THE JOURNAL OF NAVIGATION (2015), **68**, 989–998. © The Royal Institute of Navigation 2015
doi:10.1017/S0373463315000284

A Discrete Fourier Transformation-based Method for Phase Delay Estimation in X-ray Pulsar Navigation

Haoye Lin and Bo Xu

(*School of Astronomy and Space Science, Nanjing University, Nanjing 210023, China*)
(E-mail: xubo@nju.edu.cn)

X-ray pulsar navigation is a promising technology for autonomous spacecraft navigation. The key measurement of pulsar navigation is the time delay (phase delay). There are various methods to estimate phase delay, but most of them have high computational complexities. In this paper, a new method for phase delay estimation is proposed, which is based on the time-shift property of Discrete Fourier Transformation (DFT). With this method, the time complexity can be greatly reduced. Also, a delta-function approximation can be used to further decrease the computational cost. Numerical simulation shows that the proposed method is effective for phase delay estimation, and the reduced complexity makes our method more suitable for on board implementation.

KEY WORDS

1. Pulsar navigation. 2. Phase delay estimation. 3. DFT-based method. 4. Computational complexity

Submitted: 1 February 2015. Accepted: 15 March 2015. First published online: 13 April 2015.

1. INTRODUCTION. Pulsar navigation, proposed by Downs (1974), is an autonomous navigation technology that utilises the periodic signals from remote pulsars to perform self-positioning. With the characteristics of independence and wide availability, pulsar navigation is particularly suitable for the fields of deep-space exploration. Among various pulsar sources, the X-ray pulsar is of great potential as small-sized detectors can be employed for navigation.

The recent decade has seen many studies in X-ray pulsar navigation, including the pulsar database, phase delay estimation, and resolution of periodic ambiguity, etc. For example, Sala et al. (2004) analysed the feasibility of pulsar navigation in a report for the European Space Agency (ESA). They introduced a Maximum Likelihood method to estimate the time delay, which provides unbiased minimum variance estimations with mean square error approaching the Cramer-Rao Lower Bound. Sheikh and Pines (2006) proposed a new approach that provides corrections to estimated spacecraft position and achieved 200 m or less errors of positioning in numerical simulation. In Emadzadeh et al. (2009), as well as Emadzadeh and Speyer (2010), epoch folding was put forward to obtain the empirical rate function and a non-linear least-squares estimator was proposed to evaluate the time delay, which has less complexity than the Maximum Likelihood estimator. Later, in Emadzadeh and Speyer (2011a), a cross-correlation method for time delay estimation was proposed and its performance against the Cramer-Rao lower bound was analysed.

The key measurement of X-ray pulsar navigation, whether used for absolute positioning or relative positioning, is the time delay, which is obtained from the phase delay of the pulsar signal. Generally, there are three methods to estimate the phase delay: cross-correlation estimator, non-linear least squares estimator and maximum-likelihood estimator (Emadzadeh and Speyer, 2011b). A main problem of these estimators lies in the requirement to optimise non-convex functions, which will introduce a heavy computational burden. Considering that the process of autonomous navigation is realised by the computers on the spacecraft, a low-computational-complexity navigation algorithm is required for on board implementation.

In this paper, a new method for phase delay estimation is proposed, which is based on the time-shift property of a Discrete Fourier Transformation (DFT). The main advantage of this method is that the computational complexity of phase delay estimation can be greatly reduced, and with the utilisation of delta-function approximation, the computational cost as well as the memory cost can be further decreased.

The remainder of the paper is organised as follows: in Section 2, the process of epoch folding is reviewed and the computational complexities of the existing estimators are analysed. Then a DFT-based method for phase delay estimation is proposed in Section 3. Section 4 presents a numerical simulation to compare the performance of the proposed method and the cross-correlation estimator. Finally, a conclusion of the paper is given in Section 5.

2. A REVIEW OF EPOCH FOLDING AND ESTIMATORS

2.1. *Fundamentals of pulsar navigation.* The fundamentals of pulsar navigation can be expressed by the following equation

$$\boldsymbol{r_{12}} \cdot \boldsymbol{n} = c \cdot (t_e - \Delta t) \tag{1}$$

where $\boldsymbol{r_{12}}$ is the relative position vector between two spacecraft, $\boldsymbol{n}$ is the unit vector in the direction of the pulsar, c is the speed of light, t_e is the time delay of pulsar arrival between two spacecraft, and Δt is the clock error. In the navigation process, the value of time delay is obtained by estimating the phase delay of pulsar signals.

2.2. *Epoch folding procedure.* Epoch folding is usually the first step to estimate the phase delay. In this subsection, a brief review of the epoch folding procedure is given.

To describe the rate of photons arrival, an overall rate function $\lambda(t)$ is defined by (Golshan and Sheikh, 2007)

$$\lambda(t) = \lambda_b + \lambda_s h(\varphi(t)) \tag{2}$$

where λ_b is a background arrival rate, λ_s is a source arrival rate, $h(\varphi)$ is the periodic pulsar profile, and $\varphi(t)$ is the phase of the pulsar signal.

Epoch folding is the procedure that recovers the X-ray pulsar intensity density function from the measured photons' Times Of Arrival (TOAs). This procedure was formulated mathematically by Emadzadeh et al. (2009).

Assume that the spacecraft continually observes a pulsar for N_P pulsar periods, and a pulsar period is divided into N_b bins equally. So the length of each bin is $T_b = P/N_b$,

https://doi.org/10.1017/S0373463315000284 Published online by Cambridge University Press

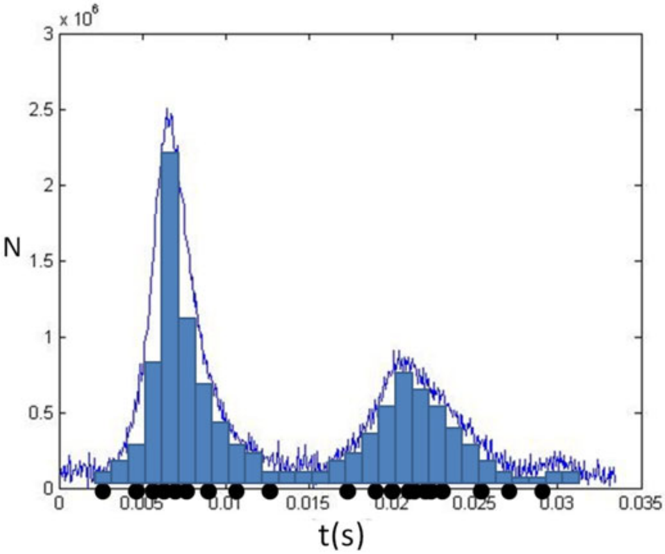

Figure 1. A frequency distribution histogram.

where P is the period of the pulsar. Let $c(t_i)$ represent the number of photons in the ith bin whose centre is t_i, then the empirical rate function can be written as (Emadzadeh and Speyer, 2009)

$$\hat{\lambda}(t_i) = \frac{1}{N_P T_b} \sum_{j=1}^{N_P} c_j(t_i) \tag{3}$$

and it can be proved that for a long observation time T_{obs},

$$\hat{\lambda}(t_i) = \lambda(t_i) + \hat{n}(t_i) \tag{4}$$

where $\hat{n}(t_i)$ is an uncorrelated noise that satisfies

$$E[\hat{n}(t_i)] = 0 \tag{5}$$

$$\text{var}[\hat{n}(t_i)] = \frac{N_b}{T_{obs}} \lambda(t_i) \tag{6}$$

The overall rate function $\lambda(t)$ can be regarded as a probability density function of the arrival photons. The purpose of epoch folding is to recover the probability density function using the frequency distribution histogram of the samples (i.e., TOAs of photons).

Figure 1 shows a typical frequency distribution histogram, where a photon (black spot) is represented by a rectangle with width of T_b and a unit height. By adding up all these rectangles, we can obtain the empirical rate function $\hat{\lambda}(t_i)$.

Assume that the number of photons arriving during the observation time is N. For every photon, it is necessary to find the exact bin it falls in and update the corresponding $c_j(t_i)$. So, the computation complexity of the epoch folding procedure is $O(N)$.

https://doi.org/10.1017/S0373463315000284 Published online by Cambridge University Press

In Section 3.2, it will be shown that representing a photon by a delta-function instead of a rectangle can diminish the computational complexity.

2.3. *Phase delay estimators.* In this subsection, three methods to estimate the phase delay are presented: maximum-likelihood estimator, non-linear least squares estimator and cross-correlation estimator.

The fundamental of maximum-likelihood estimator is to find a phase delay that maximises the joint probability of the observed TOAs (Sala et al., 2008). In practice, log-likelihood is commonly used

$$\Psi(\varphi) = \sum_{i=1}^{N} \ln(\lambda(t_i; \varphi)) \tag{7}$$

And the estimation of phase delay is

$$\hat{\varphi} = \arg\max \Psi(\varphi) \tag{8}$$

which requires optimising a non-convex function, and the grid search is a general choice. Assume that we search N_b candidates of φ, and for every φ, $\psi(\varphi)$ should be computed, which requires a computational complexity of $O(NN_b)$. Besides, finding the maximum of all the $\psi(\varphi)$s needs $O(N_b)$ complexity. Therefore, the total computational complexity is $O(NN_b + N_b)$ for the maximum-likelihood estimator.

The non-linear least squares estimator minimises a so-called cost function (Emadzadeh et al., 2009)

$$J(\varphi) = \sum_{n=1}^{N_b} \left(\hat{\lambda}(n; \varphi) - \lambda(n)\right)^2 \tag{9}$$

where $\lambda(n)$ and $\hat{\lambda}(n; \varphi)$ are the discretizations of the overall and empirical rate functions, respectively. The estimation of phase delay is

$$\hat{\varphi} = \arg\min J(\varphi) \tag{10}$$

This process needs epoch folding with the complexity of $O(N)$, and for every φ we should compute $J(\varphi)$, which requires a complexity of $O(N_b^2)$. To find the minimal of $J(\varphi)$, $O(N_b)$ a computational cost is generated. Thus, the total computational complexity of the non-linear least squares estimator is $O(N + N_b^2 + N_b)$.

The cross-correlation estimator maximises the cross correlation function of the overall rate function and the empirical rate function (Emadzadeh and Speyer, 2011a), which can be expressed as

$$R(\varphi) = \int_{-\infty}^{\infty} \lambda^*(t)\hat{\lambda}(t; \varphi)dt \tag{11}$$

And the estimation of phase delay is

$$\hat{\varphi} = -\arg\max R(\varphi) \tag{12}$$

In practice, DFT is usually employed to compute the cross correlation function

$$\begin{aligned} \Lambda_1(k) = DFT[\lambda(n)],\ \Lambda_2(k) = DFT[\hat{\lambda}(n)], \\ R(n) = IDFT[\Lambda_1(k) * \Lambda_2{}^*(k)] \end{aligned} \tag{13}$$

This process also needs epoch folding, and the DFT process requires a computational cost of $O(N_b \log N_b)$ with the fast Fourier transformation (FFT) algorithm

https://doi.org/10.1017/S0373463315000284 Published online by Cambridge University Press

Table 1. Computational complexities of estimators.

	Maximum-Likelihood estimator	Non-linear least squares estimator	Cross-correlation estimator
Computational complexity	$O(NN_b + N_b)$	$O(N + N_b^2 + N_b)$	$O(N + N_b \log N_b + N_b)$

(Cooley and Tukey, 1965). Then the maximum of $R(n)$ can be found with a complexity of $O(N_b)$. So the total complexity is $O(N + N_b \log N_b + N_b)$ for the cross-correlation estimator.

Table 1 summarises the computational complexities of the above three estimators. It can be seen that when the number of photons N and the number of bins N_b are large, the cross-correlation estimator has the least computational cost.

3. A DFT-BASED METHOD FOR PHASE DELAY ESTIMATION.

3.1. *Proposal of the DFT based method.* In this section, a method for phase delay estimation based on the time-shift property of DFT is proposed.

According to the time-shift property of DFT, a cyclic shift of a sequence in time domain is equivalent to a phase shift in frequency domain. That is, if

$$\begin{cases} X(k) = DFT[x(n)] \\ y(n) = x(n-m)_N \end{cases} \tag{14}$$

then,

$$DFT[y(n)] = X(k)\exp\left(-j\frac{2\pi}{N}mk\right) \tag{15}$$

where $j^2 = -1$.

As a consequence, the time shift in time domain m can be obtained by computing the phase shift $-\frac{2\pi}{N}mk$ in frequency domain.

The intention of phase delay estimation is to find a phase delay $\hat{\varphi}$ in time domain such that

$$\hat{\lambda}(n) = \lambda(n-\hat{\varphi})_{N_b} + \hat{n}(n) \tag{16}$$

where $\hat{n}(n)$ is an uncorrelated noise.

Assume that

$$\begin{cases} \Lambda_1(k) = DFT[\lambda(n)] \\ \Lambda_2(k) = DFT[\hat{\lambda}(n)] \end{cases} \tag{17}$$

and according to the time-shift property of DFT,

$$DFT[\lambda(n-\hat{\varphi})] = \Lambda_1(k)\exp\left(-j\frac{2\pi}{N_b}\hat{\varphi}k\right) \tag{18}$$

It can then be obtained by performing a DFT on Equation (16) that

$$\Lambda_2(k) = \Lambda_1(k)\exp\left(-j\frac{2\pi}{N_b}\hat{\varphi}k\right) + N(k), \quad 0 \le k \le N_b - 1 \tag{19}$$

https://doi.org/10.1017/S0373463315000284 Published online by Cambridge University Press

where $N(k) = DFT[\hat{n}(n)]$ is the DFT of the noise, and Equation (19) is valid for any $k \in [0, N_b - 1]$.

So, the estimation of phase delay is

$$\hat{\varphi} = -\frac{N_b}{2\pi k} ang \frac{\Lambda_2(k)}{\Lambda_1(k)} \tag{20}$$

where $ang \frac{\Lambda_2(k)}{\Lambda_1(k)}$ is the phase angle of the complex $\frac{\Lambda_2(k)}{\Lambda_1(k)}$, and Equation (20) is also valid for any $k \in [0, N_b - 1]$.

Instead of computing the whole DFT sequence, this method only computes $\Lambda_1(k)$ and $\Lambda_2(k)$ of a certain $k \in [0, N_b - 1]$ to obtain the estimation of phase delay. Using the definition of DFT, it can be obtained that

$$\Lambda_2(k) = \sum_{n=1}^{N_b} \hat{\lambda}(n) \exp\left(-j\frac{2\pi}{N_b}nk\right) \tag{21}$$

With the Horner Algorithm (Horner, 1819), the computations of $\Lambda_1(k)$ and $\Lambda_2(k)$ by Equation (21) only require $O(N_b)$ complexity. This process also needs epoch folding with the complexity of $O(N)$. As a result, the total computational complexity is $O(N + N_b)$, smaller than that of estimators presented in Section 2.

As stated above that for any $k \in [0, N_b - 1]$, the phase delay can be estimated by Equation (20). The best estimation of phase delay can be obtained by choosing the 'best' k which minimises the effect of the noise $\hat{n}(n)$.

Equation (19) leads to

$$\frac{\Lambda_2(k)}{\Lambda_1(k)} = \exp\left(-j\frac{2\pi}{N_b}\hat{\varphi}k\right) + \frac{N(k)}{\Lambda_1(k)} \tag{22}$$

Since the amplitude and phase angle of $N(k)$ are random, the k to be chosen is the one that maximises the amplitude of $\Lambda_1(k)$.

$$k = \arg\max |\Lambda_1(k)| \tag{23}$$

In this way, the effect of $\frac{N(k)}{\Lambda_1(k)}$ can be minimised.

Searching for the 'best' k requires the computation of the whole DFT sequence of $\lambda(n)$, but the overall rate function $\lambda(n)$ is 'known' before the spacecraft launched, so the 'best' k and corresponding $\Lambda_1(k)$ can be found 'offline', which means the computations of k and corresponding $\Lambda_1(k)$ impose no complexity for on board computation.

3.2. *Delta-Function Approximation.* Since k and $\Lambda_1(k)$ are obtained 'offline', during the process of navigation, only $\Lambda_2(k)$ is computed using Equation (21), where the empirical rate function $\hat{\lambda}(n)$ can be obtained through the epoch folding process. In this subsection, a delta-function approximation is put forward to replace epoch folding. With this approximation, one can directly obtain $\Lambda_2(k)$ in a lower complexity.

As discussed before, the epoch folding procedure regards photons as rectangles and uses a histogram to estimate the overall probability density function. Similarly, we can regard the photons as delta functions

$$\delta(t - \mu) = \begin{cases} \infty, t - \mu = 0 \\ 0, t - \mu \neq 0 \end{cases} \tag{24}$$

https://doi.org/10.1017/S0373463315000284 Published online by Cambridge University Press

whose centres are the 'fractions' of TOAs. Assume that the TOA of a photon contains N_P pulsar periods, then the 'fraction' is defined as

$$\mu = TOA - N_P P \tag{25}$$

The empirical rate function $\hat{\lambda}(t)$ can be redefined by

$$\hat{\lambda}(t) = \sum_{i=1}^{N} \delta(t - \mu_i) \tag{26}$$

where μ_i is the 'fraction' of the ith photon.

By noting the time-shift property of DFT and $DFT[\delta(t)] = 1$, it can be derived that

$$DFT[\delta(t - \mu)] = \exp(-jk\mu) \tag{27}$$

With the use of DFT on Equation (26), it can be obtained that

$$\Lambda_2(k) = \sum_{i=1}^{N} e^{-jk\mu_i} \tag{28}$$

where μ_i is the 'fraction' of the ith photon.

Equation (28) means that, with the delta-function approximation, the computation of empirical rate function $\hat{\lambda}(t)$ can be skipped and $\Lambda_2(k)$ can be obtained directly from TOAs of photons. As a consequence, the total computational cost can be reduced to $O(N)$, significantly less than that of the proposed method in Section 3.1. On the other hand, employing the delta-function approximation will not significantly lose accuracy, which will be shown in Section 4.

Furthermore, with the delta-function approximation, the memory cost is also decreased, because there is no need to save the empirical rate function $\hat{\lambda}(n)$, whose length is N_b.

3.3. *Scheme of the proposed method.* Based on the above description, the scheme of the proposed method is clarified in this subsection.

When $k > 1$, the phase angle of the complex $\frac{\Lambda_2(k)}{\Lambda_1(k)}$ may exceed 2π, meaning that the $2M\pi$ part of $ang\frac{\Lambda_2(k)}{\Lambda_1(k)}$ cannot be found directly by using trigonometric calculation. Consequently, the estimation of phase delay cannot be obtained directly through Equation (20) and it should be rewritten as

$$\hat{\varphi} = -\frac{N_b}{2\pi k}\left(ang\frac{\Lambda_2(k)}{\Lambda_1(k)} + 2M\pi\right) \tag{29}$$

where M is an integer.

To solve this problem, let us set $k = 1$ at first, and make a 'rough' estimation of $\hat{\varphi}$ by

$$\hat{\varphi}_1 = -\frac{N_b}{2\pi} ang\frac{\Lambda_2(1)}{\Lambda_1(1)} \tag{30}$$

Then, substitute the 'rough' estimation into Equation (29) to solve M

$$M = round\left(-\frac{k\hat{\varphi}_1}{N_b} - \frac{1}{2\pi} ang\frac{\Lambda_2(1)}{\Lambda_1(1)}\right) \tag{31}$$

https://doi.org/10.1017/S0373463315000284 Published online by Cambridge University Press

where *round*(x) is the arithmetical operation to find the closest integer of x. Using the solution of M, the estimation of phase delay $\hat{\varphi}$ can be finally determined by Equation (29).

The scheme of the DFT-based method with the delta-function approximation can be summarised as follows:

(1) Offline procedure:

a) Compute $\Lambda_1(k) = FFT\,[\lambda(n)]$

b) Find $k = \arg\max[\Lambda_1(k)]$

(2) Online procedure:

a) Compute $\Lambda_2(1) = \sum_{i=1}^{N} e^{-j\mu_i}$, where $\mu = TOA - N_P P$

b) Compute $\Lambda_2(k) = \sum_{i=1}^{N} e^{-jk\mu_i}$

c) 'Rough' estimation $\hat{\varphi}_1 = -\frac{N_b}{2\pi} ang \frac{\Lambda_2(1)}{\Lambda_1(1)}$

d) Find $M = round\left(-\frac{k\hat{\varphi}_1}{N_b} - \frac{1}{2\pi} ang \frac{\Lambda_2(1)}{\Lambda_1(1)}\right)$

e) $\hat{\varphi} = -\frac{N_b}{2\pi k}\left(ang \frac{\Lambda_2(k)}{\Lambda_1(k)} + 2M\pi\right)$

4. NUMERICAL SIMULATION.

As stated above, when the number of photons N and the number of bins N_b are large, the cross-correlation estimator has the lowest computational complexity ($O(N + N_b \log N_b + N_b)$) among the three estimators presented in Section 2. By using the DFT-based method, the computational complexity can be effectively reduced to $O(N + N_b)$, and with the aid of delta-function approximation, it can be further lowered to $O(N)$. In this section, a numerical simulation is made to compare the performance of the DFT-based method and the cross-correlation estimator.

In the simulation, we choose $N_b = 1048576$, and N ranges from 1024 to 1048576. The phase delay is estimated using pulsars B0833-45 and J1420-6048. By using an Intel(R) Xeon(R) 3·60 GHz CPU processor. The resulting computation time and phase delay estimation errors are listed in Tables 2 and 3.

To illustrate the comparison more clearly, Figure 2 shows the time delay estimation errors of the three methods. From the results, it can be seen that

- The phase delay estimation error of the DFT-based method is generally in the same order as that of the cross-correlation estimator.
- It takes the cross-correlation estimator about 15 seconds CPU time to estimate a phase delay, while the DFT-based method only takes a CPU time of about 0·4 seconds.
- With the use of delta-function approximation, the time consumption of the DFT-based method can be further decreased without loss of accuracy.

5. CONCLUSION.

This paper reviews some commonly used phase delay estimators in X-ray pulsar navigation. Considering the high computational complexities of these existing estimators, a new method for phase delay estimation is proposed, which is based on the time-shift property of DFT.

https://doi.org/10.1017/S0373463315000284 Published online by Cambridge University Press

Table 2. The simulation result of pulsar B0833-45.

B0833-45						
	Cross-correlation		DFT-Epoch-folding		DFT-Delta-Function	
N	CPU time(s)	Error(s)	CPU time(s)	Error(s)	CPU time(s)	Error(s)
1024	15·18761	5·23E-05	0·436654	3·41E-05	0·000246	3·41E-05
2048	15·18471	2·26E-05	0·434947	4·73E-05	0·000481	4·73E-05
4096	15·1665	1·06E-05	0·43473	2·62E-05	0·000952	2·62E-05
8192	15·16706	2·4E-05	0·435832	9·48E-06	0·001931	9·44E-06
16384	15·18009	1·15E-05	0·435303	5·88E-06	0·003842	5·92E-06
32768	15·16515	1·07E-05	0·435667	5·18E-06	0·007619	5·14E-06
65536	15·1672	1·34E-06	0·437058	2·09E-06	0·015294	2·04E-06
131072	15·17465	3·74E-05	0·438591	1·04E-05	0·03059	1·05E-05
262144	15·19263	1·05E-05	0·443876	8·15E-06	0·06114	8·19E-06
524288	15·20099	8·58E-06	0·453433	5·37E-06	0·122282	5·42E-06
1048576	15·22107	2·96E-06	0·472133	5·16E-06	0·244776	5·2E-06

Table 3. The simulation result of pulsar J1420-6048.

J1420-6048						
	Cross-correlation		DFT-Epoch-folding		DFT-Delta-Function	
N	CPU time(s)	error(s)	CPU time(s)	error(s)	CPU time(s)	error(s)
1024	15·18173	1·47E-05	0·434187	6·12E-05	0·000244	6·12E-05
2048	15·18326	9·79E-05	0·43461	3·61E-05	0·000483	3·6E-05
4096	15·18701	8·69E-05	0·434705	3·97E-05	0·000965	3·97E-05
8192	15·18594	0·000122	0·435118	9·97E-06	0·001906	1E-05
16384	15·19368	2·87E-05	0·434262	4·08E-06	0·003819	4·12E-06
32768	15·19175	1·58E-05	0·435754	3·69E-06	0·007599	3·66E-06
65536	15·16925	3·83E-05	0·436697	6·34E-08	0·015266	3·08E-08
131072	15·17498	1·24E-05	0·439096	1·24E-06	0·030651	1·21E-06
262144	15·18835	5·03E-06	0·443302	3·04E-06	0·061143	3·08E-06
524288	15·19541	1·24E-05	0·453982	1·77E-06	0·122545	1·8E-06
1048576	15·21535	1·1E-05	0·471779	2·23E-06	0·244813	2·27E-06

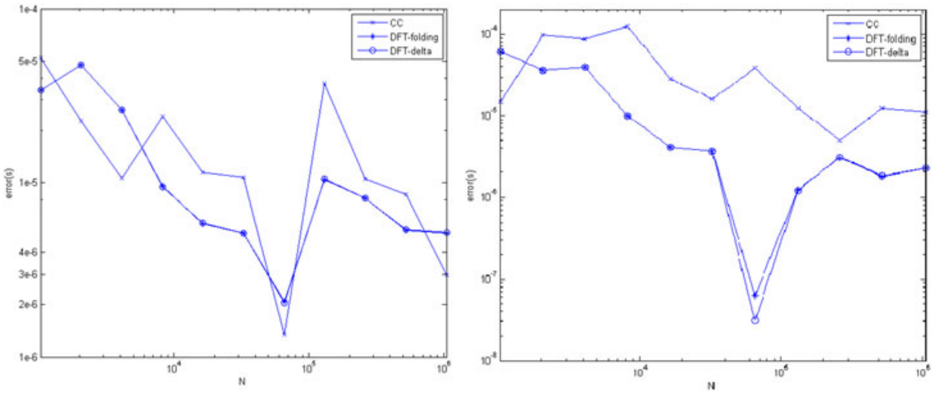

Figure 2. Time delay estimation error. Left:B0833-45. Right: J1420-6048.

https://doi.org/10.1017/S0373463315000284 Published online by Cambridge University Press

In the proposed method, only a specific term of the DFT sequence, instead of the whole DFT sequence is computed, which accelerates the estimation of phase delay. Besides that, a delta-function approximation is put forward to replace the epoch folding process. With this approximation, both the time complexity and memory cost can be greatly reduced.

Finally, a numerical simulation was conducted to evaluate the performance of the proposed method. The simulation results indicate that the DFT-based method with delta-function approximation has the lowest computational complexity and the same-order accuracy in phase delay estimation. Due to its efficiency and superiority, the proposed method will be more suitable for future on board application.

ACKNOWLEDGMENTS

This work was carried out with financial support from the National Basic Research Program 973 of China (2013CB834103).

REFERENCES

Cooley, J.W. and Tukey, J.W. (1965). An algorithm for the machine calculation of complex Fourier series. *Mathematics of computation*, 1965, **19**(90), 297–301.

Downs, G.S. (1974). Interplanetary Navigation Using X-ray Pulsating Radio Sources. *NASA Technical Reports* N74-34150, Oct.1974, 1–12.

Emadzadeh, A.A, Golshan, A. R. and Speyer, J.L. (2009). Consistent Estimation of Pulse Delay for X-ray Pulsar Based Relative Navigation. *Joint 48th IEEE Conference on Decision and Control and 28th Chinese Control Conference*, Shanghai, 2009.

Emadzadeh, A.A. and Speyer, J.L. (2011a). X-Ray Pulsar-Based Relative Navigation using Epoch Folding. *IEEE Transactions on Aerospace and Electronic Systems*, **47**, No 4, 2317–2328.

Emadzadeh, A.A. and Speyer, J.L. (2011b). *Navigation in Space by X-ray Pulsars*. Springer New York, Dordrecht, Heidelberg, London.

Emadzadeh, A.A. and Speyer, J.L. (2010). On modelling and pulse phase estimation of X-ray pulsars. *IEEE Transactions on Signal Processing*, **58**(9), 4484–4495.

Golshan, A.R. and Sheikh, S.L. (2007). On Pulsar Phase Estimation and Tracking of Variable Celestial X-ray Sources. 6*3rd Annual Meeting of Institute of Navigation (ION)*, Cambridge, MA, USA, April, 413–422.

Horner, W.G. (1819). A new method of solving numerical equations of all orders, *by continuous approximation.* Philosophical Transactions of the Royal Society of London, **1819**, 308–335.

Sala, J., Urruela, A., Villares, X., Estalella, R. and Paredes, J.M. (2004). Feasibility Study for a Spacecraft Navigation System relying on Pulsar Timing Information. Final Report, ESA Advanced Concepts Team, ARIADNA Study 03/4202.

Sala, J., Urruela, A., Villares, X., Jordi, R. and Sebastia, B. (2008). Pulsar Navigation. ESA Research Report, Project ARIADNA, January 14, 2008.

Sheikh, S.I. and Pines, D.J. (2006). Recursive estimation of spacecraft position and velocity using X-ray pulsar time of arrival measurements. *Navigation*, **53**, 149–166.

https://doi.org/10.1017/S0373463315000284 Published online by Cambridge University Press