



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2015-0054534

(43) 공개일자 2015년05월20일

(51) 국제특허분류(Int. Cl.)

G06F 19/00 (2011.01)

(21) 출원번호 10-2013-0137089

(22) 출원일자 2013년11월12일

심사청구일자 없음

(71) 출원인

국립대학법인 울산과학기술대학교 산학협력단

울산광역시 울주군 언양읍 유니스트길 50

(72) 발명자

마르코토볼

울산 울주군 언양읍 유니스트길 50, (교수아파트)

(74) 대리인

유미특허법인

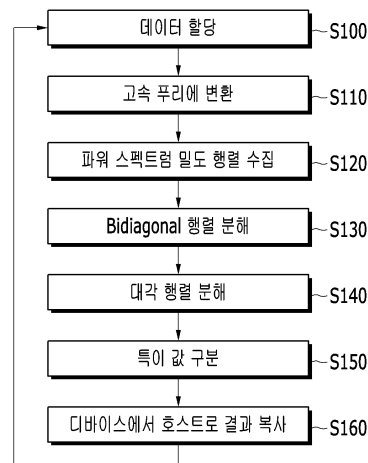
전체 청구항 수 : 총 4 항

(54) 발명의 명칭 실시간 주파수 영역 분해 방법

(57) 요약

구조물 건전성 모니터링 시스템이 실시간으로 주파수 영역을 분해하기 위하여, 구조물 건전성 모니터링 시스템은 CPU(Central Processing Unit) 및 GPU(Graphics Processing Units)를 포함하며, CPU가 CPU의 메모리 및 GPU의 메모리 사이에 메모리를 할당하면, 메모리가 할당된 GPU가 입력되는 신호에 고속 푸리에 변환을 수행하여 파워 스펙트럼 밀도 함수를 계산한다. GPU가 특이 값을 분해하면, CPU는 GPU가 분해한 특이 값을 GPU의 메모리로부터 CPU의 메모리로 복사하여 주파수 영역을 분해한다.

대표도 - 도2



이 발명을 지원한 국가연구개발사업

과제고유번호 1.130075.01

부처명 교육과학기술부

연구관리전문기관 국립대학법인 울산과학기술대학교 산학협력단

연구사업명 UMI미래전략과제

연구과제명 병렬 컴퓨팅을 이용한 원자력 발전소의 확률론적 위험성 평가

기 여 율 1/1

주관기관 국립대학법인 울산과학기술대학교 산학협력단

연구기간 2013.09.01~2014.08.31

명세서

청구범위

청구항 1

구조물 건전성 모니터링 시스템이 실시간으로 주파수 영역을 분해하는 방법에 있어서,

상기 구조물 건전성 모니터링 시스템은 CPU(Central Processing Unit) 및 GPU(Graphics Processing Units)를 포함하며,

상기 CPU가 상기 CPU의 메모리 및 상기 GPU의 메모리 사이에 메모리를 할당하는 단계;

메모리가 할당된 상기 GPU가 입력되는 신호에 고속 푸리에 변환을 수행하여 파워 스펙트럼 밀도 함수를 계산하는 단계;

상기 GPU가 특이 값을 분해하는 단계; 및

상기 CPU가 상기 GPU가 분해한 특이 값을 상기 GPU의 메모리로부터 상기 CPU의 메모리로 복사하는 단계를 포함하는 주파수 영역 분해 방법.

청구항 2

제1항에 있어서,

상기 특이 값을 분해하는 단계는,

상기 GPU가 상기 계산한 파워 스펙트럼 밀도 함수로부터 파워 스펙트럼 밀도 행렬을 수신하는 단계; 및

상기 수신한 파워 스펙트럼 밀도 행렬로부터 특이 값을 분해하는 단계

를 포함하는 주파수 영역 분해 방법.

청구항 3

제2항에 있어서,

상기 특이 값을 분해하는 단계는,

상기 파워 스펙트럼 밀도 행렬을 bidiagonal 행렬로 변환하는 단계;

상기 bidiagonal 행렬을 대각 행렬로 변환하는 단계; 및

상기 대각 행렬로부터 특이 값을 구분하여 분해하는 단계

를 포함하는 주파수 영역 분해 방법.

청구항 4

제1항에 있어서,

상기 파워 스펙트럼 밀도 함수를 계산하는 단계는,

주파수 영역 분해를 이용하여 상기 신호의 파워 스펙트럼 밀도 함수를 계산하는 주파수 영역 분해 방법.

발명의 설명

기술 분야

본 발명은 병렬연산 기반의 실시간 주파수 영역 분해 방법에 관한 것이다.

배경 기술

[0001]

[0002] 일반적으로 구조물의 고유 진동수, 모드 형태, 감쇠비를 측정하는 구조물 건전성 모니터링(SHM: Structure Health Monitoring) 시스템을 통해 수집한 데이터를 기반으로 구조물의 모달 특성을 확인하는 다양한 방법 중 주파수 영역 분해(Frequency Domain Decomposition)는 가장 많이 이용되는 방법이다. 주파수 영역 분해는 고속 푸리에 변환 단계, 파워 스펙트럼 매트릭스 생성 단계, 특이 값 분해(singular value decomposition) 단계 및 특이 값에서 고유 진동수를 확인하는 단계 등의 절차를 통해 구조물의 모달 특성을 확인한다.

[0003] 이러한 주파수 영역 분해를 일반 컴퓨터에서 수행하는 경우 수초의 시간이 걸리기 때문에, 슈퍼컴퓨터와 같은 고성능 컴퓨터를 통해 빠른 처리가 요구되고 이에 따라 많은 비용이 요구된다. 그리고 디지털 신호를 모달 특성으로 변환하기 위해서는 많은 단계를 거쳐야 되기 때문에, 실시간으로 주파수 영역 분해를 수행할 수 없으며, 구조물의 비정상적인 변화를 실시간으로 탐지하기 어려운 단점이 있다.

발명의 내용

해결하려는 과제

[0004] 따라서, 본 발명은 범용 GPU(Graphics Processing Units)를 이용하여 저비용으로 실시간으로 주파수 영역 분해 방법을 제공한다.

과제의 해결 수단

[0005] 상기 본 발명의 기술적 과제를 달성하기 위한 본 발명의 하나의 특징인 구조물 건전성 모니터링 시스템이 실시간으로 주파수 영역을 분해하는 방법은,

[0006] 상기 구조물 건전성 모니터링 시스템은 CPU(Central Processing Unit) 및 GPU(Graphics Processing Units)를 포함하며, 상기 CPU가 상기 CPU의 메모리 및 상기 GPU의 메모리 사이에 메모리를 할당하는 단계; 메모리가 할당된 상기 GPU가 입력되는 신호에 고속 푸리에 변환을 수행하여 파워 스펙트럼 밀도 함수를 계산하는 단계; 상기 GPU가 특이 값을 분해하는 단계; 및 상기 CPU가 상기 GPU가 분해한 특이 값을 상기 GPU의 메모리로부터 상기 CPU의 메모리로 복사하는 단계를 포함한다.

[0007] 상기 특이 값을 분해하는 단계는, 상기 GPU가 상기 계산한 파워 스펙트럼 밀도 함수로부터 파워 스펙트럼 밀도 행렬을 수신하는 단계; 및 상기 수신한 파워 스펙트럼 밀도 행렬로부터 특이 값을 분해하는 단계를 포함할 수 있다.

[0008] 상기 특이 값을 분해하는 단계는, 상기 파워 스펙트럼 밀도 행렬을 bidiagonal 행렬로 변환하는 단계; 상기 bidiagonal 행렬을 대각 행렬로 변환하는 단계; 및 상기 대각 행렬로부터 특이 값을 구분하여 분해하는 단계를 포함할 수 있다.

[0009] 상기 파워 스펙트럼 밀도 함수를 계산하는 단계는, 주파수 영역 분해를 이용하여 상기 신호의 파워 스펙트럼 밀도 함수를 계산할 수 있다.

발명의 효과

[0010] 본 발명에 따르면 범용 GPU에서 병렬로 주파수 영역을 분해할 수 있으므로, 저비용으로 구조물 건전성을 모니터링할 수 있다. 또한, 실시간으로 고유 진동수와 모드 형태를 모니터링할 수 있으며, 범용 GPU에서 병렬로 주파수 영역 분해를 처리하기 때문에 기존의 슈퍼컴퓨터의 CPU보다 적은 에너지 소모가 발생한다.

도면의 간단한 설명

[0011] 도 1은 일반적인 주파수 영역 분해 방법에 대한 예시도이다.

도 2는 본 발명의 실시예에 따른 병렬 주파수 영역 분해를 위한 흐름도이다.

발명을 실시하기 위한 구체적인 내용

[0012] 아래에서는 첨부한 도면을 참고로 하여 본 발명의 실시예에 대하여 본 발명이 속하는 기술 분야에서 통상의 지식을 가진 자가 용이하게 실시할 수 있도록 상세히 설명한다. 그러나 본 발명은 여러 가지 상이한 형태로 구현될 수 있으며 여기에서 설명하는 실시예에 한정되지 않는다. 그리고 도면에서 본 발명을 명확하게 설명하기 위해서 설명과 관계없는 부분은 생략하였으며, 명세서 전체를 통하여 유사한 부분에 대해서는 유사한 도면 부호를

붙였다.

- [0013] 명세서 전체에서, 어떤 부분이 어떤 구성요소를 "포함"한다고 할 때, 이는 특별히 반대되는 기재가 없는 한 다른 구성요소를 제외하는 것이 아니라 다른 구성요소를 더 포함할 수 있는 것을 의미한다.
- [0014] 이하 도면을 참조로 하여 본 발명의 실시예에 따른 실시간 주파수 영역 분해 방법에 대해 설명한다. 본 발명의 실시예에 대해 설명하기 앞서, 일반적인 주파수 영역 분해 방법에 대해 먼저 설명한다.
- [0015] 도 1은 일반적인 주파수 영역 분해 방법에 대한 예시도이다.
- [0016] 도 1에 도시된 바와 같이, 일반적으로 구조물 건전성 모니터링 시스템에서 사용되는 주파수 영역을 분해하기 위해 먼저 각각의 신호의 파워 스펙트럼 밀도(PSD: Power Spectra Density) 함수를 고속 푸리에 변환을 통해 계산한다. 그리고 신호 각 쌍의 자기 상관(Self-correlation)과 상호 상관(Cross correlation)의 파워 스펙트럼 밀도를, 두 번째 신호 쌍의 파워 스펙트럼 밀도의 복소 공액과 첫 번째 신호의 파워 스펙트럼 밀도 함수를 곱하여 계산한다.
- [0017] 여기서 각각의 w_i 의 파워 스펙트럼 밀도 행렬은 w_i 에서 신호의 자기 상관 값과 비대각 신호의 상호 상관 값을 갖는 정방 행렬이다. 그 다음 각 파워 스펙트럼 밀도 행렬의 가장 큰 특이 값과 해당 특이 벡터를 구하기 위해 특이 값 분해를 계산한다. 이들 특이 값은 구조물의 고유 진동수를 확인하기 위해 주파수 영역에서 그려질(plotted) 수 있다. 주파수 영역 분해 방법은 이미 알려진 사항으로, 본 발명의 실시예에서는 상세한 사항에 대한 설명을 생략한다.
- [0018] 이러한 일반적인 주파수 영역 분해 방법을 이용하기 위해서는 슈퍼컴퓨터와 같은 고성능 컴퓨터를 통해 빠른 처리가 요구되고 이에 따라 많은 비용이 요구되며, 실시간으로 주파수 영역 분해가 불가능하다는 단점이 있다. 따라서, 본 발명의 실시예에서는 범용 GPU 및 CPU에서 병렬로 주파수 영역을 분해할 수 있으므로, 저비용으로 구조물 건전성을 모니터링할 뿐만 아니라 실시간으로 고유 진동수와 모드 형태를 모니터링할 수 있도록 한다. 이에 대해 도 2를 참조로 설명한다.
- [0019] 본 발명의 실시예에 따라 병렬 주파수 영역 분해를 위해서, NVIDIA 그래픽 카드 상의 GPU 개발 도구인 CUDA C를 사용하여 구현되는 것을 예로 하여 설명하나, 반드시 이와 같이 한정되는 것은 아니다. 그리고 CUDA C를 통해 병렬로 데이터를 처리하기 위해 CPU와 GPU 사이에 송수신되는 신호의 흐름은 이미 알려진 사항으로, 본 발명의 실시예에서는 상세한 설명을 생략한다.
- [0020] 도 2는 본 발명의 실시예에 따른 병렬 주파수 영역 분해를 위한 흐름도이다.
- [0021] 도 2에 도시된 바와 같이, 먼저 CPU(Central Processing Unit)는 병렬 주파수 영역 분해를 위한 알고리즘을 초기화한 후, 데이터를 할당한다(S100). 여기서 데이터 할당이라 함은 호스트인 CPU와 디바이스인 GPU 사이에 메모리를 할당하는 것으로, 호스트로부터 디바이스로 디지털 신호를 전달하여 저장하는 것이다.
- [0022] 여기서 신호는 구조물 건전성 모니터링 시스템으로부터 획득하거나, CPU에 미리 저장된 데이터 또는 랜덤하게 생성된 신호 중 어느 하나일 수 있다. 그리고 신호를 호스트 메모리인 CPU 메모리에서 디바이스 메모리인 GPU 메모리로 제공한다.
- [0023] S100 단계를 통해 데이터를 할당하기 위해 다음의 수학식 1에 대한 알고리즘 내 함수들이 요구된다.

수학식 1

```
float Sh[i];
float Svdh[j];
float Uh[k];
float Sd[i];
cufftComplex Fd[j];
float Ad[k];
float Ud[k];
cudaHostAlloc (Sh[i]);
cudaHostAlloc (Svdh[j]);
cudaHostAlloc (Uh[j]);
cudaMalloc (Sd[i]);
cudaMalloc (Fd[j]);
cudaMalloc (Ad[k]);
cudaMalloc (Ud[k]);
```

[0024]

[0025]

여기서 $S^h[i]$ 는 길이 i 의 배열이다. $Svd^h[j]$ 는 특이 값을 포함하는데 j 는 $(i/2 + 1)$ 로 계산된다. 그리고 $U^h[k]$ 는 특이 벡터를 포함하는데 $k=n*n*(i/2 + 1)$ 이다. 이들 값들은 호스트에서 얻어진다.

[0026]

한편, $S^d[i]$ 는 n 개의 디지털 신호의 복사본을 포함하고, $F^d[j]$ 는 신호의 파워 스펙트럼 밀도 함수를 포함하는데, $j=n*(i/2 + 1)$ 이다. 그리고 $A^d[k]$ 는 파워 스펙트럼 밀도 행렬을 포함하는데 여기서 $k=n*n*(i/2 + 1)$ 이고, $U^d[k]$ 는 특이 벡터를 포함한다. 이들 값들은 디바이스 메모리에서 얻어진다.

[0027]

S100 단계의 메모리 할당 절차는 또 다른 배열을 요구하는데, 임시 데이터가 저장되는 곳에 저장되어 메모리가 할당된 뒤 구조물 건전성 모니터링 시스템으로부터 출력되거나 랜덤하게 생성된 신호로부터 호스트 메모리에 복사될 수도 있다. 즉, S100 단계를 통해 메모리가 할당되면, 구조물 건전성 모니터링 시스템의 출력 정보 또는 랜덤하게 생성된 신호는 호스트의 메모리 $S^h[i]$ 로 복사되거나, 호스트 메모리 $S^h[i]$ 에서 디바이스 메모리 $S^d[i]$ 로 복사된다. 오직 초기화 절차 동안에, 디지털 신호들은 초기 소스로부터 호스트 메모리로 복사된다.

[0028]

S100 단계의 메모리 할당 절차가 완료되면, GPU는 모든 신호에 대해 고속 푸리에 변환을 수행한다(S110). 일반적으로 고속 푸리에 변환시 신호 상관 함수의 파워 스펙트럼 밀도 함수는 두 방향에서 계산될 수 있다.

[0029]

하나는 FFT에서 파워 스펙트럼 밀도 함수를 계산하는 대신에 시간축 도메인에서 신호의 상관을 계산하는 것이고, 또 다른 하나는 FFT를 이용하여 각각의 신호의 파워 스펙트럼 밀도 함수를 직접 계산하는 것이다. 본 발명의 실시예에서는 FFT를 이용하여 직접 계산하는 것을 예로 하여 설명한다.

[0030]

본 발명의 실시예에서는 GPU가 FFT를 실행하는데, FFT를 실행하기 위한 입력은 디지털 신호 S^d , 출력은 파워 스펙트럼 밀도 함수 F^d 이다. 파워 스펙트럼 밀도 함수 F^d 는 파워 스펙트럼 밀도 함수를 포함하며, 단일 정밀도 부동 소수점 복소 배열(precision floating point complex array)이다.

[0031]

S110 단계에서 고속 푸리에 변환이 완료되면, GPU는 출력으로 나오는 파워 스펙트럼 밀도 함수로부터 파워 스펙트럼 밀도 행렬을 수집한다(S120). 이때 수집되는 행렬은 $N \times N$ 의 X 번째 행렬로, X 는 주파수 수이고 N 은 신호 수이다. 각각의 쓰레드(thread)는 i 번째 파워 스펙트럼 밀도 함수의 주파수 X 와 j 번째 파워 스펙트럼 밀도 함수의 주파수 X 의 복소 곱의 곱으로 구해진다.

[0032]

이러한 방법을 통하면, 각 쓰레드는 GPU에서 두 번의 쓰기(wirte)와 한 번의 읽기(read)로 구해질 수 있다. 그 결과 행렬은 대각에서 자기 상관 값과 비대각에서 상호 상관 값을 가지며, 이를 위한 함수는 다음 수학식 2와 같다.

수학식 2

"Function call" _____
 BuildPSD<<<NFFT, (NSIGNAL, NSIGNAL, 1)>>>(F^d, A^d);

"Function definition" _____
 create empty A^s {PSD matrix array on shared memory}
 F^s[block] ← F^d[block] {copy the necessary correlation PSD values to shared memory}
 A^s[threadIdx.x, threadIdx.y] ← F^s[threadIdx.x] · conj (F^s[threadIdx.y])
 A^d[block] ← A^s[block] {copy the PSD matrix to the global memory}

[0033]

[0034]

수학식 2에 나타난 함수는 파워 스펙트럼 밀도 행렬이 수집된 스레드(threads)의 NFFT 블록에서 수행되고, 각각의 블록은 NSIGNAL 스레드에 의해 NSIGNAL을 생성한다. 각각의 스레드는 오직 한번의 곱셈을 수행하고, 파워 스펙트럼 밀도 행렬의 오직 하나의 값을 채운다. 동일한 블록에 있는 스레드는 동일하게 공유된 메모리에 접근할 수 있으며, 이를 통해 빠른 메모리에서 하나의 블록의 모든 입/출력을 효과적으로 할당할 수 있게 된다.

[0035]

S120 단계를 통해 파워 스펙트럼 밀도 행렬을 수집하면, GPU는 특이 값 분해 절차를 수행한다. 특이 값 분해 단계는 전체 알고리즘 중 가장 시간 소모가 많이 요구되는 부분이다 따라서, 이를 효과적으로 수행하기 위해 다음 세 단계로 구분하여 특이 값을 분해하는데, bidiagonal 행렬 분해 단계, 대각 행렬 분해 단계 및 특이 값 구분 단계로 구분한다.

[0036]

먼저 대각 행렬을 분해하기 위해, GPU는 파워 스펙트럼 밀도 행렬 A^d로부터 하우스홀더 변환(householder transformation)을 통해 bidiagonal 행렬 Σ 로 변환된다(S130). FDD에서 모든 행렬은 정방 행렬이고, 인덱스 m은 신호의 수와 동일하다. 하우스홀더 변환은 다음 수학식 3과 같이 수행된다.

수학식 3

$$\begin{aligned} A_{mxm} &= U_{mxm} \cdot \Sigma_{mxm} \cdot V_{mxm} \\ A^{k+\frac{1}{2}} &= H^{k+\frac{1}{2}} \cdot A^k \\ A^{k+1} &= A^{k+\frac{1}{2}} \cdot G^{k+1} \\ H^{k+\frac{1}{2}} &= I - \sigma_{k+\frac{1}{2}, col} \cdot U^{k+\frac{1}{2}} \cdot V^{k+\frac{1}{2}T} \\ G^{k+1} &= I - \sigma_{k+1, col} \cdot V^{k+1} \cdot V^{k+1T} \\ Q &= \prod_{i=1}^{NSIGNAL} H^i, P = \prod_{i=1}^{NSIGNAL} G^i \end{aligned}$$

[0037]

[0038]

수학식 3을 통해 각각 반복되는 절차에서 하나의 행과 하나의 열이 제거되고, 행렬의 남은 부분이 bidiagonal 행렬로 된다. 즉, 본 발명의 실시예에 따라 하우스홀더 변환 알고리즘의 병렬 실행은 네 개의 함수로 분해되는데, 네 개의 함수는 다음과 같다. 하나는 k번째 행(column)을 제거하는 함수이고, 또 다른 하나는 A^{k+1/2}와 G^{k+1/2} 행렬의 갱신 함수이다.

[0039]

그리고 k번째 열을 제거하는 함수이고, 마지막으로 A^{k+1}와 G^{k+1} 행렬의 갱신 함수이다. 또한 수학식 3에서 행렬 Q와 행렬 P는 특이 벡터를 포함할 수 있다. 이러한 함수는 다음 수학식 4와 같이 정의될 수 있다.

수학식 4

```

"Function call"_____
for i = 1 to NSIGNAL
    ColHouseholder<<<NFFT, NSIGNAL>>>(uvd, Ad, i) {compute the householder on column i}
    UpdateHA<<<NFFT, (NSIGNAL, NSIGNAL, 1)>>>(Ad, uvd, i) {update matrix Ad}
    RowHouseholder<<<NFFT, NSIGNAL>>>(uvd, Ad, i) {compute the householder on row i}
    UpdateAG<<<NFFT, (NSIGNAL, NSIGNAL, 1)>>>(Ad, uvd, i) {update matrix Ad}
end

```

[0040]

[0041]

상기에서 설명한 하우스홀더 변환에 대한 사항은 이미 알려진 것으로, 본 발명의 실시예에서는 파워 스펙트럼 밀도 행렬이 어떠한 방법에 의해 bidiagonal 행렬로 변환되는지에 대한 상세한 설명은 생략한다.

[0042]

그리고 나서 bidiagonal 행렬은 행렬의 고유치를 구하는 "목시적 QR" 알고리즘을 통해 대각 행렬 B로 변환된다 (S140). 대각 요소는 특이 값에 해당한다. 제로 시프트(zero shift)와 함께 목시적 QR은 대각 행렬 B를 계산하기 위해 다음 수학식 5를 이용한다.

수학식 5

$$\Sigma_{mxm} = X_{mxm}^T \cdot B_{mxm} \cdot Y_{mxm}$$

[0043]

[0044]

일반적으로 목시적 QR 알고리즘은 직렬로 수행되고, 다중 쓰레드 작업이 동시에 일어나는 경우 코드 작성이 불가능하다. 그러나, bidiagonal 행렬 상의 목시적 QR을 수행하는 단일 쓰레드를 포함하는 각각의 GPU 블록에서는 여전히 목시적 QR 알고리즘을 이용하여 대각 행렬을 계산한다. 따라서, 본 발명의 실시예에서는 작은 행렬을 처리하기 위해 목시적 QR 알고리즘을 채택하여 사용하며, 목시적 QR 알고리즘을 위한 함수는 다음 수학식 6과 같다.

수학식 6

```

"Function call"_____
MoveDE<<<NPOINTFFT, NSIGNAL>>>(Ad, dd, ed) {move the non-zero elements to compact arrays}
ImplicitQR<<<NPOINTFFT, 1>>>(dd, ed) {execute implicit QR zero-shift}

```

[0045]

[0046]

그러나 만약 고유 벡터가 계산될 수 있다면, 알고리즘은 변형될 수 있다. 이 경우, 단일 쓰레드는 목시적 QR 알고리즘의 제어를 받지만, 모든 쓰레드는 고유 벡터를 가지는 X 행렬과 Y 행렬로 갱신될 수 있다. 고유 벡터를 가지는 X 행렬과 Y 행렬의 갱신을 위한 함수는 다음 수학식 7과 같다.

수학식 7

```

"Function call"_____
MoveDE<<<NPOINTFFT, NSIGNAL>>>(Ad, dd, ed) {move the non-zero elements to compact arrays}
ImplicitQR_eigenvalues<<<NPOINTFFT, (NSIGNAL, NSIGNAL, 1)>>>(Xd, Yd, dd, ed) {execute implicit QR zero-shift}

```

[0047]

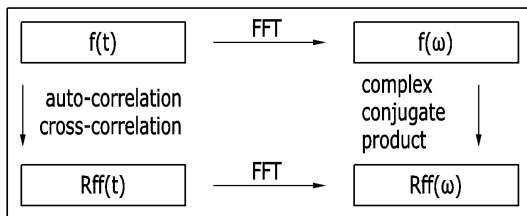
[0048]

이상에서 설명한 QR 알고리즘은 이미 알려진 알고리즘으로, QR 알고리즘을 통해 bidiagonal 행렬 Σ 이 대각 행렬 B로 변환되는 방법에 대한 상세한 설명은 생략한다.

- [0049] S140 단계를 통해 bidiagonal 행렬이 대각 행렬로 변환되면, GPU는 대각 행렬로부터 특이 값을 구분한다(S150). 즉, 행렬 X 와 행렬 Y 가 행렬 Q 와 행렬 P 에 의해 계산되는 동안 벡터 d^d 가 특이 값을 가지고 있으므로, GPU는 벡터 d^d 로부터 특이 값을 구분한다.
- [0050] 이상의 절차를 통해 특이 값이 구분되면, CPU는 구분한 특이 값을 GPU로부터 CPU 내의 메모리로 복사한다(S160). 여기서 각각의 파워 스펙트럼 밀도 행렬의 특이 값은, Q 행렬 내에 특이 값이 재 나열되는 것에 따라 큰 수치부터 정렬된다. 그리고 나서 가장 큰 특이 값과 특이 벡터가 CPU 내의 메모리에 복사된다.
- [0051] 이상에서 본 발명의 실시예에 대하여 상세하게 설명하였지만 본 발명의 권리범위는 이에 한정되는 것은 아니고 다음의 청구범위에서 정의하고 있는 본 발명의 기본 개념을 이용한 당업자의 여러 변형 및 개량 형태 또한 본 발명의 권리범위에 속하는 것이다.

도면

도면1



도면2

