Incorporating Application-Level Fault Tolerance and Detection into Radar Angular Super-Resolution

Hua Yang,
Beijing Institute of Control Engineering

Israel Koren and C.M. Krishna
Electrical and Computer Engineering Dept.
University of Massachusetts, Amherst

The radar application we discuss seeks to compute a spatial angular spectrum based on a set of narrow-band signals from a variety of directions. An array of sensors collect the target signals, and frames of input data from the sensors are distributed to a set of parallel processes. Each computing node processes one frame of data and computes the corresponding spatial angular spectrum. Unless fault tolerance is provided, should a process fail - we lose the frame that it is currently working on. Instead of using massive processor redundancy which may impose computational overheads of 100% or more, we explore application-level fault-tolerance (ALFT) which exploits inherent redundancy, in the form of data correlations, and creates a lightweight, yet highly effective, alternative to hardware redundancy. We have previously shown the applicability of ALFT in space-telescope [7], target-tracking [3], and orbiting spectrometer [1] applications. Here we demonstrate that ALFT is a practical approach in the radar application.

Two algorithms to estimate the application parameters were developed by Helme and Nikias [5] and Marple [8].

Faulty input data (due to bit flips) can have a significant effect on the computed spectrum. Faulty computed data can be detected by either of the following:
(1) No power spectrum output may be greater than 0 dB.
(2) Data should only change gradually - sudden changes in the peak angle are suspicious.

ALFT operates as follows: Each process is "shadowed" by a secondary process. We check the primary process output for questionable data, and if suspect, run the corresponding secondary process and use its output instead. As opposed to traditional redundancy where the secondary has the same complexity as the primary, the secondary in ALFT has a greatly decreased complexity, at the cost of lower output accuracy. We suggest three different approaches to reducing the secondary complexity. The metric we use for output inaccuracy is the *relative bias*, denoted by R_b, which is the relative error in the peak angle:

$$R_b = \left| \frac{\text{computed peak angle} - \text{actual peak angle}}{\text{actual peak angle}} \right|$$

We used simulation and analyzed two cases - two targets and four targets. Figure 1 shows how ALFT improves the fault spectrum output in the two-target case with a bit flip probability of 2%.

In our experiments we compared the following three approaches for constructing the secondary:
1. The first approach uses the entire data set and the same algorithm but a lower recursion order. Figure 2 shows the relationship between the recursion order (expressed as a percentage of its corresponding primary) and the relative bias. Figure 3 shows the overhead in terms of the execution time as a function of the recursion order. We see that for the two-target case, maintaining the computing relative bias at less than 0.05 requires the secondary to be at least 28% of the primary, resulting in a secondary overhead of 35%.

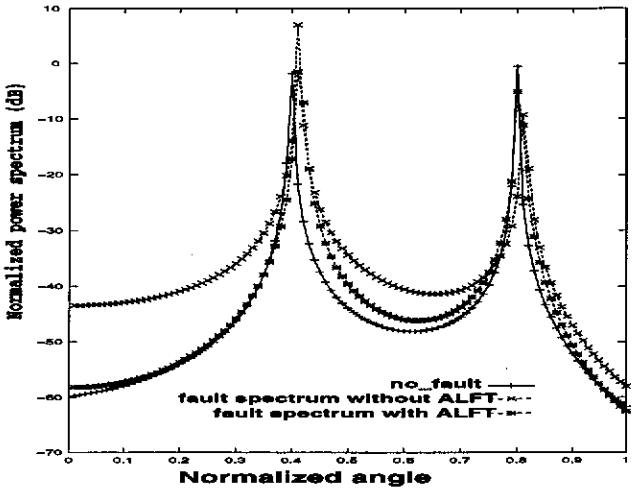

Fig. 1. Angular power spectrum with and without ALFT

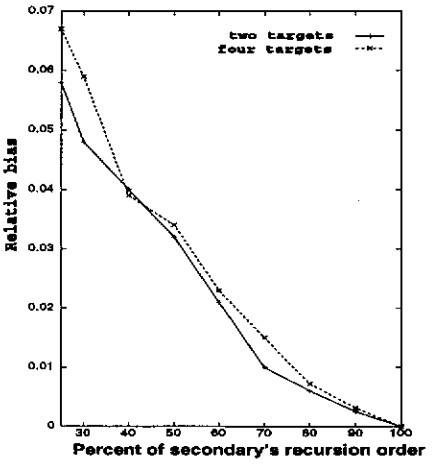

Fig. 2. Relative bias vs. percent of secondary's recursion order

2. The second approach uses a simpler (lower-quality) algorithm, with the entire data set. Adopting the Marple

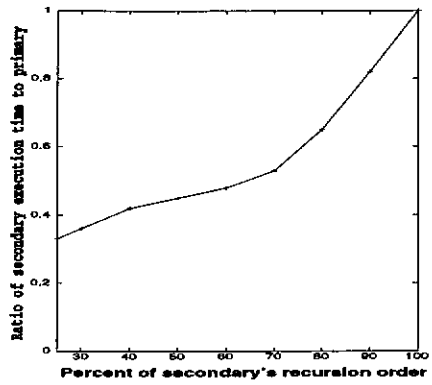

Fig. 3. Ratio of secondary execution time to primary vs. percent of secondary's recursion order

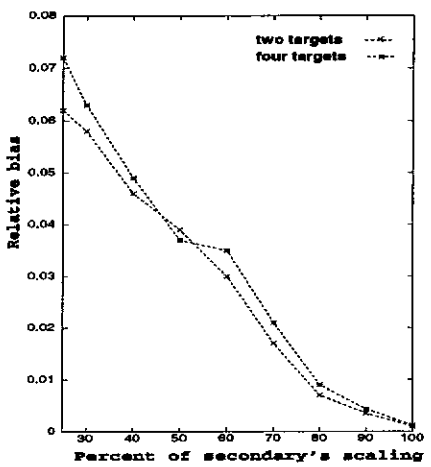

Fig. 4. Relative bias vs. percent of secondary scale

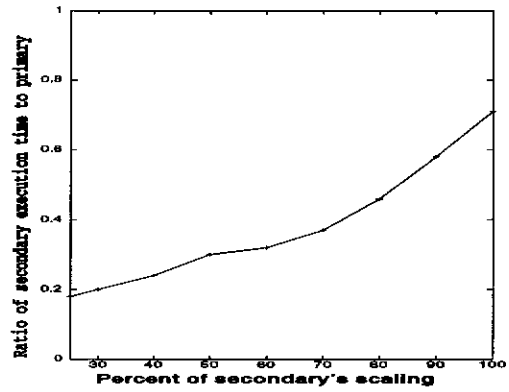

Fig. 5. Ratio of secondary execution time to primary vs. percent of secondary scale

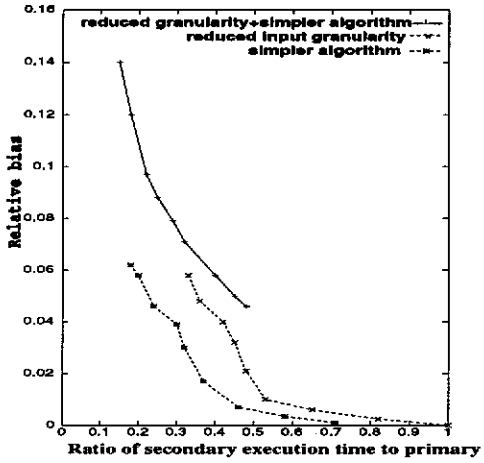

Fig. 6. Comparing secondary approaches: two targets

algorithm for the primary and the Helme-Nikias algorithm for the secondary, we obtain the results shown in Figures 4 and 5, for the relative bias and overhead, respectively. To achieve a relative bias no larger than 0.05, the computational complexity of the secondary must be at least 40% of the primary, which results in a 23% secondary overhead.

3. The third approach is to combine the first and second schemes, i.e., use a simpler algorithm with only part of the data set. In this case, In order to achieve a relative bias smaller than 0.05, the secondary data overlap must be over 90% requiring a secondary overhead of more than 40% of the primary.

Figure 6 compares the performance of the three approaches as a function of the secondary process overhead for two targets case. The second approach (based on the use of the Helme-Nikias algorithm) outperforms the other two approaches for all values of the relative bias. This algorithm, when used for the secondary process, can achieve a relative bias of 0.05 or less for a secondary overhead which is no larger than 25% of the primary's execution time - a reasonable overhead to pay for the fault tolerance provided.

REFERENCES

[1] E. C. Ciocca, *Application-Level Fault Tolerance and Detection*, M.S. Thesis, University of Massachusetts, Amherst, 2001.

[2] F. Cristian, "Understanding Fault-tolerant Distributed Systems," *Communications of the ACM*, vol. 34, No. 2, 1991, pp. 56-78.

[3] J. Haines, V. Lakamraju, I. Koren, C.M. Krishna, "Application-Level Fault Tolerance as a Complement to System-Level Fault Tolerance," *Journal of Supercomputing*, Vol. 16, pp. 53-68, 2000.

[4] R. Harper, J. Lala and J. Deyst, "Fault Tolerant Parallel Processor Architecture Overview," *Int. Conf. Fault-Tolerant Computing, FTCS-18*, 1988, pp. 252–257.

[5] B. Helme and G.L. Nikias, "Improved Spectrum Performance via a Data Adaptive Weighed Burg Technique," *IEEE Trans. Acoustics Speech and Signal Processing*, Vol. ASSP-33, No.9, August 1985.

[6] K.H. Huang and J.A. Abraham "Algorithm-Based Fault Tolerance for Matrix Operations," *IEEE Trans. Computers*, Vol. C-33, No. 6, 1984, pp. 518–528.

[7] J. K. Nair, Z. Koren, I. Koren, and C. M. Krishna, "Preprocessing Input Data to Augment Fault Tolerance in Space Applications," *International Performance and Dependability Symposium*, pp. 491-500, June 2003.

[8] L. Marple, "A New Autoregressive Spectrum Analysis Algorithm," *IEEE Trans. Acoustics, Speech and Signal Processing*, vol. ASSP-28, No.4. August 1980.