IMTC 2007 – IEEE Instrumentation and Measurement
Technology Conference
Warsaw, Poland, May 1–3, 2007

Spectral Observers for Unevenly Sampled Data

K. Molnár, G. Péceli

Department of Measurement and Information Systems,
Budapest University of Technology and Economics,
Magyar tudósok krt. 2., 1521 Budapest, Hungary.
Phone:+36-1-463-2057, Fax:+36-1-463-4112, E-mail: karoly.molnar@mit.bme.hu

Abstract – *This paper investigates the problem of spectral observation in case of unevenly sampled input data. In [1] Hostetter proposed a dead-beat observer structure and an algorithm to calculate the observer gain values online. In this paper we give a formal proof of correctness of this algorithm and we also propose an efficient numerical method to reduce the amount of computations required by the algorithm. Moreover, we show a significantly more efficient solution of spectral observation for a special case of uneven sampling, namely when the sampling time instants are uneven but periodic.*

Keywords – *Observers, Signal sampling, Spectral analysis, Discrete Fourier transforms*

I. INTRODUCTION

The theory of spectral observers is a well-studied area, where the Fourier spectral analysis is performed in real-time, by ongoing recursive calculations [2]. The spectral observer structure is derived from the state-variable formulation and it is essentially a deadbeat state observer. Spectral observers can be used as parametric signal estimators when a measured signal is considered as an output signal of a system represented by state space model. Fig.1 shows the block diagram of the spectral observer structure. A computationally efficient realization is already developed for the regularly sampled case in [3]. The main advantage of this structure is its realizability in hard real-time embedded systems, as the coefficients can be computed off-line and only modest amount of calculations need to be performed online.

Unfortunately, the efficient solution heavily rests on the assumption that the input signal is sampled regularly. However, in some cases this assumption does not hold, as it is possible that due to implementation issues or resource constraints only uneven samples are available. This problem can arise in embedded environments where the designer has to deal with serious resource constraints. E.g. the processor has to share its limited amount of CPU cycles among its tasks and it can not be guaranteed that sampling is performed precisely at a given time instant if it is not the highest priority task.

In this paper we investigate the problem of spectral observers working with unevenly sampled data. In Section II we outline the background of our work. In Section III we examine a solution given by Hostetter [1] to the general case of the problem of spectral observation for unevenly sampled data. We give the correctness proof of this solution in Appendix A and we propose a numerical method that speeds up the calculation of the algorithm in Appendix B. In Section IV we propose a different, efficiently implementable solution to a special case of uneven sampling. Finally, we draw the conclusions.

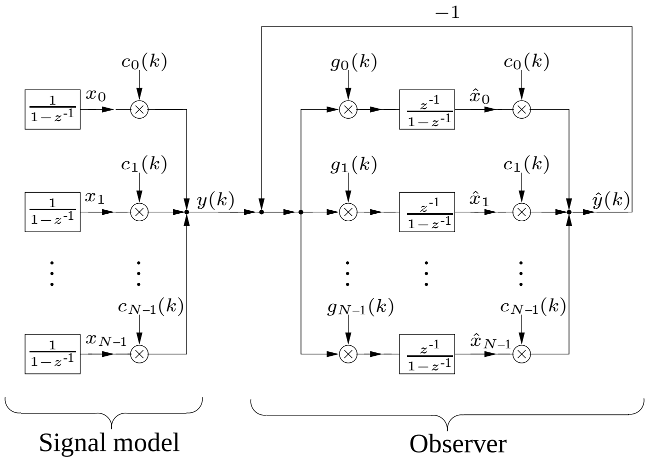

Figure 1. Spectral observer

II. UNEVEN SAMPLING

The block diagram of the conceptual signal model and the basic observer structure is shown in Fig.1. Let's denote y_k the sample of the input signal at sampling time instant t_k. Similarly, the values $c_m(k)$ and $g_m(k)$ denote the coefficients of the model at the time instant t_k. Note that in case of regular sampling $t_k = kT_s$ where T_s is the sampling interval.

Several different observers can be realized by the above structure, depending on the values $c_m(k)$. If $c_m(k) = e^{j(2\pi/N)mk}$, then a spectral observer is realized. Let's denote $\mathbf{c}(i) = [c_0(i)\ c_1(i)\ \ldots c_{N-1}(i)]^T$, $\mathbf{g}(i) = [g_0(i)\ g_1(i)\ \ldots g_{N-1}(i)]^T$ and similarly $\mathbf{x}(i) = [x_0(i)\ x_1(i)\ \ldots x_{N-1}(i)]^T$. The observer is termed deadbeat

1-4244-0589-0/07/$20.00 ©2007 IEEE

if $\mathbf{x}(k{+}1){-}\hat{\mathbf{x}}(k{+}1) = 0 \;\; \forall k > N$. From the state equations it can be derived that:

$$\prod_{i=k-N+1}^{k} \Big[\mathbf{I}{-}\mathbf{g}(i)\mathbf{c}^T(i)\Big] = 0 \quad \Rightarrow \quad \mathbf{x}(k{+}1){-}\hat{\mathbf{x}}(k{+}1) = 0 \tag{1}$$

for $\forall k > N$. It is shown in [3], that if the signal is evenly sampled, then E.q. 1 is satisfied for every basis/reciprocal basis system. In case of the spectral observer $c_m(k) = e^{j(2\pi/N)mk}$ and $g_m(k) = \frac{1}{N}e^{-j(2\pi/N)mk}$. In case of uneven sampling there is no constant sampling frequency, hence explicit time indexing has to be used. The terms f_s and T_s can be eliminated from the coefficients by multiplying the exponents by $f_sT_s = 1$:

$$c_m(k) = e^{j(2\pi/N)mk} = e^{\frac{j2\pi f_s}{N}m(kT_s)} = e^{j\omega_1 m t_k}$$

where $\omega_1 = 2\pi f_s/N$ is the angular frequency of the first Fourier component in the model (the component with lowest nonzero angular frequency).

III. GENERAL CASE - OBSERVER GAIN CALCULATION ALGORITHM

In [1] Hostetter proposed a solution that provides a deadbeat spectral observer working with arbitrarily sampled data. The observer structure used by Hostetter is equivalent to the system depicted in Fig.1, the only difference is that the observer gain values $\mathbf{g}(k)$ are not known a priori, they are calculated online for every new time instant. The calculation of the $\mathbf{g}(k)$ values is done by a computationally intensive algorithm, the Observer Gain Calculation algorithm (OGC). The OGC algorithm itself was published in [1] but the formal proof of the deadbeat property of the observer is absent to the best of our knowledge. We give the proof in Appendix A.

The OGC algorithm requires the calculation of $N-1$ matrix multiplications for each sampling time instant. The complexity of matrix multiplication is in itself $O(N^3)$ in practice, i. e. the complexity of the algorithm is $O(N^4)$ if the calculations are carried out naively. In Appendix B we show an efficient numerical method which reduces the required computational complexity to $O(N^{3.5})$.

IV. SPECIAL CASE - PERIODIC UNEVEN SAMPLING

In case the samples are taken at random time instants then the above presented OGC algorithm is the most efficient known solution for the spectral observer. However, we show that a more efficient solution does exist for a special case of uneven sampling. This special case is the periodic uneven sampling, when there are repeating patterns in the sampling time instants. If there is periodicity in the sampling process, then the sampling time instants follow each other with arbitrary difference, but the time difference between t_k and t_{k+N} is constant: $t_{(k+N)} - t_k = T_p \quad \forall k = 0..\infty$.

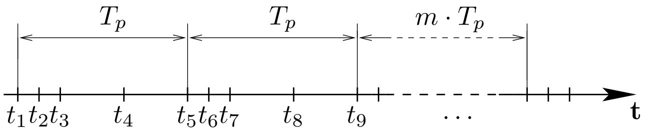

Figure 2. Periodic uneven sampling, N=4

Fig.2 shows an example of periodic uneven sampling with $N = 4$, where $t_1, t_2, ...t_i$ are the sampling time instants and T_p is the period. This situation, when a pattern N long is repeated periodically in the sampling time instants is realistic in some applications, e.g. when a burst of samples are taken periodically (burst sampling).

If the above property is true and the repeating pattern is known a priori, then the spectral observer can be utilized to reconstruct the measured signal in a deadbeat fashion i.e. with zero error. Denote $\mathbf{C}(k)$ the N x N matrix formed from the last N $\mathbf{c}(k)$ column vectors: $\mathbf{C} = [\mathbf{c}(k)\; \mathbf{c}(k-1)\; \mathbf{c}(k-2)\; \ldots\; \mathbf{c}(k-N+1)]$ and similarly $\mathbf{G} = [\mathbf{g}(k)\; \mathbf{g}(k-1)\; \mathbf{g}(k-2)\; \ldots\; \mathbf{g}(k-N+1)]$.

With these notations it can be easily shown that if we choose $\mathbf{G}(k)$ to satisfy $\mathbf{G}(k) = \mathbf{C}(k)^{-1}$, then Eq.1 is satisfied for $\forall k > N$. As $\mathbf{C}(k)$ is an N x N matrix formed from the last N $\mathbf{c}(i)$ column vectors ($i = k, \ldots k-N+1$), therefore $\mathbf{C}(k+1)$ is formed from the column vectors $\mathbf{c}(i),\ i = k+1, \ldots k-N$, i.e. it is a shifted version of the columns of $\mathbf{C}(k)$ except for the last column. If we ensure that the last column of $\mathbf{C}(k+1)$ is the same as the first column of $\mathbf{C}(k)$ for all k, then all $\mathbf{C}(i)$ matrices will consist of the same columns, but they are circularly shifted. In this case all of the corresponding inverse matrices $\mathbf{G}(i)$ will consist of the same rows which are shifted circularly. This means that the $\mathbf{G}(k)$ can be calculated off-line and the actual $\mathbf{g}(k)$ values can be read out easily.

In the following, we show that the columns $\mathbf{c}(i+N)$ and $\mathbf{c}(i)$ are equal if and only if the time difference between the sampling time instants t_i and $t_i - N$ is a multiplier of T_1, the period of the lowest frequency Fourier component in the model.

$$\mathbf{c}(i{+}N) = \mathbf{c}(i) \iff c_m(i{+}N) = c_m(i) \quad \forall m \in 1 \ldots N{-}1$$

$$\begin{aligned} e^{jm\omega_1 t_{i+N}} &= e^{jm\omega_1 t_i} \\ jm\omega_1 t_{i+N} &= jm\omega_1 t_i + k2\pi \qquad k \in \mathbf{N} \\ t_{i+N} &= t_i + \frac{k2\pi}{m\omega_1} = t_i + \frac{kT_1}{m} \\ t_i - t_{i-N} &= k\frac{T_1}{m} \qquad \forall m \in 1 \ldots N-1 \qquad k \in \mathbf{N} \end{aligned}$$

This has to be true for all $m \in 1 \ldots N-1$. But as k can be arbitrary, if the property is true for $m = 1$ then it is true for all other m, as an appropriate k multplier can always be found to satisfy the equation. We can summarize this as follows:

$$c(i+N) = c(i) \iff t_i - t_{i-N} = T_p = kT_1. \tag{2}$$

An attractive property of the spectral observer is that the frequencies of the Fourier components can be tuned easily by using the Adaptive Fourier Analyzer (AFA) algorithm proposed in [4]. If the AFA is used, Eq. 2 can be forced by tuning the frequency of the first Fourier component to $\frac{1}{T_p}$.

Another condition that falls out from the above deduction is that exactly N arbitrary samples are needed from the T_p period. If there are more samples available, N of them has to be chosen in a way that the N different $\mathbf{c}(k)$ vectors are linearly independent of each other to ensure that the $\mathbf{C}(k)$ matrix is invertible.

It is important to point out a possible numerical problem of the solution. In case the $\mathbf{C}(k)$ matrix is ill-conditioned i.e. its condition number is so large that its reciprocal is close to the numerical precision of the computing machine, then the calculated $\mathbf{G}(k)$ values can be very unprecise. Therefore, if there are more than N samples are available from the T_p period, those samples have to be chosen which provide the minimum condition number for $\mathbf{C}(k)$. In case this minimum condition number is still too large, then the spectral observation is not possible by the proposed solution.

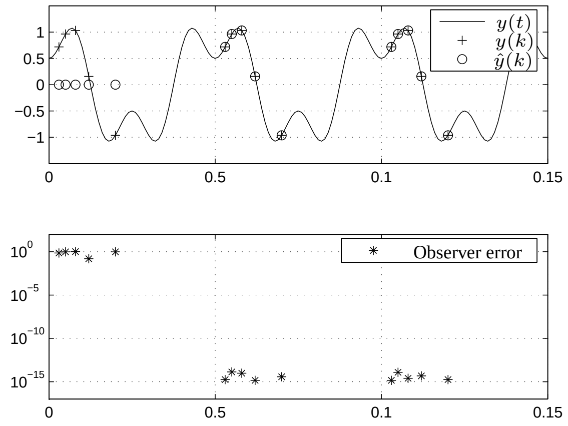

Figure 3. Observation of a periodic signal and the observer error

In Fig. 3 we present an example with an input signal sampled periodically and unevenly ($N = 5$). The input signal is a simple sinusoid formed from two components, a base harmonic and the third upper harmonic. The original analog signal $y(t)$ is plotted with a solid line, the sampled signal $y(k)$ is represented by the "+" signs, and the output of the observer $\hat{y}(k)$ is shown by the "o" signs. in the second figure the observer error $[e(k) = y(k) - \hat{y}(k)]$ is plotted with a logarithmic scale. It is clear that the observer error is reduced to zero (within numeric precision) after N steps, proving the deadbeat property.

V. CONCLUSIONS

This paper investigates the problem of spectral observation in case of unevenly sampled input data. Spectral observation is a well researched area, where efficient solutions exists for standard, evenly sampled input data. Spectral observers are designed to be dead-beat i.e. the observation error is reduced to zero after a finite number of samples. In order to handle the unevenly sampled input, the same structure and state-space description is used as in the evenly sampled case, but the observer gain values $\mathbf{g}(k)$ are not known a priori, they have to be calculated online. Hostetter proposed an algorithm (the OGC algorithm) that calculates these observer gain values for arbitrarily unevenly sampled data in [1].

In this paper we give formal proof of correctness of the OGC algorithm in Appendix A. Moreover, we propose a numerical method to speed up the calculation of the OGC algorithm in Appendix B. This method significantly reduces the required number of computations in exchange for more memory usage. Regardless of the proposed faster realization the online calculation of the OGC algorithm is still a computationally intensive task. In Section IV we show that further simplifications can be made, for a special case of uneven sampling. In case the sampling time instants have a repeating pattern (periodical uneven sampling), we show a solution where the observer gain values can be calculated off-line. The solution therefore requires significantly less online calculations, making it implementable also in low-resource embedded environments.

APPENDIX A.

However the OGC algorithm itself was proposed in [1] by Hostetter, it's formal proof of correctness was not published so far. In the following we rehearse the original algorithm and afterwards we give the formal proof of it's correctness, e.g. it provides a deadbeat observer.

A. The formal algorithm

The algorithm is the following. Let $\mathbf{u}_1, \mathbf{u}_2 \cdots \mathbf{u}_N$ be the unit coordinate vectors, and denote $\mathbf{F}_i = \mathbf{I} - \mathbf{g}(i)\mathbf{c}^T(i)$. Let $\mathbf{F}_0 = \mathbf{F}_{-1} = ... = \mathbf{F}_{-N+2} = \mathbf{I}$.

At each new sampling time instant t_k, a new $\mathbf{c}(k)$ vector is obtained directly from the system model. The values of the $\mathbf{c}(k)$ vector are dependent on t_k only. The corresponding $\mathbf{g}(k)$ is calculated from the following equation:

$$\mathbf{g}(k) = \frac{\mathbf{F}_{k-1}\,\mathbf{F}_{k-2}\ldots\mathbf{F}_{k-N+1}\,\mathbf{u}_{k \bmod N}}{\mathbf{c}^T(k)\,\mathbf{F}_{k-1}\,\mathbf{F}_{k-2}\ldots\mathbf{F}_{k-N+1}\,\mathbf{u}_{k \bmod N}} \qquad (3)$$

In case the denominator of Eq. 3 is zero, then the calculation has to be repeated with another unit vector instead of $\mathbf{u}_{k \bmod N}$. At least one unit vector $\mathbf{u}_i$ can always be found that makes the $[\mathbf{F}_{k-1}\ \mathbf{F}_{k-2}\ldots\mathbf{F}_{k-N+1}\ \mathbf{u}_i]$ expression nonzero. In case the $[\mathbf{c}^T(k)\ \mathbf{F}_{k-1}\ \mathbf{F}_{k-2}\ldots\mathbf{F}_{k-N+1}\ \mathbf{u}_i]$ is still zero with all of the unit vectors, then the $\mathbf{c}^T(k)$ vector is in the null space of

$\mathbf{F}_{k-1}\ \mathbf{F}_{k-2}\ldots\mathbf{F}_{k-N+1}$. In this case, no solution exists for $\mathbf{g}(k)$ and therefore the measurement at sampling time instant t_k can be discarded.

B. Proof of the deadbeat property

We introduce the following notations: $\mathrm{rank}(\mathbf{F}_i) = \dim\mathrm{range}(\mathbf{F}_i)$, and $\dim\ker(\mathbf{F}_i) = \mathrm{null}(\mathbf{F}_i)$, where $\mathrm{null}(\mathbf{A})$ denotes the nullity of a matrix $\mathbf{A}$. Before proving that the observer is deadbeat with the above calculated gains, we prove the following two lemmata:

Lemma 1: Let $\mathbf{F}_i = \mathbf{I} - \mathbf{g}(i)\mathbf{c}^T(i)$. If $\exists x \neq 0$ where $\mathbf{F}_i x = 0$ then $\mathrm{rank}(\mathbf{F}_i) = N-1$.

Proof: First we prove that $\mathrm{rank}(\mathbf{F}_i) < N$. As $\exists \mathbf{x} \neq 0$ where $\mathbf{F}_i\mathbf{x} = 0$, therefore the columns of $\mathbf{F}_i$ are linearly dependent, i.e. $\mathrm{rank}(\mathbf{F}_i) < N$.

Second, we prove by contradiction that $\mathrm{rank}(\mathbf{F}_i) \geq N-1$. Let's assume that $\mathrm{rank}(\mathbf{F}_i) < N-1$. In this case $\mathbf{F}_i$ has a minimal dyadic decomposition [5] of maximum $N-2$ dyads. If we add one more dyad (i.e. the $\mathbf{g}(i)\mathbf{c}^T(i)$ dyad) to $\mathbf{F}_i$, then the minimal dyadic decomposition of $\mathbf{F}_i + \mathbf{g}(i)\mathbf{c}^T(i)$ consists of maximum $N-1$ dyads. But $\mathbf{F}_i + \mathbf{g}(i)\mathbf{c}^T(i) = \mathbf{I}$, and $\mathrm{rank}(\mathbf{I}) = N$, which is a contradiction. ■

Lemma 2: If $\mathbf{A}_1, \mathbf{A}_2, ..\mathbf{A}_{N-1}$ are NxN matrices and $\mathrm{rank}(\mathbf{A}_1) = \mathrm{rank}(\mathbf{A}_2) = \cdots = \mathrm{rank}(\mathbf{A}_{N-1}) = N-1$, then $\mathrm{rank}(\mathbf{A}_{N-1}\mathbf{A}_{N-2}\cdots\mathbf{A}_2\mathbf{A}_1) \geq 1$.

Proof: Applying Sylvester's nullity theorem [5] to the product matrices:

$$\begin{aligned}
\mathrm{rank}(\mathbf{A}_2\mathbf{A}_1) &\geq \mathrm{rank}(\mathbf{A}_2) - \mathrm{null}(\mathbf{A}_1)\\
\mathrm{rank}(\mathbf{A}_2\mathbf{A}_1) &\geq (N-1) - 1 = N-2\\
\mathrm{rank}(\mathbf{A}_3\mathbf{A}_2\mathbf{A}_1) &\geq \mathrm{rank}(\mathbf{A}_2\mathbf{A}_1) - \mathrm{null}(\mathbf{A}_3)\\
\mathrm{rank}(\mathbf{A}_3\mathbf{A}_2\mathbf{A}_1) &\geq (N-2) - 1 = N-3\\
&\vdots\\
\mathrm{rank}(\mathbf{A}_{N-2}\cdots\mathbf{A}_1) &\geq \mathrm{rank}(\mathbf{A}_{N-3}\cdots\mathbf{A}_1) - \mathrm{null}(\mathbf{A}_{N-2})\\
\mathrm{rank}(\mathbf{A}_{N-2}\cdots\mathbf{A}_1) &\geq 3 - 1 = 2\\
\mathrm{rank}(\mathbf{A}_{N-1}\cdots\mathbf{A}_1) &\geq \mathrm{rank}(\mathbf{A}_{N-2}\cdots\mathbf{A}_1) - \mathrm{null}(\mathbf{A}_{N-1})\\
\mathrm{rank}(\mathbf{A}_{N-1}\cdots\mathbf{A}_1) &\geq 2 - 1 = 1
\end{aligned}$$

■

Theorem 1: The spectral observer is **deadbeat** if the observer gain vectors are calculated by the algorithm presented in section A.

Proof:

(i) First we prove that equation (1) is satisfied after the first N steps. The $\mathbf{g}(1)$, $\mathbf{g}(2)$, ... $\mathbf{g}(N)$ values are calculated according to equation (3):

$$\begin{aligned}
\mathbf{g}(1) &= \frac{\mathbf{u}_1}{\mathbf{c}^T(1)\ \mathbf{u}_1}\\
\mathbf{g}(2) &= \frac{\mathbf{F}_1\ \mathbf{u}_2}{\mathbf{c}^T(2)\ \mathbf{F}_1\ \mathbf{u}_2}\\
\mathbf{g}(3) &= \frac{\mathbf{F}_2\ \mathbf{F}_1\ \mathbf{u}_3}{\mathbf{c}^T(3)\ \mathbf{F}_2\ \mathbf{F}_1\ \mathbf{u}_3}\\
&\vdots\\
\mathbf{g}(N) &= \frac{\mathbf{F}_{N-1}\ \mathbf{F}_{N-2}\ldots\mathbf{F}_1\ \mathbf{u}_N}{\mathbf{c}^T(N)\ \mathbf{F}_{N-1}\ \mathbf{F}_{N-2}\ldots\mathbf{F}_1\ \mathbf{u}_N}
\end{aligned}$$

After elementary reordering of the equations:

$$\begin{aligned}
&\Big[\mathbf{I} - \mathbf{g}(1)\mathbf{c}^T(1)\Big]\mathbf{u}_1 = 0 \ \Rightarrow\ \mathbf{F}_1\mathbf{u}_1 = 0\\
&\Big[\mathbf{I} - \mathbf{g}(2)\mathbf{c}^T(2)\Big]\ \mathbf{F}_1\mathbf{u}_2 = 0 \ \Rightarrow\ \mathbf{F}_2\mathbf{F}_1\mathbf{u}_2 = 0\\
&\Big[\mathbf{I} - \mathbf{g}(3)\mathbf{c}^T(3)\Big]\ \mathbf{F}_2\mathbf{F}_1\mathbf{u}_3 = 0 \ \Rightarrow\ \mathbf{F}_3\mathbf{F}_2\mathbf{F}_1\mathbf{u}_3 = 0\\
&\qquad\qquad\vdots\\
&\Big[\mathbf{I} - \mathbf{g}(N)\mathbf{c}^T(N)\Big]\ \mathbf{F}_{N-1}\cdots\mathbf{F}_1\mathbf{u}_N = 0 \ \Rightarrow\\
&\qquad\qquad\qquad\qquad \mathbf{F}_N\mathbf{F}_{N-1}\cdots\mathbf{F}_1\mathbf{u}_N = 0
\end{aligned}$$

From these equations it is obvious that:

$$\mathbf{F}_N\mathbf{F}_{N-1}\cdots\mathbf{F}_2\mathbf{F}_1 = 0.$$

(ii) Now we assume that for a given k

$$\mathbf{F}_j\mathbf{F}_{j-1}\cdots\mathbf{F}_{j-(N-2)}\mathbf{F}_{j-(N-1)} = 0 \quad \forall j < k, \tag{4}$$

e.g. we assume that the deadbeat property is satisfied for all the previous sampling time instants before t_k. We prove that the deadbeat property is maintained with the new $\mathbf{F}_k$. Let's denote

$$\tilde{\mathbf{F}}_k = \mathbf{F}_{k-1}\mathbf{F}_{k-2}\cdots\mathbf{F}_{k-(N-1)}$$

Note that $\tilde{\mathbf{F}}_k\mathbf{F}_{k-N} = 0$ according to Eq.4.

From Lemma 1 we know that $\mathrm{rank}(\mathbf{F}_{k-N}) = N-1$, so $\mathrm{null}(\mathbf{F}_{k-N}) = 1$. Applying Sylvester's nullity theorem [5] once again:

$$\begin{aligned}
\mathrm{rank}(\tilde{\mathbf{F}}_k) - \mathrm{null}(\mathbf{F}_{k-N}) &\leq \mathrm{rank}(\tilde{\mathbf{F}}_k\mathbf{F}_{k-N}) = 0\\
\mathrm{rank}(\tilde{\mathbf{F}}_k) &\leq 1. \qquad (5)
\end{aligned}$$

As $\tilde{\mathbf{F}}_k$ is a product of $(N-1)$ different rank $(N-1)$ matrices, according to Lemma 2 it is also true that:

$$\mathrm{rank}(\tilde{\mathbf{F}}_k) \geq 1. \tag{6}$$

From equations (5) and (6):

$$\mathrm{rank}(\tilde{\mathbf{F}}_k) = 1. \tag{7}$$

The last equation means that $\dim\mathrm{range}(\tilde{\mathbf{F}}_k) = 1$, e.g. exists at least one unit vector $\mathbf{u}_k$ that $\tilde{\mathbf{F}}_k\mathbf{u}_k \neq 0$. This ensures that the denominator of the formula (3) in the algorithm is nonzero unless the $\mathbf{c}(k)$ is in $\ker(\tilde{\mathbf{F}}_k)$. (If $\mathbf{c}(k)$ is in $\ker(\tilde{\mathbf{F}}_k)$ then no solution exists for the sampling time instant t_k i.e. the sample can be discarded.)

The algorithm calculates $\mathbf{g}(k)$ according to the formula (3):

$$\begin{gathered}
\mathbf{g}(k) = \frac{\tilde{\mathbf{F}}_k\ \mathbf{u}_k}{\mathbf{c}^T(k)\ \tilde{\mathbf{F}}_k\ \mathbf{u}_k}\\
\big[\mathbf{I} - \mathbf{g}(k)\mathbf{c}^T(k)\big]\ \tilde{\mathbf{F}}_k\ \mathbf{u}_k = 0
\end{gathered}$$

substituting $\mathbf{F}_k = \mathbf{I} - \mathbf{g}(k)\mathbf{c}^T(k)$:

$$\mathbf{F}_k \tilde{\mathbf{F}}_k \ \mathbf{u}_k = 0. \tag{8}$$

According to the rank-nullity theorem [6], the rank of a matrix plus its nullity equals the dimension of its domain. Since $\dim \ker(\tilde{\mathbf{F}}_k) = N - 1$, and $u_k \notin \ker(\tilde{\mathbf{F}}_k)$ therefore $\ker(\tilde{\mathbf{F}}_k)$ and $\mathbf{u}_k$ spans the whole vector space:

$$\mathrm{span}(\ker(\tilde{\mathbf{F}}_k), \mathbf{u}_k) = \mathbb{C}^N. \tag{9}$$

Denote $\mathbf{b}_1, \mathbf{b}_2, ..\mathbf{b}_{N-1}$ an arbitrary basis of $\ker(\tilde{\mathbf{F}}_k)$. If we take an arbitrary vector $\mathbf{x} \in \mathbb{C}^N$, then $\mathbf{x}$ can be expressed as the linear combination of the basis vectors:

$$\mathbf{x} = \lambda_1 \mathbf{b}_1 + \lambda_2 \mathbf{b}_2 ... + \lambda_{N-1} \mathbf{b}_{N-1} + \lambda_N \mathbf{u}_k.$$

Expressing $\mathbf{F}_k \tilde{\mathbf{F}}_k \mathbf{x}$:

$$\begin{gathered} \mathbf{F}_k \tilde{\mathbf{F}}_k \mathbf{x} = \mathbf{F}_k \tilde{\mathbf{F}}_k (\lambda_1 \mathbf{b}_1 + ... + \lambda_{N-1} \mathbf{b}_{N-1} + \lambda_N \mathbf{u}_k) = \\ \Big[\mathbf{F}_k \tilde{\mathbf{F}}_k (\lambda_1 \mathbf{b}_1 + ... + \lambda_{N-1} \mathbf{b}_{N-1}) \Big] + \mathbf{F}_k \tilde{\mathbf{F}}_k \lambda_N \mathbf{u}_k = \\ \mathbf{F}_k (\lambda_1 \tilde{\mathbf{F}}_k \mathbf{b}_1 + ... + \lambda_{N-1} \tilde{\mathbf{F}}_k \mathbf{b}_{N-1}) + \lambda_N \mathbf{F}_k \tilde{\mathbf{F}}_k \mathbf{u}_k = 0. \end{gathered}$$

The first term is zero as $\mathbf{b}_1, \mathbf{b}_2, ..\mathbf{b}_{N-1} \in \ker(\tilde{\mathbf{F}}_k)$, and according to equation (8) the second term is zero as well. This shows that $\mathbf{F}_k \tilde{\mathbf{F}}_k \mathbf{x} = 0$ with arbitrary $\mathbf{x} \in \mathbb{C}^N$, proving that

$$\begin{aligned} \mathbf{F}_k \tilde{\mathbf{F}}_k &= 0 \\ \mathbf{F}_k \mathbf{F}_{k-1} \mathbf{F}_{k-2} \ldots \mathbf{F}_{k-N+1} &= 0. \end{aligned}$$

This proves that the application of the observer gain vectors $\mathbf{g}(k)$ calculated by the OGC algorithm provides a deadbeat observer for the unevenly sampled data, e.g. the observation error is reduced to zero in maximum N samples after the parameters of the input signal are changed. ■

APPENDIX B.

The OGC algorithm requires the calculation of $N-1$ matrix multiplications for each sampling time instant. This calculation is the most computational intensive task in the algorithm. The complexity of matrix multiplication is in itself $O(N^3)$, if the computation are carried out naively. The complexity can be reduced somewhat by several algorithms [e.g. to $O(N^{2.807})$ by Strassen's algorithm]. However, in practice these algorithms are rarely used, especially in embedded implementations. Hence the complexity of the observer gain calculation algorithm is $O(N^4)$ for every sampling time instant, if the calculations are carried out naively. In the following, we show an efficient method which reduces the required computational complexity to $O(N^{3.5})$.

For every time instant, the actual $g(k)$ is calculated according to Eq. 3, so the chain matrix computation $\mathbf{F}_{k-1} \ \mathbf{F}_{k-2} \ldots \mathbf{F}_{k-N+1}$ has to be evaluated, which means the multiplication of $N-1$ matrices of size NxN. E.g. if $N = 5$, then for $\mathbf{g}(5)$ we calculate $\mathbf{F}_4\mathbf{F}_3\mathbf{F}_2\mathbf{F}_1$, for $\mathbf{g}(6)$ we calculate $\mathbf{F}_5\mathbf{F}_4\mathbf{F}_3\mathbf{F}_2$ and so forth. There is always one new matrix in the left side of the chain and the rightmost matrix is removed from the chain. [Note that the $\mathbf{F}_i$ matrices are not invertible, as $\mathrm{rank}(\mathbf{F}_k) = N - 1$.] It is not efficient to calculate the whole chain for every sampling time instant as $N - 3$ matrices in the middle of the chain are the same as in the previous calculation. The idea is to store the inner product matrices in the chain in addition to the $\mathbf{F}_i$ matrices and look them up from memory instead of recalculating them. E.g. we store the pairwise product matrices, $\mathbf{F}_{32} = \mathbf{F}_3\mathbf{F}_2$ and $\mathbf{F}_{43} = \mathbf{F}_4\mathbf{F}_3$, and we calculate the product $\mathbf{F}_4\mathbf{F}_{32}\mathbf{F}_1$ for $\mathbf{g}(5)$ and $\mathbf{F}_5\mathbf{F}_{43}\mathbf{F}_2$ for $\mathbf{g}(6)$, etc. With this solution, the number of total matrix multiplications are reduced in exchange for more memory usage.

The total number of required matrix multiplications M can be expressed:

$$M(N,k) = k - 2 + \frac{N - (N \bmod k)}{k} + (N \bmod k) \tag{10}$$

where k is the number of matrices multiplied for a product matrix (the k value was 2 in the above example).

For every different N, there is an optimal k, that can be calculated easily by finding the minimum M for a given N.

$$k_{opt}(N) = \arg\min_k M(N,k). \tag{11}$$

A possible numerical solution for finding $k_{opt}(N)$ for a given N is to compute $M(N,k)$ for all $k = 1..N$ and choose the $k_{opt}(N)$ to be the k value that resulted in the minimal M. (E.g. for $N = 5$, $k_{opt} = 2$.) In order to calculate the computational complexity of the solution, we have to find an upper bound of the number of matrix calculations with $k_{opt}(N)$.

$$\begin{gathered} M(N, k_{opt}(N)) \leq M(N, \lceil\sqrt{N}\rceil) \leq \\ \leq \lceil\sqrt{N}\rceil - 2 + \frac{N}{\lceil\sqrt{N}\rceil} + \lceil\sqrt{N}\rceil \leq 3\sqrt{N} \end{gathered}$$

The first inequality is true since $k_{opt}(N)$ is the optimal solution. The second inequality is also true as

$$\frac{N - (N \bmod \lceil\sqrt{N}\rceil)}{\lceil\sqrt{N}\rceil} \leq \frac{N}{\lceil\sqrt{N}\rceil}$$

and $(N \bmod \lceil\sqrt{N}\rceil) \leq \lceil\sqrt{N}\rceil$. The last inequality is straightforward. This proves that the required number of matrix multiplications are bounded by $3\sqrt{N}$, making the overall computational complexity to be $O(\sqrt{N}N^3 = O(N^{3.5})$.

Finally, the cost of the solution in terms of memory usage must be given. As there are $N - k + 1$ product matrices of size that have to be stored in addition to the original $\mathbf{F}_k$ matrices, the required memory is slightly less than double the original solution.

ACKNOWLEDGEMENT

This work was supported by the Hungarian Fund for Scientific Research under Grants OTKA TS49743.

[1] Gene H. Hostetter, "Recursive discrete fourier transformation with unevenly spaced data," *IEEE Trans. on Acoustics, Speech, and Signal Processing*, vol. ASSP-31, no. 1, pp. 206–209, February 1983.

[2] Gene H. Hostetter, "Fourier analysis using spectral observers," *Proceedings of the IEEE*, vol. 68, no. 2, February 1980.

[3] Gábor Péceli, "A common structure for recursive discrete transforms," *IEEE Trans. on Circuits and Systems*, vol. CAS-33, no. 10, October 1986.

[4] Ferenc Nagy, "Measurement of signal parameters using nonlinear observers," *IEEE Trans. on Instrumentation and Measurement*, vol. IM-41, pp. 152–155, February 1992.

[5] Pál Rózsa, *Linear Algebra and its Applications (in Hungarian)*, Tankönyvkiadó, Budapest, 1991.

[6] Jim Hefferon, *Linear Algebra*, Mathematics Department of Saint Michael's College in Colchester, Vermont, USA, 2001.