A Low-Cost Algorithm to Find the Minimum Sampling Frequency for Multiple Bandpass Signals

Subhonmesh Bose, *Student Member, IEEE*, Varun Khaitan, *Student Member, IEEE*, and Ajit Chaturvedi, *Senior Member, IEEE*

***Abstract*—We consider the problem of finding the minimum sampling frequency required for N non-overlapping, bandpass signals. Recently a novel algorithm with a significantly reduced computational cost has been proposed for this problem. By exploiting a redundancy in this algorithm, we propose a method which further reduces the cost significantly. We use the fact that a valid sampling frequency for a set of bandpass signals must be a valid sampling frequency for any subset of the signals. Several examples are given to illustrate the savings in the computational cost achieved.**

***Index Terms*—Bandpass sampling, software defined radio.**

I. INTRODUCTION

BANDPASS sampling is a technique by which we can down-convert bandpass signals to the baseband without the use of analog mixers. It is widely used for multiple, non-overlapping, bandpass signals without requiring high-quality filters at high frequencies to retrieve the information in the bands. It finds applications in several areas such as the software defined radio (SDR).

The basic principles and techniques for bandpass sampling can be found in [3], [4], and [6]. Techniques for finding valid sampling frequencies have been proposed in [1] and [2]. The case of simultaneous sampling of two bandpass signals has been dealt with in [2], and it has been extended to the general case of N signals in [5]. In [5], all the $2^N \times N!$ possible arrangements of the sampled signals in the baseband has been discussed whereas [1] describes an algorithm for a specific arrangement of the signals. In [1], a guard-band has also been considered to separate the down-converted signals in the baseband.

In this letter, we build upon the novel algorithm for bandpass sampling (ABS) proposed in [1] to further reduce the computational cost. We use the fact that a valid sampling frequency for a set of bandpass signals must be a valid sampling frequency for any subset of the signals. This reduces the search space considerably.

In Section II, we introduce the notation and give the problem statement. In Section III, we derive the necessary conditions for the algorithm. We give the algorithm in Section IV and analytically prove its computational efficiency in Section V. We compare our results with ABS in Section VI and finally conclude the letter in Section VII.

II. PRELIMINARIES

As in [1], let the N signals be denoted as $f_k(t)$ for $k = 1, 2, \ldots, N$ and f_{L_k}, f_{U_k} represent the lower and the upper frequency limits, respectively, of $f_k(t)$. Thus, the bandwidth BW_k for $f_k(t)$ is given by

$$BW_k = f_{U_k} - f_{L_k}.$$

We assume that there is no overlap in the signals and the signals are ordered such that

$$f_{L_1} < f_{U_1} < f_{L_2} < \cdots < f_{L_N} < f_{U_N}.$$

We sample the composite signal with a sampling frequency f_s. Our motive is to find the minimum value of f_s such that the N signals are down-converted to the baseband without aliasing.

III. PROPOSED METHOD

Let GB denote the width of the guardband between successive signals in the baseband and $r_k = \lfloor f_{L_k}/f_s \rfloor$, where $\lfloor\ \rfloor$ denotes the floor function. As in [1], f_s satisfies the following two conditions:

$$f_s \geq 2\left[\sum_{k=1}^{N} BW_k + (N+1)GB\right] \tag{1}$$

and

$$\frac{f_{U_N} + GB}{r_N + 0.5} \leq f_s \leq \min\{f_{UB_{0,1}}, f_{UB_{1,2}}, \ldots, f_{UB_{N-1,N}}\} \tag{2}$$

where

$$f_{UB_{j,j+1}} = \frac{f_{L_{j+1}} - f_{U_j} - GB}{r_{j+1} - r_j} \text{ for } 0 \leq j \leq N-1. \tag{3}$$

For convenience of notation, we take $f_{U_0} = f_{L_0} = 0$ and $r_0 = 0$.

Now we observe that the value of f_s for N signals will also be a valid sampling rate for all $k < N$ signals. In other words, if f_s fails to be a valid sampling rate for any $k < N$ signals, it cannot qualify to be a valid sampling rate for the N signals. This observation is the fundamental difference between our algorithm and ABS. Thus, replacing N by k in (2), we get

Manuscript received February 13, 2008; revised July 02, 2008. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Benoit Champagne.

The authors are with the Department of Electrical Engineering, Indian Institute of Technology, Kanpur, India (e-mail: smbose@iitk.ac.in; varunkh@iitk.ac.in; akc@iitk.ac.in).

Digital Object Identifier 10.1109/LSP.2008.2005446

$$\frac{f_{U_k}+GB}{r_k+0.5} \le f_s \le \min\{f_{UB_{0,1}}, f_{UB_{1,2}}, \ldots, f_{UB_{k-1,k}}\} \tag{4}$$

for $k = 1, 2, \ldots N-1$.

We note that in the RHS of (4), $f_{UB_{k-1,k}}$ contains the term r_k and none of the preceding $f_{UB_{0,1}}, f_{UB_{1,2}}, \ldots f_{UB_{k-2,k-1}}$ contains it. Thus, we need to treat these two cases separately as the LHS also contains a term involving r_k. Hence, from (3) and (4), we get the following two inequalities for $k = 1, 2, \ldots, N-1$:

$$\frac{f_{U_k}+GB}{r_k+0.5} \le f_{UB_{k-1,k}} = \frac{f_{L_k} - f_{U_{k-1}} - GB}{r_k - r_{k-1}} \tag{5}$$

and

$$\frac{f_{U_k}+GB}{r_k+0.5} \le m_k \tag{6}$$

where

$$m_k = \min\{f_{UB_{0,1}}, \ldots, f_{UB_{k-2,k-1}}\} \text{ for } 2 \le k \le N.$$

From (5), it follows that

$$r_k \le \left\lfloor \frac{r_{k-1}(f_{U_k}+GB) + 0.5(f_{L_k} - f_{U_{k-1}} - GB)}{f_{U_k} - f_{L_k} + f_{U_{k-1}} + 2GB} \right\rfloor. \tag{7}$$

Thus, (7) gives us the upper bound for r_k. Further, from (6), it follows that

$$r_k \ge \left\lceil \frac{f_{U_k}+GB}{m_k} - 0.5 \right\rceil \text{ for } 2 \le k \le N \tag{8}$$

where $\lceil\ \rceil$ denotes the ceiling function. This gives us the lower bound for r_k.

The problem can now be rephrased as that of finding the N-tuple of positive integers $\{r_1, r_2, \ldots, r_N\}$ which would satisfy the constraints as given in (7) and (8) which define a valid range for each r_k. Let the lower limit of this range be denoted as $r_{k_{\min}}$ and the upper limit as $r_{k_{\max}}$. However, having chosen $r_1, r_2, \ldots, r_{k-1}$, the computed $r_{k_{\max}}$ might turn out to be less than $r_{k_{\min}}$ which implies that our choice of $r_1, r_2, \ldots, r_{k-1}$ is not a valid one. Thus, we need to iterate on each r_k and construct a valid tuple $\{r_1, r_2, \ldots, r_N\}$. Further, it can be seen from (2) that r_N is in the denominator, and hence to minimize f_s, we need to choose r_N as large as possible. It is clear from (7) and (8) that the range of r_k depends on r_{k-1} and hence by induction on $r_1, r_2, \ldots, r_{k-1}$. Thus, to choose maximum possible r_N, we need to choose maximum possible $r_k\ \forall\ k = 1, 2, \ldots, N-1$. Thus we compute the minimum f_s denoted as $f_{s_{\min}}$ from

$$f_{s_{\min}} = \frac{f_{U_N}+GB}{r_{N_{\max}}+0.5}. \tag{9}$$

We note that (8) does not give us $r_{1_{\min}}$. Since r_1 is a positive integer and there are no constraints on $r_{1_{\min}}$, we choose $r_{1_{\min}}$ to be unity. Further, we note that the upper bound given by (7) for r_1 does not take into account the presence of $f_2(t), f_3(t), \ldots, f_N(t)$. However, in [1], the upper bound for r_1 is derived from the boundary condition taking into account all the signals. Hence, that bound is tighter. Therefore, we choose the upper bound on r_1 as in [1], i.e.,

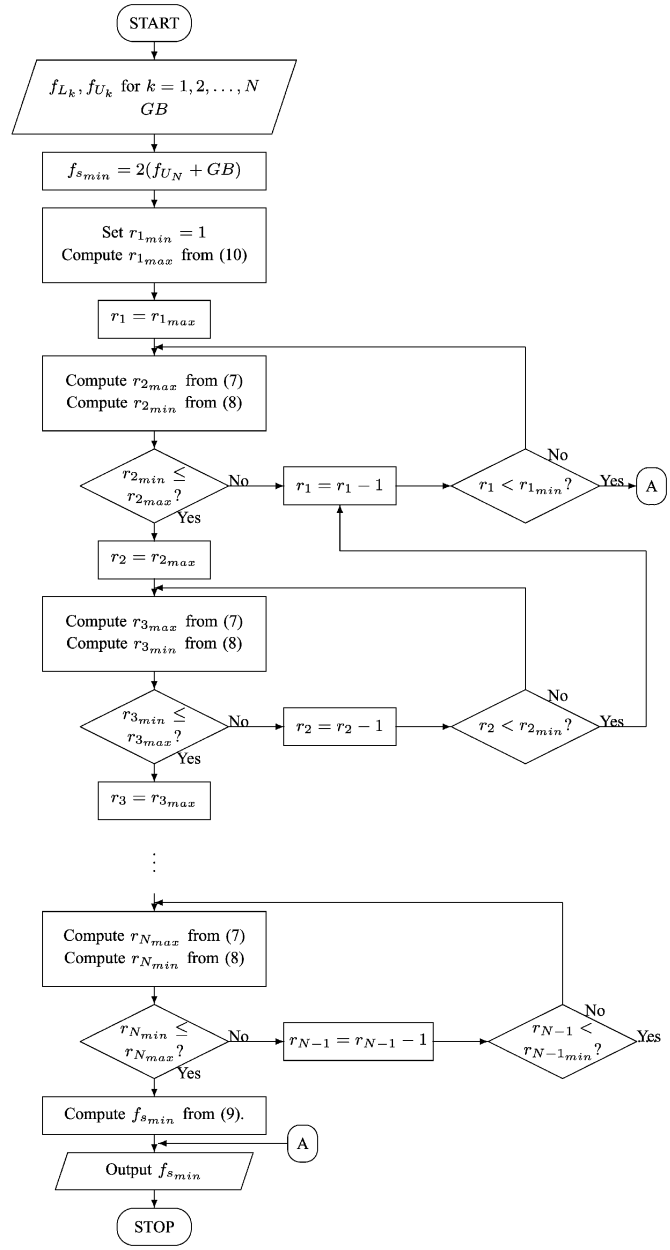

Fig. 1. Flowchart for the proposed algorithm.

$$r_{1_{\max}} = \left\lfloor \frac{f_{L_1} - GB}{2\left[\sum_{k=1}^{N} BW_k + (N+1)GB\right]} \right\rfloor. \tag{10}$$

In Section IV, we enumerate the steps of the proposed algorithm to compute $f_{s_{\min}}$. The flowchart of the algorithm is given in Fig. 1.

IV. ALGORITHM

1) Input the frequencies f_{L_k} and f_{U_k} for $k = 1, 2, \ldots, N$.

2) Input the width of the guard-band GB.

3) Initialize $f_{s_{\min}}$ to the Nyquist rate $2(f_{U_N} + GB)$.

4) Set $r_{1_{\min}}$ to 1 and calculate $r_{1_{\max}}$ from (10).

5) Set r_1 to $r_{1_{\max}}$.

6) Compute RHS of (7) and set it to $r_{2_{\max}}$ and compute the RHS of (8) and set it to $r_{2_{\min}}$.

7) If $r_{2_{\max}} < r_{2_{\min}}$, decrease r_1 by 1.

8) If $r_1 < r_{1_{\min}}$, go to step 16; else go back to step 6.

9) Set r_2 to $r_{2_{\max}}$.

10) Compute RHS of (7) and set it to $r_{3_{\max}}$ and compute the RHS of (8) and set it to $r_{3_{\min}}$.

11) If $r_{3_{\max}} < r_{3_{\min}}$, decrease r_2 by 1.

12) If $r_2 < r_{2_{\min}}$, decrease r_1 by 1 and go to step 6; else go to step 10.

13) Continue the process similarly until we get a valid set of $r_1, r_2, \ldots, r_{N-1}$.

14) Compute RHS of (7) and set it to $r_{N_{\max}}$ and compute the RHS of (8) and set it to $r_{N_{\min}}$.

15) If $r_{N_{\max}} \geq r_{N_{\min}}$, then calculate $f_{s_{\min}}$ from (9) and go to step 16; else decrease r_{N-1} by 1 and go back to step 13.

16) Output $f_{s_{\min}}$.

17) End.

V. Analysis

Now we prove that our algorithm takes lesser iterations as compared to that in [1]. We start with some preliminaries that we require in the proof. In step 15) of our algorithm, rather than breaking from the loop structure after computing $f_{s_{\min}}$, we simply list all possible N-tuples $(r_1, r_2, \ldots, r_N)$ that satisfy the conditions given by (7), (8), and (10). Let us denote each such N-dimensional vector $(r_1, r_2, \ldots, r_N)$ by R_N and the set of all possible R_N's as S_N. Each $R_N \in S_N$ gives a range of sampling frequencies as

$$\left[\frac{f_{U_N} + GB}{r_N + 0.5}, \min\{f_{UB_{0,1}}, \ldots, f_{UB_{N-1,N}}\}\right]. \tag{11}$$

We suitably generalize ABS to give us all possible R_N's. We note that these modifications to both the algorithms do not affect the computational efficiency of the algorithms but lists all possible R_N's instead of just generating one such R_N corresponding to $f_{s_{\min}}$. We compare the generalized versions of both of these algorithms in the theorem, and the result for the specific case of $f_s = f_{s_{\min}}$ follows trivially. We observe that both of the algorithms give us the same set of R_N's because they use the same theoretical constraints to obtain such R_N's. Given any N signals $f_1(t), f_2(t), \ldots, f_N(t)$, for every $R_N = (r_1, r_2, \ldots, r_N) \in S_N$, let us define $a(R_N)$ as the number of iterations taken by our algorithm to generate R_N and $b(R_N)$ to be the number of iterations taken by ABS to do the same. Now we prove the following theorem by induction.

Theorem 1: For $N > 2$ signals, $a(R_N) \leq b(R_N)$ for every $R_N \in S_N$.

Proof: First we prove it for $N = 2$. We note that the range of r_1 is the same in both of the algorithms. Now given an r_1, our algorithm constructs the set

$$A_2 = \{r_2 : (r_1, r_2) \in S_2\}$$

directly using $k = 2$ in (7) and (8). Now we define

$$B_2 = \{r_2 : r_2 \text{ satisfies (7) of [1]}\}.$$

For each $r_2 \in B_2$, we check the condition given in (2) for $N = 2$. Now we note that for each $r_2 \in A_2$, we get $(r_1, r_2) \in S_2$. But $r_2 \in B_2$ may or may not satisfy (2). If it does, then $(r_1, r_2) \in S_2$ and hence $r_2 \in A_2$. If it does not, then $r_2 \in B_2$ but $r_2 \notin A_2$. But for each $r_2 \in A_2$, r_2 also belongs to B_2, as otherwise it would mean that $R_2 = (r_1, r_2) \in S_2$ is not generated by ABS, which is not possible given the fact that both of the algorithms generate the same set S_2. Hence, $A_2 \subseteq B_2$. Thus, we iterate on a smaller search space to reach each element of S_2 leading to savings in computations and hence $a(R_2) \leq b(R_2)$.

Now let the theorem be true for $N = k$. We add a signal $f_{k+1}(t)$ to the set of signals $f_1(t), f_2(t), \ldots, f_k(t)$ and construct the set S_{k+1}. We note that for all $R_{k+1} = (r_1, r_2, \ldots, r_k, r_{k+1}) \in S_{k+1}$, the corresponding $R_k = (r_1, r_2, \ldots, r_k) \in S_k$. This key observation forms the basis for the difference between the algorithms and leads to the computational efficiency of our approach. Now by hypothesis, $a(R_k) \leq b(R_k)$. Given this R_k, we see how both of the algorithms construct r_{k+1} and hence R_{k+1}. As earlier, let us define

$$A_{k+1} = \{r_{k+1} : (r_1, r_2, \ldots, r_k, r_{k+1}) \in S_{k+1}\} \text{ and}$$
$$B_{k+1} = \{r_{k+1} : r_{k+1} \text{ satisfies (7) in [1]}\}.$$

By the same chain of arguments as in $N = 2$ case, we obtain that for every $r_{k+1} \in A_{k+1}$, we have $r_{k+1} \in B_{k+1}$, but there might exist $r_{k+1} \in B_{k+1}$ that fails to satisfy (2) and consequently $r_{k+1} \notin A_{k+1}$. Thus, $A_{k+1} \subseteq B_{k+1}$, and hence, $a(R_{k+1}) \leq b(R_{k+1})$. Hence, it is proved. ∎

For any set of signals, S_N comprises of all possible R_N's that correspond to a range of sampling frequencies as given by (11). Now the theorem proves that for each R_N in S_N, the modified version of our algorithm reaches R_N in equal or lesser number of steps than the modified version of ABS. Let us denote the R_N in S_N that corresponds to $f_s = f_{s_{\min}}$ by R'. From the theorem, it follows that $a(R') \leq b(R')$. However, since the algorithms in their original form only compute $f_{s_{\min}}$, they only generate R'. Hence, $a(R') \leq b(R')$ guarantees us that the proposed algorithm performs the task in less or equal number of iterations than ABS. For equality to hold, i.e., $a(R') = b(R')$, we must have the search space of both the algorithms to be the same. In other words, we require $A_k = B_k$ for $k = 2, 3, \ldots, N$.

Now we illustrate through an example how our algorithm achieves the computational efficiency as compared to ABS. Suppose we have $N = 3$ signals and we have already chosen r_1. Now both of the algorithms choose a value for r_2. The value of r_2 as chosen by ABS may be such that $r_2 \in B_2$ but $r_2 \notin A_2$. But this (r_1, r_2) cannot yield an r_3 for which $(r_1, r_2, r_3) \in S_3$. But ABS iterates over each $r_3 \in B_3$ with this pair of (r_1, r_2) for which we already know (r_1, r_2, r_3) cannot

TABLE I
PERFORMANCE COMPARISON

Sl. no.	No of signals (N)	Frequencies MHz	GB MHz	$f_{s_{min}}$ MHz	No of iterations [1]	No of iterations Proposed
1	3	$f_1 \in (864.2, 864.4)$,$f_2 \in (890.2, 890.4)$,$f_3 \in (935.6, 935.8)$	0	1.6093	422	230
2	3	$f_1 \in (864.2, 864.4)$,$f_2 \in (890.2, 890.4)$,$f_3 \in (935.6, 935.8)$	0.2	3.20	89	53
3	4	$f_1 \in (434.3, 434.6)$,$f_2 \in (654.2, 654.6)$ $f_3 \in (891.4, 891.5)$,$f_4 \in (958.5, 958.7)$	0.1	16.6748	894	185
4	4	$f_1 \in (434.3, 434.6)$,$f_2 \in (654.2, 654.6)$ $f_3 \in (891.4, 891.5)$,$f_4 \in (958.5, 958.7)$	0	16.6730	1446	301
5	5	$f_1 \in (156.5, 156.7)$,$f_2 \in (348.2, 348.4)$,$f_3 \in (476.3, 476.6)$ $f_4 \in (762.3, 762.5)$,$f_5 \in (943.2, 943.5)$	0.1	1877.20	3208	93
6	5	$f_1 \in (256.5, 256.7)$,$f_2 \in (348.2, 348.4)$,$f_3 \in (476.3, 476.6)$ $f_4 \in (762.3, 762.5)$,$f_5 \in (943.2, 943.5)$	0	7.5179	1668	121

belong to S_3. On the contrary, our algorithm generates only those r_2 such that $(r_1, r_2) \in S_2$. Thus, we save on iterations. Furthermore, to find the minimum sampling frequency for three signals, we do not iterate on r_3. Rather, A_3 is computed directly. If the bounds given by (7) and (8) for $k = 3$ are such that $r_{3_{\min}} > r_{3_{\max}}$, then $A_3 = \phi$, and thus, the given (r_1, r_2) cannot have a corresponding value of r_3. If $A_3 \neq \phi$, then $r_{3_{\max}}$ gives us $R_3 = (r_1, r_2, r_{3_{\max}}) \in S_3$. However, ABS does not take into account this fact and iterates over r_3. Thus, we save on the iterations on r_3.

VI. Numerical Results

In this section, we compare the performance of our algorithm with ABS. Consider the three bandpass signals as given in serial number 1 in Table I with carrier frequencies 864.3 MHz, 890.3 MHz, 935.7 MHz, and $BW_1 = BW_2 = BW_3 = 0.2$ MHz. With $GB = 0$ MHz, our algorithm gives $r_1 = 537$, $r_2 = 553$, $r_3 = 581$, and $f_{s_{\min}} = 1.6093$ MHz. As expected, these values are identical to the values obtained in [1]. However, the number of iterations taken by our algorithm is only 230 while ABS needs 422 iterations. For the same case when GB is changed to 0.2 MHz as given at serial number 2 in Table I, we obtain $r_1 = 270$, $r_2 = 278$, $r_3 = 292$, and $f_{s_{\min}} = 3.20$ MHz. In this case, our algorithm gives the result in 53 iterations as compared to 89 iterations taken by ABS.

In Table I, we illustrate four more cases to compare the computational efficiency of the proposed algorithm with ABS. In each case, as expected, the values obtained for $f_{s_{\min}}$ are the same for both the algorithms. However, the proposed algorithm needs fewer iterations. Thus, for example, at serial number four in Table I, it can be seen that the number of iterations needed by ABS is 3208 while the proposed algorithm needs only 93 iterations that is a reduction by more than a factor of 30. Similarly we can see the savings for the other cases reported in Table I. The savings obtained depends on the frequencies of the individual signals as well as on the guard-band and vary from case to case. Since the cost of each step involved in the proposed algorithm is the same as that of ABS, the savings in the number of iterations translates directly into lower computational cost of the proposed algorithm.

VII. Conclusion

In this letter, we have proposed an algorithm for finding the minimum sampling frequency for N non-overlapping bandpass signals. The proposed algorithm provides significant savings on the computational cost of existing algorithms by exploiting the fact that a frequency is a valid sampling frequency for N signals, only if it is a valid sampling frequency for a subset of the signals.

References

[1] J. Bae and J. Park, "An efficient algorithm for bandpass sampling of multiple RF signals," *IEEE Signal Process. Lett.*, vol. 13, no. 4, pp. 193–196, Apr. 2006.

[2] C. H. Tseng and S. C. Chou, "Direct downconversion of multiple RF signals using bandpass sampling," in *Proc. ICC*, May 2003, vol. 3, pp. 2003–2007.

[3] R. G. Vaughan, N. L. Scott, and D. R. White, "The theory of bandpass sampling," *IEEE Trans. Signal Process.*, vol. 39, no. 9, pp. 1973–1983, Sep. 1991.

[4] M. Choe and K. Kim, "Bandpass sampling algorithm with normal and inverse placements for multiple RF signals," *IEICE Trans. Commun.*, vol. E88, no. 2, pp. 754–757, Feb. 2005.

[5] C.-H. Tseng and S.-C. Chou, "Direct downconversion of multiband RF signals using bandpass sampling," *IEEE Trans. Wireless Commun.*, vol. 5, no. 1, pp. 72–76, Jan. 2006.

[6] A. Mahajan, M. Agarwal, and A. K. Chaturvedi, "A novel method for down-conversion of multiple bandpass signals," *IEEE Trans. Wireless Commun.*, vol. 5, no. 2, pp. 427–434, Feb. 2006.