논문 2007-44SP-4-13

다중 DSP 프로세서 기반의 병렬 수중정합장처리 알고리즘 설계

(Design of Parallel Algorithms for Conventional Matched-Field Processing over Array of DSP Processors)

김 건 욱*

(Keonwook Kim)

요 약

고성능 네트워크와 분산처리구조가 병렬처리와 함께 결합되면, 전체적인 디지털 신호처리 시스템의 계산능력, 신뢰도, 다양성을 향상시킨다. 본 논문에서는, 발전된 형태의 수중레이더 (sonar) 알고리즘인 수중정합장처리 (Matched-Field Processing; MFP)를 위한 병렬처리 알고리즘을 디자인하고 다중 DSP 프로세서 기반의 병렬처리 시스템 상에서 성능분석과 함께 최적의 병렬처리 솔루션을 제안한다. 각각의 병렬 알고리즘은 특정한 도메인에서 주어진 계산량을 분산시키며 이를 통한 속도향상을 추구한다. 필요한 연산량과 형태에 따라서 병렬 알고리즘은 각기 다른 성능향상을 보여준다. 또한, 알고리즘의 계산량 분산방식, 프로세서간의 통신방식, 알고리즘의 복잡도, 프로세서의 속도, 목적하는 시스템의 구성에 따라서 다양한 성능지표를 보여준다. 제안하는 주파수와 출력값 기반의 병렬 알고리즘은 상당한 계산량을 요구하는 수중정합처리 알고리즘을 적절히 다중 프로세서에 균형 있게 분산시켜 프로세서의 개수와 비례하는 성능향상을 보여주고 있다.

Abstract

Parallel processing algorithms, coupled with advanced networking and distributed computing architectures, improve the overall computational performance, dependability, and versatility of a digital signal processing system. In this paper, novel parallel algorithms are introduced and investigated for advanced sonar algorithm, conventional matched-field processing (CMFP). Based on a specific domain, each parallel algorithm decomposes the sequential workload in order to obtain scalable parallel speedup. Depending on the processing requirement of the algorithm, the computational performance of the parallel algorithm reveals different characteristics. The high-complexity algorithm, CMFP shows scalable parallel performance on the array of DSP processors. The impact on parallel performance due to workload balancing, communication scheme, algorithm complexity, processor speed, network performance, and testbed configuration is explored.

Keywords : Parallel Processing, Match-Field Processing, DSP Processor Array, Beamforming, SONAR

I. Introduction

During a long period of time, plane-wave beamforming algorithms have been pervasive as techniques to find the Direction of Arrival (DOA) of the target for underwater signal processing. However, under certain situations such as shallow water, plane-wave beamforming increases the likelihood to miss the target or produce misleading target locations because of the mismatch problem of the data model. The simple phased model of plane-wave beamforming is inappropriate to describe the underwater wave propagation. Due to the limited performance of the plane-wave beamforming algorithms, Matched-Field Processing (MFP) based on a more realistic model was introduced to enhance the capability of sonar signal processing. In the new

* 평생회원, 동국대학교 전자공학과
(Electronics Engineering, Dongguk University)
접수일자: 2007년3월19일, 수정완료일: 2007년6월11일

model, the waveguide nature of the ocean and indirect paths between the target and the sensor are considered to describe wave propagation reliably.

These advanced beamforming algorithms from a complex data model exhibit high levels of computational complexity and memory utilization. These hurdles make implementation in real-time sonar array systems a significant challenge. Because of the computational resources necessary to carry out the MFP, it is appropriate to combine these signal processing efforts with modern data telemetry and networking technologies. Taken together, these trends make imperative the development and use of advanced distributed and parallel processing techniques in terms of algorithm, architecture, network, and system design to reduce the significant processing delay of the MFP algorithms.

The MFP is a generalization of plane-wave beamforming in which the steering vectors are derived from the spatial point source response of the medium. The Conventional Matched-Field Processing (CMFP) algorithm was first proposed by Bucker[1] and first implemented by Fizell[2]. To determine the location of a source, Bucker's MFP algorithm compares the measured pressure of signals at a receiver to the theoretical transmissions at the source. By contrast, with plane-wave beamforming the image of a point source is reconstructed on the basis of simple time delay; such a model is inadequate in an ocean waveguide. In ocean acoustics, the reflection of sound at the boundaries, represented as a number of discrete arrivals, is measured at the far field of an acoustic source because of waveguide propagation. In the MFP algorithm, the target location is determined by matching the measured acoustic pressure field at the sensor outputs to a predicted pressure field, which is based on an assumed source location. The choice of a suitable acoustic propagation model determines the predicted acoustic field at the sensors. The objective of the propagation model is to examine which analytical waveguide model corresponds to the real waveguide, along with the actual conceptual feasibility of localization of a source in a real waveguide. The experimental possibility of reconstructing and localizing a source in a waveguide, which is described by an analytical model, is confirmed in many papers such as[3].

The CMFP algorithm requires both significant processing power, as well as memory demands that supercede the capabilities of most current real-time uniprocessor systems. To cope with the computational and storage challenges of the CMFP, efficient distributed parallel algorithms for the CMFP need to be developed. Such parallel algorithms might be expected to provide a feasible solution to real-time, deployable, and cost-effective beamforming by distributing a workload over an array of processors.

To demonstrate the feasibility of the algorithms being developed, a series of parallel experiments are performed on an array of digital signal processors (DSPs). The array employs efficient interprocessor communication using a service known as MPI-SHARC[4], to accelerate complex collective communication. Performance results from the array will be provided in this paper.

Section II presents the theoretical background of the CMFP algorithm used as a basis for this study, and Section III is dedicated to the explanation of two novel parallel algorithms for the CMFP. Experimental testbeds used to analyze the parallel algorithms are described in Section IV. Section V then explores and examines the performance of the parallel CMFP algorithms in terms of the speedup. Finally, Section VI provides the brief conclusion of this paper.

II. CMFP Algorithm

The MFP localizes a source more accurately than plane-wave methods; in particular, range and depths can be estimated well since they incorporate the full wave physics of the acoustics, including both signal and noise processes, into the array processing. For precise target localization, the propagation model must be accurate; otherwise, the performance of the

MFP will suffer from the significant mismatch problems described by Baggeroer et al.[5]. The calculation of the predicted acoustic field is far more difficult than plane-wave beamforming due to the more realistic propagating effect and the need for solving the wave equation given by Equation 1

$$\nabla^2 p(r,z)+\frac{\omega^2}{c^2(r,z)}p(r,z)=\frac{-\delta(r-r_s)\delta(z-z_s)}{r} \qquad (1)$$

Here, ∇^2 represents second-order partial derivatives, $p(r,z)$ is the pressure at range r and depth z, $c(r,z)$ is the ocean sound speed, ω is the angular frequency of the source located at range r_s and depth z_s, and $\delta(x-x_o)$ is the delta function, also known as the unit impulse function. The derivation and physical translation of the wave equations are illustrated in[6]. By solving the wave equation, the acoustic pressure $p(r,z)$ is obtained as a function of range and depth for the steering vector of the MFP algorithm.

The solution of the wave equation depends on the acoustic models applied. There is no single closed-form solution of the given wave equation due to the initial conditions and the boundary conditions. In the acoustic model, we assume a certain form of solution and wave propagation to meet the conditions. Numerous acoustic models have been proposed to solve the wave equation. Although one model may be able to handle most of the situations encountered, at least some of the cases are usually more efficiently treated by another model. Computational pattern and complexity are quite diverse for each model as well.

When the signal path can be modeled as a simple waveguide, the normal-mode solution is efficient in terms of model performance and computation. The pressure field at any point in the waveguide can be represented as a sum of vertical standing waves when one source excites the waveguide. The main limitation to the normal-mode solution is that the solution is considered principally in the context of range-invariant conditions. However, the normal-mode solution can be extended in various ways to both range-dependent problems and fully three-dimensional problems[7]. The normal-mode solution begins with the assumption that the solution of the wave equation consists of the product of range- and depth-dependent terms. By separation of variables, the solution of the wave equation is provided as follows:

$$p(r,z)\approx\frac{i}{\sqrt{8\pi r}}e^{\frac{-i\pi}{4}}\sum_{m=1}^{\infty}\Psi_m(z_s)\Psi_m(z)\frac{e^{ik_m r}}{\sqrt{k_m}} \qquad (2)$$

In the equation, z_s is the source depth, r is the range between source and receiver, k_m^2 is a separation constant where k_m is the horizontal wave number for the mth mode, and $\Psi_m(z)$ is the normal-mode depth function corresponding to k_m^2. The horizontal wave number and normal-mode depth function are calculated from the eigenvalue problem with environmental parameters. The ambient environment data alter the number of parameters and parameter values on the eigenvalue problem. The accuracy of environment information, therefore, is very important. Many existing methods are used for the normal-mode solution. The most widely used method is KRAKEN, which is based on a finite-different algorithm. The popularity of KRAKEN originates from the matrix that is set up as a tri-diagonal matrix for which very fast eigenvalue and eigenvector solution techniques are available. The work herein is based on the KRAKEN model developed by Porter and described in[7~9]. Figure 1 shows the sample normal-mode intensity plot from the Porter code[10] for a wide underwater area with a 1000m-depth source. This figure clearly illustrates the complex-wave propagation in the ocean.

The CMFP is the most fundamental form of the MFP algorithm. Advanced techniques, such as adaptive methods, can be applied in order to improve the MFP output. The CMFP has no computational difference with conventional beamforming[11] after the steering vectors have been computed. A match is performed by testing a set of modeled steering vectors and comparing them to the Cross-Specral Matrix (CSM) formed by the signal received at the sensor array. There are generally four important

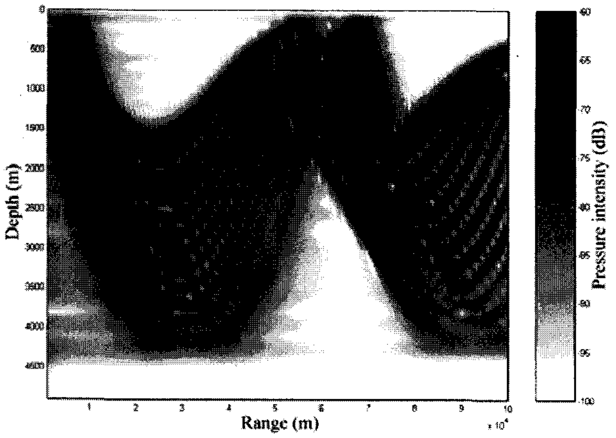


그림 1. 1000m 위치한 음원에 대한 Normal-mode 전파[10]

Fig. 1. Normal-mode propagation with a source at 1000m depth[10].

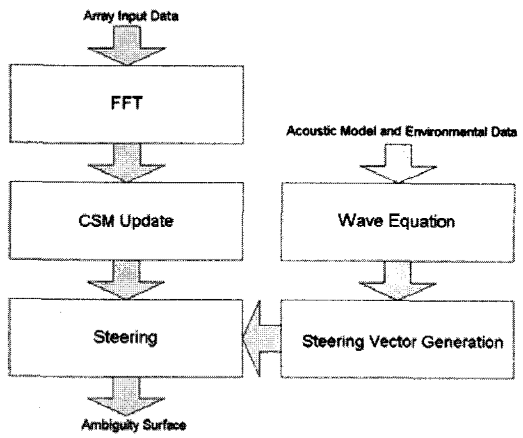


그림 2. CMFP 블록도

Fig. 2. CMFP block diagram.

computational stages: Fast Fourier Transform (FFT), CSM update, steering, and steering vector generation. The FFT and CSM update stage are identical with those of conventional beamforming algorithm. The steering procedure estimates the field distribution versus the sensor input. The stage is both a function of the CSM and the steering vectors. The steering vector generation stage uses normal-mode solution and environmental data to solve the wave equation for the spatial response of a source with a given location. Figure 2 illustrates the computational flow of the CMFP.

To summarize the previous discussion, the equation for the CMFP algorithm is provided below.

$$DF(r,z) = G(r,z)^H \cdot C \cdot G(r,z) \tag{3}$$

Here, the $DF(r,z)$ is the detection factor (ambiguity surface) at range r and depth z, C is the CSM and $G(r,z)$ represents the normalized steering vectors for a given range and depth. The output of the CMFP produces a three-dimensional plot of detection factors versus range and depth for specific frequency. For broadband processing, the multiple CMFP results are averaged over a wide range of frequency. Independent steering vectors and the CSM are required for each frequency CMFP result. The final resulting surface is commonly referred to as the ambiguity surface and presents the likeliness of detection for a given data set. Peak locations on the plot represent probable source locations. Figure 3 shows the CMFP ambiguity surface using the normal-mode solution for a source at a 50m depth and 10Km range. The range/depth plane consisted of 5 to 44Km in range and 0 to 158m in depth. The resolution used was 1Km in range and 2m in depth. The frequency resolution used in the Fourier decomposition was 1.0Hz across a band from 200 to 231Hz. The vertical array contained 32 sensors spaced at 4m with the top sensor at 10m depth. The generated data do not contain any noise and no uncertainty exits about the environment data.

The CMFP equation, Equation 3, is almost identical to the conventional beamforming algorithm, except

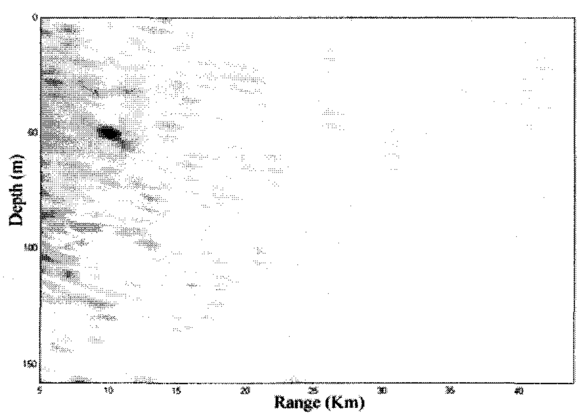


그림 3. 32노드를 갖춘 CMFP의 예제 출력

Fig. 3. Sample output of the CMFP for a 32-node array.

the steering vectors are now a function of range and depth. The computational pattern of both algorithms is similar. However, the required complexity and size of the steering vectors for the CMFP are much larger due to the complicated wave equation solution and more resolvable coordinate of steering vectors, such as range and depth. Therefore, the CMFP requires more processing power and memory capacity. The real computational problem with the CMFP is not only requiring more computation to execute, but also that it requires intensive computation and an extensive amount of storage to calculate and preserve steering vectors in the initial phase.

Ⅲ. Parallel Algorithms for the CMFP

The two parallel algorithms presented in this section make use of decomposition in two different domains: frequency and output points.

1. Frequency Decomposition

The first decomposable space of the CMFP algorithm is the frequency domain. The CMFP algorithm generates multiple frequency results by independent computation between frequency bins. The frequency decomposition algorithm distributes the processing load by decomposing the frequency bins. Each node calculates the CMFP results for a certain number of desired frequency bins from the same sample set. The block diagram illustrating this algorithm in operation is shown in Figure 4 for a 3-node array.

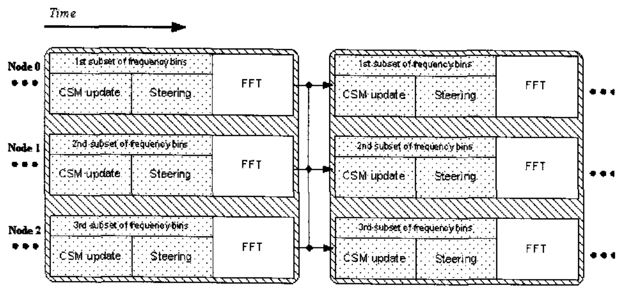


그림 4. 주파수 분할
Fig. 4. Frequency decomposition.

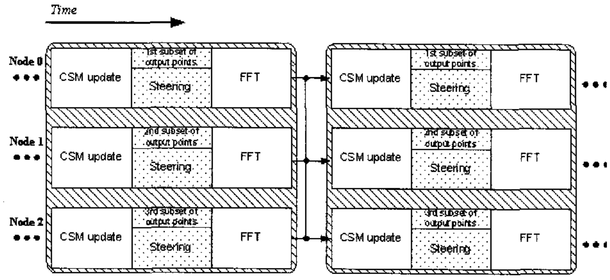


그림 5. 섹션 분할
Fig. 5. Section decomposition.

2. Section Decomposition

The second parallel algorithm decomposes the CMFP using a coarse-grained approach as well. The communication method is identical with the frequency decomposition. The section decomposition distributes the processing load by decomposing another domain, the output points. Each node calculates the CMFP results for a certain subset of output from the same data. Before doing so, all participating nodes must have a copy of the data from all other nodes. After completing this all-to-all communication, each node performs a given workload based on the output points. A block diagram illustrating this algorithm in operation on a 3-node array is shown in Figure 5.

Ⅳ. Experimental Testbed

The testbed has configuration with multiple processing units connected by a communication channel. As for the software, the algorithms were implemented via message-passing parallel programs written in C with the message-passing interface (MPI)[12]. On the DSP array, a single iteration was measured since this testbed provides deterministic time measurement with no transient overhead from an operating system.

The target testbed of this experiment consists of 10 Bittware Blacktip-EX DSP development boards[13] connected to one another in a ring topology, as shown in Figure 6. Each board includes a single ADSP-21062 Super HArvard ARChitecture (SHARC) processor from Analog Devices, as well as additional hardware for links to other nodes, off-chip memory,

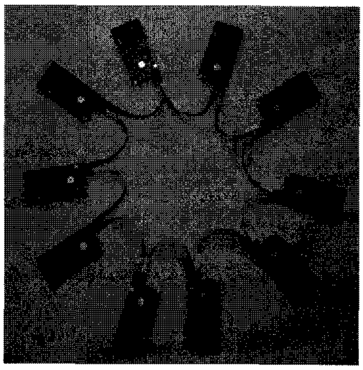


그림 6. 디지털 신호처리 프로세서 어레이
Fig. 6. Digital signal processor array.

and so forth.

The development board contains two link ports with external connectors to enable communications with other devices. To eliminate the need for external routing or switching hardware, the two link ports are dedicated to separate send and receive channels. This configuration allows the boards to be arranged in a uni-directional ring topology.

The network service in this DSP array was implemented and optimized for this particular architecture by a previous research effort[4]. To provide the MPI functionality on an array of DSPs, the MPI-SHARC network service was created. Although the MPI-SHARC is a subset of the full specification, the functionality and syntax are identical to the MPI found on common distributed systems, allowing users to easily port applications developed on other platforms to an embedded, distributed DSP system.

V. Performance Analysis of Parallel CMFP Algorithms

This section explores the performance of the parallel CMFP algorithms on the testbed. The measured parameter includes execution times of sequential and parallel algorithms. Based on the parameter, dependent results can be derived for scaled speedup. Investigation of these results demonstrates the performance effects of the design issues.

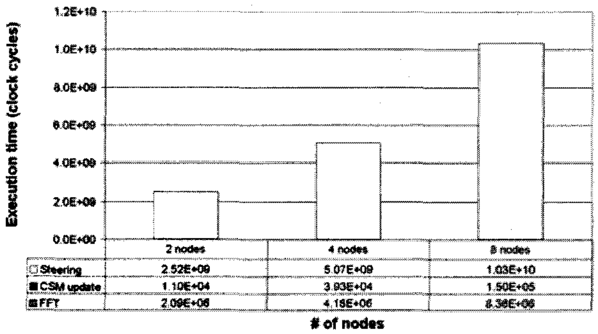


그림 7. 센서 개수에 따른 단일프로세서 기반의 실행시간
Fig. 7. Sequential execution times vs. number of sensor nodes.

1. Sequential Execution Time

The first experiment involves the execution of the sequential CMFP algorithm on a single processor, where the number of sensors is varied to study the effects of problem size. The results from the testbed is shown in Figure 7. Due to the processing unit limitation in the DSP array, the execution time results are measured up to 8 sensor nodes on the testbed.

2. Parallel Execution Time

Figure 8 and Figure 9 illustrate the parallel execution times for frequency decomposition and section decomposition, respectively. Each execution time measured is the effective execution time, which represents the amount of time between successive outputs.

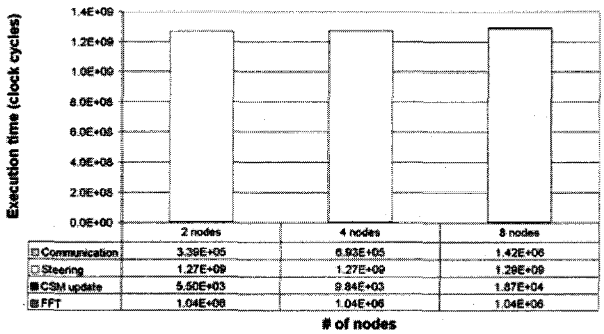


그림 8. 노드 개수에 따른 주파수 분할 병렬 알고리즘 실행시간
Fig. 8. Parallel execution time of frequency decomposition vs. number of nodes.

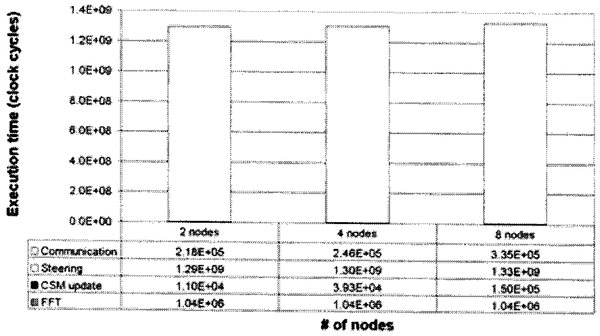


그림. 9. 노드 개수에 따른 섹션 분할 병렬 알고리즘 실행시간

Fig. 9. Parallel execution time of section decomposition vs. number of nodes.

3. Scaled Speedup

Speedup descries how much faster a parallel algorithm executes as opposed to the corresponding sequential algorithm. The larger the speedup, the better the quality of the algorithm.

On speedup, frequency decomposition provides better performance than section decomposition due to the well-balanced workload and efficient communication of DSP array. The communication is managed well on the DSP array so the performance dependency from the communication is subtle. Rather, balanced workload plays an important role on the performance on the DSP array. The DSP array achieve a scalable performance because of the model selection and efficient communication on the DSP array.

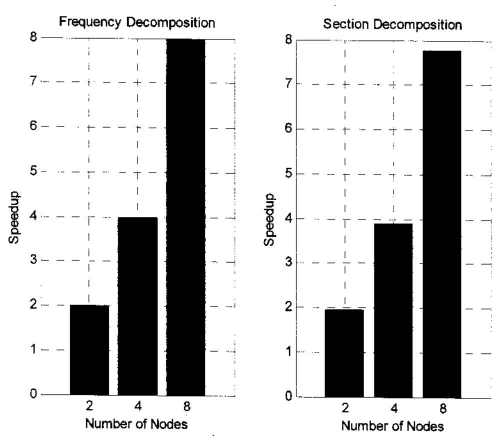


그림 10. 병렬 알고리즘 성능 (Speedup)

Fig. 10. Parallel performance.

VI. Conclusions

The parallel algorithms introduced in this paper distribute the task of CMFP in two different domains: frequency and output points. The distinct parallel testbed was employed to study the parallel performance in terms of scalable speedup.

Overall, both parallel algorithms show promising performance. For the frequency decomposition on the DSP array, scaled speedup results achieve the number close to the system size which is the ideal speedup of parallel system. Similarly, the advantage of short messages on section decomposition makes the scaled speedup increase linearly for the given problem size.

The parallel CMFP techniques described in this paper present many opportunities for increased performance, reliability, and flexibility in a distributed parallel sonar array. Future work will involve new avenue of research for distributed and parallel sonar processing. The distributed and parallel techniques developed herein can be extended to more advanced forms of sonar signal processing such as adaptive MFP algorithms for high fidelity information in the mission time.

참 고 문 헌

[1] H. P. Bucker, "Use of calculated sound fields and matched-field detection to locate sound sources in shallow water." J. Acoust. Soc. Am., Vol. 59, No. 2, pp. 368-373 (1976).

[2] R. G. Fizell and S. C. Wales, "Source localization range and depth in an arctic environment," J. Acoust. Soc. Am., Vol. 78, p. S5 (1985).

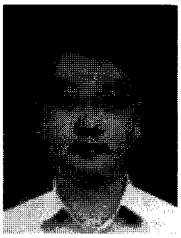
[3] N. V. Zuikova and V. D. Svet, "Matched-field processing of signals in ocean waveguides (review)," Sov. Phys. Acoust., Vol. 39, No. 3, pp. 203-210 (1993).

[4] J. Kohout, "Design and performance analysis of MPI-SHARC: a high-speed network service for distributed DSP systems," Master's thesis, ECE Department, University of Florida (Spring 2001).

[5] A. B. Baggeroer, W. A. Kuperman, and P. N. Mikhalevsky, "An overview of matched field methods in ocean acoustics," IEEE J. Ocean Eng.

Vol. 18, No. 4, pp. 401-423 (1993).
[6] United States, Office of Scientific Research and Development, National Defense Research Committee, "Physics of sound in the sea," Dept. of the Navy, Headquarters Naval Material Command, Washington, DC, pp. 8-40 (1969).
[7] F. B. Jensen, W. A. Kuperman, M. B. Porter, and H. Schmidt, "Computational ocean acoustics," AIP Press, New York (1994).
[8] M. Porter, The KRAKEN normal mode program, SCALANT Memorandum, SM-245 (Sept. 1991).
[9] M. Porter and E. L. Reiss, "A numerical method for ocean-acoustic normal modes," J. Acoust. Soc. Am., Vol. 76, No. 1, p. 244 (1984).
[10] M. Porter, Kraken Normal mode program, ftp://oalib.saic.com/pub/oalib/demo/modes.m
[11] M. S. Bartlett, "Smoothing periodograms from time series with continuous spectra," Nature, Vol. 161, pp. 686-687 (1948).
[12] Message Passing Interface Forum, "MPI: a message-passing interface standard," Technical Report CS-94-230, Computer Science Dept., Univ. of Tennessee (Apr. 1994).
[13] Bittware Research Systems, User's Guide: Blacktip-EX, Bittware Research Systems, Concord, NH, Revision 0 (1997).

저 자 소 개



김 건 욱(평생회원)
1995년 동국대학교 전자공학과 학사
1997년 University of Florida 석사
2001년 University of Florida 박사
2001년～2003년 Florida State University Assistant Professor
2003년～현재 동국대학교 전자공학과 조교수
<주관심분야 : 병렬신호처리 시스템 설계, 위치추적, 무선센서망 응용>