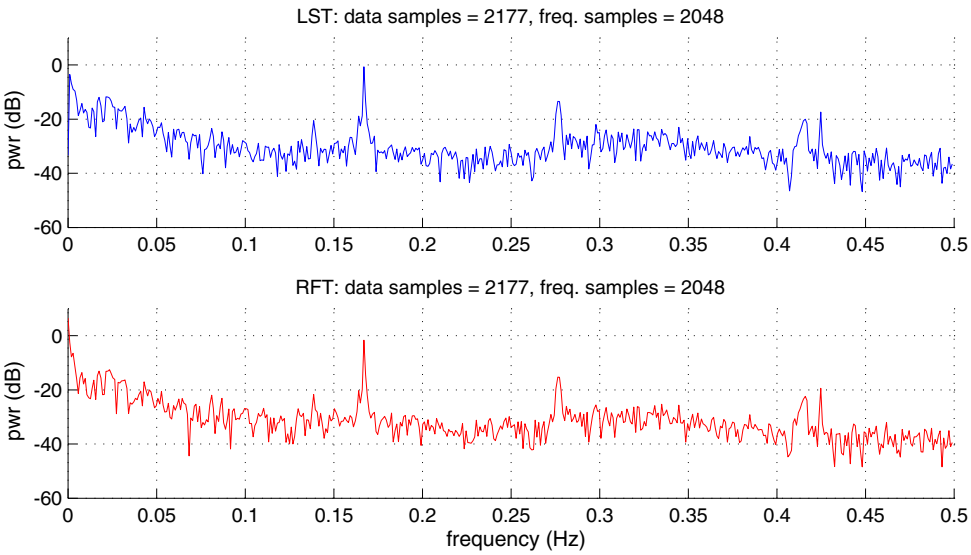

Fig. 6 These plots of ECG record 100 in the MIT-BIH Arrhythmia database from Physionet recreate the same features and qualitative character as the LST plot in the"RR Intervals, Heart Rate, and HRV Howto" at http://www.physionet.org. The sharp peaks at 0.167, 0.28, and 0.42 Hz are artifacts from the analog tape recorder and playback used to record the data

$10 \times \log_2(512) = 90$ times the CPU bandwidth used by the RFT, no matter what is the particular CPU or its multiplication efficiency. Overall, the ratios and scaling of trigonometric functions in the LST are also more complex to implement using fixed-point arithmetic, which is sometimes a more significant embedded computational issue than total arithmetic operations.

However, the Fast LST could in theory gain comparatively in computational efficiency by updating the spectrum 1/90 as often as the RFT method of Eq. 16, i.e. iteratively sliding the 5-min estimation time window 90 s forward and computing a new estimate, versus sliding it 1-s forward and computing a new estimate. The relative total computational efficiency between Eq. 16 and the Fast LST depends on the data time window used for spectrum updates and the amount of window overlap. However, the requirement to buffer a history of data to minimize computation using the Fast LST introduces an additional time delay, which is sometimes undesirable in biomedical contexts (e.g. QRS detection delay for cardioversion synchronization). For

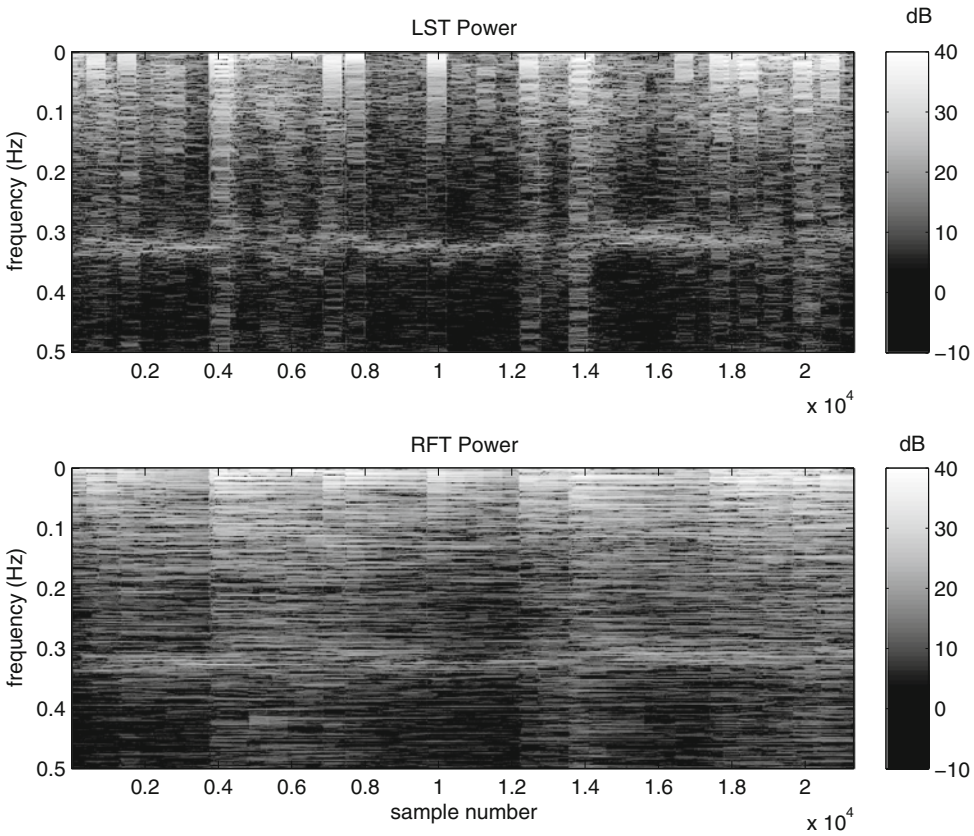

Fig. 7 This time frequency plane of long-term ECG record N1 of normal breathing taken from "Exaggerated heart rate oscillations during two meditation techniques" database at http://www.physionet.org shows a respiratory sinus arrhythmia peak around 0.3 Hz. The original ECG signal from which RR interval data was extracted was sampled at 128 samples/s. We used a moving window of 512 beat samples to compute the spectrum at each new heart beat sample (i.e. we use the present and last 511 beats)

HRV, waiting 90 beats for a calculation, means an estimate comes out every minute 30 s, compared to every second for the RFT method.

The RFT method of Eq. 16 works analogously for any discrete basis functions. In particular, the method only demands that the discrete basis functions satisfy Eq. 7, and therefore could be applied to estimate the transform coefficients in transform domains other than discrete Fourier, as in discrete Wavelets, for example.

6 Conclusion

In this paper, we have described a new method for calculating the DFT iteratively that works for discrete time signals whose sample time intervals may be widely non-uniform. The new method is a recursive least squares estimator that iteratively updates a spectrum estimate with each new sample, where each iteration has computational order N. We provided a mathematical derivation and showed equivalent performance with Lomb–Scargle periodogram methods. In particular, we provided illustrative performance examples on both synthetic and real data, and a detailed heart-rate variability estimation study aimed at assessing the proposed method.

References

1. Alexander S (1986) Fast adaptive filters: a geometrical approach 3(4):18–28
2. Cadzow J (1990) Signal processing via least squares error modeling 7(4):12–31. doi:10.1109/53.62941
3. Clifford G, Tarassenko L (2005) Quantifying errors in spectral estimates of hrv due to beat replacement and resampling. Biomed Eng IEEE Trans 52(4):630–638
4. Cooke WH, Salinas J, Convertino VA, Ludwig DA, Hinds D, Duke JH, Moore FA, Holcomb JB (2006) Heart rate variability and its association with mortality in prehospital trauma patients. J Trauma 60:363–370
5. Ellenby MS, McNames J, Lai S, McDonald BA, Krieger D, Sclabassi RJ, Goldstein B (2001) Uncoupling and recoupling of autonomic regulation of the heart beat in pediatric septic shock. Shock 16(4):274–377
6. Glentis GO, Berberidis K, Theodoridis S (1999) Efficient least squares adaptive algorithms for fir transversal filtering 16(4):13–41. doi:10.1109/79.774932
7. Hilton MF, Bates RA, Godfrey KR, Chappell MJ, Cayton RM, Gil E, Mantaras C, Aiolfi S, Cerutti S (1999) Evaluation of frequency and time–frequency spectral analysis of heart rate variability as a diagnostic marker of the sleep apnoea syndrome. Med Biol Eng Comput 37(1):760–769
8. HRV (1996) Task Force of the European Society of Cardiology and the North American Society of Pacing & Electrophysiology: Heart rate variability standards of measurement, physiological interpretation, and clinical use. Eur Heart J 17:354–381
9. Laguna P, Moody G, Mark R (1998) Power spectral density of unevenly sampled data by least-square analysis: performance and

application to heart rate signals. IEEE Trans Biomed Eng 45(6):698–715
10. Lomb N (1976) Least-squares frequency analysis of unequally spaced data. Astrophys Space Sci 39:447–462
11. Mendez MO, Bianchi AM, Montano N, Patruno V, Gil E, Mantaras C, Aiolfi S, Cerutti S (2008) On arousal from sleep: time–frequency analysis. Med Biol Eng Comput 46(4):341–351
12. Plemmons R (1993) Fft-based rls in signal processing. In: Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing ICASSP-93, vol 3, pp 571–574. doi:10.1109/ICASSP.1993.31956210.1109/ICASSP.1993.319562
13. Press WH, Teukolsky SA, Vetterling WT, Flannery BP (2002) Numerical Recipes in C++: the art of scientific computing. Cambridge University Press, Cambridge
14. Rajendra Acharya U, Paul Joseph K, Kannathal N, Lim CM, Suri JS (2006) Heart rate variability: a review. Med Biol Eng Comput 44(12):1031–1051. http://dx.doi.org/10.1007/s11517-006-0119-0
15. Sayed AH, Kailath T (1994) A state-space approach to adaptive RLS filtering. Signal Process Mag IEEE 11:18–60, 3 July. doi:10.1109/79.295229. ISSN: 1053-5888
16. Scargle JD (1982) Studies in astronomical time series analysis ii, statistical aspects of spectral analysis of unevenly spaced data. Astrophy J 263:835–853
17. Schwab K, Eiselt M, Putsche P, Helbig M, Witte H (2006) Time-variant parametric estimation of transient quadratic phase couplings between heart rate components in healthy neonates. Med Biol Eng Comput 44(12):1077–1083
18. Strang G (1986) Introduction to applied mathematics. Wellesley-Cambridge Press
19. Thong T (2006) Real-time evaluation of spectral heart rate variability. In: Proceedings of the IEEE EMBS annual international conference (EMBS'06), New York City, USA
20. Thong T, Raitt MH (2005) Ventricular tachyarrhythmia prediction. In: Proceedings of the 2005 IEEE EMBS 27th annual conference
21. Thong T, McNames J, Aboy M (2004) Lomb–Welch periodogram for non-uniform sampling. In: Proceedings of the 26th annual international conference of the IEEE EMBS, pp 271–274