(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(51) 。Int. Cl. 7
H04B 1/69

(45) 공고일자 2002년02월20일
(11) 등록번호 10 - 0324768
(24) 등록일자 2002년02월04일

(21) 출원번호 10 - 2000 - 0010162
(22) 출원일자 2000년02월29일
(65) 공개번호 특2001 - 0103844
(43) 공개일자 2001년11월24일

(73) 특허권자 엘지전자주식회사
구자홍
서울시영등포구여의도동20번지

(72) 발명자 유철우
서울특별시관악구봉천동1701번지건영아파트102동1402호
설지웅
서울특별시서초구방배2동528 - 105
강영환
서울특별시은평구갈현동308 - 19

(74) 대리인 박장원

심사관 : 정재우

(54) 차세대 이동 통신 시스템의 전송율 매칭 장치 및 방법

요약

본 발명은 차세대 이동 통신 시스템의 전송율 매칭 장치 및 방법에 관한 것으로, 종래에는 단말기에서 기지국으로 데이터 비트를 전송할 때 각 라디오 프레임별로 전송하기 때문에, 효율적인 전송율 매칭을 위해서는 블록 인터리버의 각 열에 포함되어 전송되는 데이터 비트가 균일하게 분포되어 있어야 하는데, 블록 인터리버의 각 열은 편중된 데이터 비트를 포함하게 되는 경우가 발생(특히 부호기의 비트수와 블록 인터리버의 열 수가 서로 소가 아닌 경우에 발생)하게 되는 문제점이 있다. 따라서 본 발명은 부호기를 거쳐 부호화 과정 수행 후에 블록 인터리버에서 인터리빙을 수행하고, 인터리버의 각 열에 대하여 전송 매칭을 수행한 비트들을 전송하는 통신 시스템에 있어서, 상기 부호기의 출력 비트수(n)와 블록 인터리버의 열수(Fi)가 서로 소인 경우인지 소가 아닌 경우인지를 판단하는 제1단계와, 상기에서 서로 소인 경우 부호기의 출력을 순차적으로 블록 인터리버로 입력되도록 스위칭하는 제2단계와, 상기 제1단계에서 서로 소가 아닌 경우 부호기의 출력 순서를 교차시켜 상기 블록 인터리버로 입력되도록 스위칭하여 인터리버의 각 열에 부호어 비트들을 균일하게 분포시키는 제3단계로 동작시켜, 인터리버에 데이터 비트들이 편중되지 않도록 한 것이다.

대표도

도 2

명세서

도면의 간단한 설명

도 1은 종래 단말기에서 기지국으로 데이터를 전송하는 업 - 링크(up - link) 형태의 차세대 이동 통신 시스템 구성도.

도 2는 본 발명 차세대 이동 통신 시스템의 전송 매칭 장치에 대한 제1실시예.

도 3은 본 발명 차세대 이동 통신 시스템의 전송 매칭 방법에 대한 스위칭 알고리즘.

도 4는 부호기의 부호어 비트수(n)가 4이고, 블록 인터리버의 열수(Fi)가 8인 경우, 도 3의 스위칭 알고리즘을 적용했을 때의 블록 인터리버 동작도.

도 5는 본 발명 차세대 이동 통신 시스템의 전송 매칭 방법에 대한 또 다른 스위칭 알고리즘.

도 6은 도 5의 스위칭 알고리즘을 적용할 경우의 차세대 이동 통신 시스템의 전송율 매칭 장치에 대한 구성도.

도 7은 부호기의 부호어 비트수(n)가 4이고, 블록 인터리버의 열수(Fi)가 8인 경우, 도 5의 스위칭 알고리즘을 적용했을때의 볼록 인터리버 동작도.

도 8은 도 6에서, 인터리버로 가상 비트가 인가될 경우의 새로운 블록 인터리버 구성도.

도 9은 도 2에서, 비트 종류가 2가지일 경우 매칭부의 상세 블록도.

도 10은 부호기의 부호어 비트수(n)가 2이고, 블록 인터리버의 열수(Fi)가 8인 경우 도 3의 스위칭 알고리즘을 적용했을때의 블록 인터리버 구성도.

도 11은 도 2에서, 비트 종류가 두 가지일 경우 매칭부의 상세도.

도 12는 도 9 및 도 11의 전송율 매칭 알고리즘 수행시 가상 인터리버에서 출력되는 비트들을 실제 인터리버에 저장하기 위한 위치를 조절해주는 인터리버 저장위치 조절 알고리즘.

도 13은 도 12의 알고리즘 수행시 가상 인터리버와 실제 인터리버의 저장위치를 보여주는 설명도.

도 14는 본 발명에서 블록 인터리버에 저장된 부호어 비트들과 이들을 출력시킬 경우 최적의 열 방향 치환(Column Permutation) 패턴을 보여주는 상세도.

도 15는 기지국에서 단말기로 데이터를 전송하는 다운 - 링크(down - link)의 차세대 이동 통신 시스템의 전송율 매칭 장치에 대한 구성도.

도 16 내지 도 30은 도 3의 스위칭 알고리즘과 도 12의 알고리즘을 적용하여 단말기에서 기지국으로의 데이터 전송(Up link)과 도 15에 도시한 기지국에서 단말기로의 데이터 전송(Down link)시 비트 에러율(BER)과 프레임 에러율(FER)로 보여주는 성능 비교 곡선.

***** 도면의 주요 부분에 대한 부호의 설명 *****

201 : 부호기 202 : 스위치

203 : 블록 인터리버 204 : 라디오 프레임 세그먼트

205 : 매칭부

발명의 상세한 설명

발명의 목적

발명이 속하는 기술 및 그 분야의 종래기술

본 발명은 부호화된 비트들을 블록 인터리빙한 후 전송율 매칭을 수행하여 전송하고자 할때, 블록 인터리버의 각 열에 포함되어 전송되는 데이터 비트를 균일하게 분포시키기 위한 차세대 이동 통신 시스템에 것으로, 특히 데이터 비트들의 균일한 분포를 위해 스위칭 알고리즘에 의해 부호어 비트들을 교차시키거나 가상 비트를 인터리버로 입력시켜 효율적인 인터리빙이 가능하도록 한 차세대 이동 통신 시스템의 전송율 매칭 장치 및 방법에 관한 것이다.

차세대 이동 통신 시스템에서는 다양한 전송 속도와 서비스 품질을 만족시키기 위해 오류 정정 부호와 채널 인터리버를 같이 사용하는 추세이다. 특히, IMT-2000을 위하여 세계적으로 가장 중요한 단체 중의 하나인 3rd generation partnership project(3GPP)에서도 이와 같은 방식을 채택하고 있다.

현재 사용되고 있는 인터리버는 여러가지가 있으나, 일반적으로 행과 열로 이루어져 있는 블록 인터리버를 사용한다. 3GPP도 역시 블록 인터리버의 일종을 채택하고 있다.

도 1은 일반적으로 단말기에서 기지국으로 데이터를 전송하는 업 링크(up-link) 형태의 차세대 이동 통신 시스템의 블록 구성도로서, 이에 도시된 바와 같이, 입력되는 비트열(X(t))을 부호화하여 출력하는 부호기(101)와, 상기에서 출력되는 부호어 비트들을 순차적으로 스위칭하여 출력하는 스위치(102)와, 상기에서 출력되는 부호어 비트들을 인터리빙하여 출력하는 블록 인터리버(103)와, 상기에서 인터리빙된 비트들을 하나의 라디오 프레임 단위로 잘라서 출력하는 라디오 프레임 세그먼트(104)와, 상기에서 입력되는 라디오 프레임을 받아 전송 형태를 맞추어 출력시키는 매칭부(105)로 구성된다.

이와같이 구성된 종래 기술에 대하여 살펴보면 다음과 같다.

입력되는 비트열(X(t))을 부호기(10)에서 받아 부호화 동작을 수행하여 얻은 부호어 비트들을 출력시킨다.

이렇게 출력되는 부호어 비트들을 스위치(102)가 순차적으로 스위칭하여 인터리버(103)의 제 1 행의 좌측에서 우측으로 삽입하고, 상기 제 1 행이 다 차게 되면 제 2 행, 제 2 행이 다 차게 되면 제 3행의 순서로 삽입한다.

상기에서와 같은 방법으로 모두 차게 되면 상기 인터리버(103)는 우선 제 1 열의 위에서 아래 방향으로 출력하고, 모두 출력되면 제 2 열의 위에서 아래 방향으로 출력하고, 이와 같은 방법으로 마지막 열(Fi)의 위에서 아래 방향으로 출력한다.

여기서 상기 인터리버(103)의 열수(Fi)는 전송시간간격(TTI)에 의해 정해진다. 예를 들면, TTI=10msec이면 Fi=1, TTI=20msec이면 Fi=2, TTI=40msec이면 Fi=4, TTI=80msec이면 Fi=8이 된다.

상기 인터리버(103)에서 출력되는 R개의 비트들을 라디오 프레임 세그먼트(104)에서 하나의 라디오 프레임으로 받아 총 4개의 라디오 프레임으로 만든다.

이렇게 만든 라디오 프레임을 매칭부(105)로 입력시키면, 상기 매칭부(105)는 전송 형태에 맞추기 위하여 라디오 프레임에 포함되어 있는 데이터 비트중에서 일정량의 비트를 없애거나 첨가 혹은 반복하는 전송율 매칭(rate matching)을 수행하여 기지국으로 전송한다.

발명이 이루고자 하는 기술적 과제

그러나, 상기에서와 같은 종래기술에서, 단말기에서 기지국으로 데이터 비트를 전송할 때 각 라디오 프레임별로 전송하기 때문에, 효율적인 전송율 매칭을 위해서는 블록 인터리버의 각 열에 포함되어 전송되는 데이터 비트가 균일하게 분포되어 있어야 하는데, 블록 인터리버의 각 열은 편중된 데이터 비트를 포함하게 되는 경우가 발생하게 되는 문제점이 있다. 특히 부호기의 부호어 비트수와 블록 인터리버의 열수가 서로 소가 아닌 경우에 발생하게 된다.

상기에서와 같은 종래의 문제점을 해결하기 위한 본 발명의 목적은 부호화 과정을 거쳐 인터리빙을 수행하고, 인터리버의 각 열에 대하여 전송율 매칭을 수행할 경우에, 상기 부호어를 교차적으로 인터리버에 입력시켜 인터리버의 각 열에 부호어 비트들을 균일하게 분포시키기 위한 차세대 이동 통신 시스템의 전송율 매칭 장치 및 방법을 제공함에 있다.

본 발명의 다른 목적은 부호화 과정을 거쳐 인터리빙을 수행하고, 인터리버의 각 열에 대하여 전송율 매칭을 수행할 경우에, 인터리버에 가상 비트를 입력시켜 인터리버의 각 열에 부호어 비트들을 균일하게 분포시키기 위한 차세대 이동 통신 시스템의 전송율 매칭 장치 및 방법을 제공함에 있다.

본 발명의 또 다른 목적은 메모리 버퍼와 카운터를 구비한 인터리버를 구비하여, 가상 비트 입력시엔 카운트동작을 하지 않도록 하여 많은 양의 버퍼를 필요로 하지 않도록 한 차세대 이동 통신 시스템의 전송율 매칭 장치 및 방법을 제공함에 있다.

발명의 구성 및 작용

상기 목적을 달성하기 위한 본 발명은 부호기를 거쳐 부호화 과정 수행 후에 블록 인터리버에서 인터리빙을 수행하고, 인터리버의 각 열에 대하여 전송 매칭을 수행한 비트들을 전송하는 통신 시스템에 있어서, 상기 부호기의 부호어 비트수(n)와 블록 인터리버의 열수(Fi)가 서로 소인 경우인지 소가 아닌 경우인지를 판단하는 제1단계와, 상기에서 서로 소인 경우 부호기의 출력을 순차적으로 블록 인터리버로 입력되도록 스위칭하는 제2단계와, 상기 제1단계에서 서로 소가 아닌 경우 부호기의 출력 순서를 교차시켜 상기 블록 인터리버로 입력되도록 스위칭하여 인터리버의 각 열에 부호어 비트들을 균일하게 분포시키는 제3단계로 이루어진 것을 특징으로 한다.

이하, 첨부한 도면에 의거하여 상세히 설명하면 다음과 같다.

도 3은 본 발명 차세대 이동 통신 시스템의 전송율 매칭 방법에 대한 프로그램으로서, 이에 도시한 바와 같이, 부호기의 부호어 비트수(n)와 블록 인터리버의 열수(Fi)가 서로 소인 경우인지 소가 아닌 경우인지를 판단하는 제1단계와, 상기에서 서로 소인 경우 부호기의 출력을 순차적으로 블록 인터리버로 입력되도록 스위칭하는 제2단계와, 상기 제1단계에서 서로 소가 아닌 경우 부호기의 출력 순서를 교차시켜 상기 블록 인터리버로 입력되도록 스위칭하여 인터리버의 각 열에 부호어 비트들을 균일하게 분포시키는 제3단계로 이루어진다.

또 다른 방법으로, 도 5에 도시한 바와 같이, 부호기의 출력 비트수(n)와 블록 인터리버의 열수(Fi)가 서로 소인 경우인지 소가 아닌 경우인지를 판단하는 제1단계와, 상기에서 서로 소인 경우 부호기의 출력을 순차적으로 블록 인터리버로 입력되도록 스위칭하는 제2단계와, 상기 제1단계에서 서로 소가 아닌 경우 부호기에서 순차적으로 출력되는 출력비트를 스위칭한 후 다시 가상비트가 삽입되도록 스위칭하는 동작을 반복하여 블록 인터리버로 입력시켜 인터리버의 각 열에 부호어 비트들을 균일하게 분포시키는 제3단계로 이루어진다.

이와 같이 각 단계로 이루어진 방법을 수행하기 위한 본 발명의 차세대 이동 통신 시스템의 전송율 매칭 장치 구성은, 도 2에 도시한 바와 같이, 입력 비트열 $(X^1,X^2,...,X^k)$을 부호화하고 그 부호화된 부호어 비트들$(Y^1,Y^2,Y^3,...,Y^{n-1},Y^n)$ 을 출력하는 부호기(201)와, 상기 부호기(202)에서 입력되는 부호어 비트들을 행 방향으로 저장하고 출력시엔 열 방향으로 출력토록 하는 블록 인터리버(203)와, 상기 부호기(201)의 출력 순서를 교차시켜 상기 블록 인터리버(203)로 제공하거나 상기 부호기(201)의 부호어 비트에 가상 비트를 삽입시켜 상기 블록 인터리버(203)로 제공하도록 제어하는 스위칭 알고리즘에 의해 상기 부호기(201)의 출력을 상기 블록 인터리버(203)로 스위칭하는 스위치(202)와, 상기 블록 인터리버(203)에서 열 방향으로 출력되는 부호어 비트들을 라디오 프레임 단위로 잘라 출력하는 라디오 프레임 세그먼트(204)와, 상기 라디오 프레임에 포함되어 있는 비트들을 전송율에 맞추어 매칭시켜 출력하는 매칭부(205)로 구성한다.

이와같이 구성된 본 발명의 동작 및 작용 효과에 대하여 상세히 설명하면 다음과 같다.

도 2에서, 부호기(201)는 입력되는 k비트의 입력 데이터 $(X^1(t),X^2(t),...,X^k(t))$ 를 받아 부호화 동작을 수행하여 오류 정정이 이루어진 n비트의 부호어들$(Y^1(t),Y^2(t),...,Y^n(t))$ 을 스위치(202)로 전달한다.

그러면 상기 스위치(202)는 블록 인터리버(203)의 각 열에 편중된 데이터 비트가 포함되지 않도록 하기 위한 스위칭 알고리즘(206)에 의해 스위칭하여 블록 인터리버(203)로 입력한다.

여기서 스위칭 알고리즘(206)은 도 3에 도시한 바와 같이, 부호기(201)에서 출력되는 부호어 비트수(n)와 블록 인터리버(203)의 열수(Fi)가 서로 소인 경우인지, 서로 소가 아닌 경우인지를 판단한다.

상기에서 서로 소인 경우는, 부호어 비트수(n)와 열수(Fi)의 최대공약수(GCM)가 1인 상태이고, 그 이외의 경우는 서로 소가 아닌 경우이다.

서로 소인 경우는, 부호기(201)의 출력순서값(index)과 부호어 비트수(n)에 대한 모듈러 연산(index % n)을 수행한 값에 '1'을 더한 값(k)을 스위칭하고자 하는 값Y^k을 구하고, 이를 스위칭하여 블록 인터리버(203)로 입력하도록 한다. 이와같은 동작을 부호기(201)에서 부호화되어 출력되는 총 비트수(index_Limit)만큼 될 때 까지 수행한다.

그리고, 서로 소가 아닌 경우, 부호기(201)의 출력순서값(index)과 부호어 비트수(n)에 대한 모듈러 연산(index % n)을 수행한 값에 다시 정수값(m)을 더한다. 이렇게 구한값과 부호어 비트수(n)에 대한 모듈러 연산을 다시 수행하여 얻은 값에 1을 더하여 구한 값(k)을 스위칭하고자 하는 값 (Y^k)을 구하고, 이를 스위칭하여 상기 블록 인터리버(203)로 입력하도록 한다.

단, 부호기 출력순서값(index)을, 상기 부호어 비트수(n)와 인터리버 열수(Fi)의 곱을 구하여 얻은 값과의 모듈러 연산(index % (n×Fi))을 수행하여 얻은 값이 0이면 정수값(m)을 제로(0)로 만들고, 그렇지 않으면 (index % LCM(n,Fi))의 모듈러 연산을 행한 값이 0인 경우 이전의 정수값(m)에 '1'을 더하여 정수값을 조절한다.

이상에서와 같은 스위칭 알고리즘을 수행하게 되면, 부호기(201)에서 출력되는 부호어 비트들이 블록 인터리버(203)에 교차적으로 입력된다.

따라서 도 2의 부호기(201)에서 출력되는 부호어 비트수(n)가 4이고, 블록 인터리버(203)의 열수(Fi)가 8인 경우에 도 3의 스위칭 알고리즘을 적용하게 되면, 도 4에서와 같이, 부호어 비트들은 $(y^1,y^2,y^3,y^4),(y^2,y^3,y^4,y^1),(y^3,y^4,y^1,y^2),(y^4,y^1,y^2,y^3)$ 의 순으로 블록 인터리버(203)로 입력순서가 교차되어 입력된다.

그리고 도 2의 부호기(201)에서 출력되는 부호어 비트수(n)가 2이고, 블록 인터리버(203)의 열수(Fi)가 8인 경우에 도 3의 스위칭 알고리즘을 적용하게 되면, 도 10에서와 같이, 부호어 비트들이 (y^1,y^2)와(y^2,y^1)의 순서로 교차되어 블록 인터리버(203)로 입력된다.

그리고, 또 다른 방법으로 도 5에 도시한 바와 같이, 부호어 비트수(n)와 열수(Fi)가 서로 소인 경우에는 도 3에서 설명한 바와같이 동일하게 동작하고, 소가 아닌 경우에는 부호기(201)에서 출력되는 부호비트를 한번 블록 인터리버(203)로 스위칭한 후 가상 비트(y^c)를 삽입하여 스위칭한다.

상기에서와 같은 방법으로 스위칭하게 되면 도 7에서와 같이(y^1,y^2,y^3,y^4,y^c)의 순서로 블록 인터리버(203)에 입력된다.

결국 도 3과 도 5에 도시한 스위칭 알고리즘을 이용하여 스위치(202)를 스위칭함으로써, 부호기(201)에서 출력되는 부호어 비트들이 블록 인터리버(203)로 편중되지 않고 균일하게 분포하게 된다.

그러면 상기 블록 인터리버(203)는 우선 제 1 열의 위에서 아래 방향으로 출력하고, 모두 출력되면 제 2 열의 위에서 아래 방향으로 출력하고, 이와 같은 방법으로 마지막 열(Fi)의 위에서 아래 방향으로 출력한다.

상기 인터리버(203)에서 출력되는 R개의 비트들을 라디오 프레임 세그먼트(204)은 하나의 라디오 프레임으로 받아 총 4개의 라디오 프레임으로 만든다.

이렇게 만든 라디오 프레임을 매칭부(205)로 입력하면, 상기 매칭부(205)는 라디오 프레임 단위로 일반적인 전송율 매칭(rate matching)을 수행한다.

상기에서 전송율 매칭이 이루어진 비트들을 기지국으로 전송한다.

그런데 도 7에서와 같이 가상 비트(y^c)를 블록 인터리버(203)에 삽입하게 되면 실제 필요한 인터리버를 위한 메모리 버퍼보다 더 많은 양의 버퍼를 요구하게 된다.

이와같은 단점을 해결하기 위하여 도 8에서와 같이, 카운터와 메모리 버퍼를 구비하여 부호기(201)에서 출력되는 부호어 비트들(y^1,y^2,Y^3,y^4)이 블록 인터리버(203)의 메모리 버퍼(P12-1,P12-2,P12-3.P12-4)로 입력되면, 그 메모리 버퍼를 카운트하는 카운터(P12-C1,P12-C2,P12-C3.P12-C4)는 1씩 카운터하고, 다음 가상비트 (y^c)가 메모리 버퍼(P12-5)에 입력되면 이 메모리 버퍼를 카운트하는 카운터(P12-C5)는 카운트를 증가시키지 않는다.

그리고 다음 순서에 가상 비트 이외의 부호어 비트가 입력되면 카운트를 증가시킨다.

또한 상기 메모리 버퍼(P12-5)는 가상 비트가 입력될 경우 그 신호를 저장하지 않고, 부호어 비트가 입력되면 그 비트를 저장한다.

이렇게 블록 인터리버(203)에 모두 저장된 후 열 방향으로 출력시키게 된다.

이때 통신 시스템에서 블록 인터리버의 효율을 증대시키기 위하여 인터리버에 입력된 데이터를 출력시키기 전에 인터리버의 각 열간의 순서를 바꾸어주는 열 방향 치환(column permutation)을 수행한다.

상기 열 방향 치환(column permutation)은 {0,3,2,1,6,5,4,7}로 치환하여 출력한다. 즉, 도 14에서 203a에서와 같이 블록 인터리버에 저장되어 있는 부호어 비트들을 출력할 때 제일 먼저 0번째 열의 부호어 비트들을 출력하고, 3번째 열의 부호어 비트들을 출력하고, 2번째 열의 부호어 비트들을 출력하고, 1번째 열의 부호어 비트들을 출력하는 방법으로 출력하게 되면 203b에서와 같은 순서로 부호어 비트들을 출력하게 된다.

상기에서와 같은 컬럼 순으로 치환하여 출력하게 되면 인터리버의 효율을 증대시킬 수 있다.

상기 블록 인터리버(203)에서 최적화된 컬럼 치환 순으로 출력하면, 이를 라디오 프레임 세그먼트(204)에서 받아 라디오 프레임 단위로 만들어 매칭부(205)로 전송한다.

그러면 상기 매칭부(205)는, 도 9와 도 11에서와 같이, 먼저 비트 분리기(205a)에서 라디오 프레임 세그먼트(204)로부터 제공하는 라디오 프레임에 속해있는 비트들을 종류별로 분리한 후, 각 비트 종류별로 매칭을 실시한다.

상기 비트 분리기(205a)의 출력y^i_{jk}는 j번째 라디오 프레임에서$y^i(t)$에 해당하는 비트들 중에서 k번째 비트라는 의미이다.

이렇게 분리한 비트 종류별로 각각의 전송율 매칭 알고리즘에 제공하면, 상기 전송율 매칭 알고리즘는 전송 형태를 맞추기 위하여 비트중에서 일정량의 비트를 없애거나 첨가 혹은 반복하여 생성되는 비트들$(y^{i'}_{jk})$을 비트 수집부(205b)에서 받아 상기 비트 분리기(205a)의 입력 순서대로 복원하여 다시 한 라디오 프레임을 형성하여 출력한다.

상기에서와 같이 동작하는 매칭부(205)의 전송율 알고리즘에서 생성한 종류별 비트들을 효율적인 전송율 매칭을 위해서 도 13에 도시한 바와 같이 종류별로 가상의 인터리버(501)(502)에 저장해두었다고 비트 수집부(205b)로 전달한다.

이렇게 가상의 인터리버를 사용하는 이유는 하드웨어적 복잡도 증가없이 최적의 전송율 매칭을 위한 한 방법이다.

도 12에 도시한 알고리즘을 이용하여 만들어진 가상의 인터리버를 이용하여 비트 종류별로 매칭 과정을 겪은 비트들을 받은 비트 수집부(205b)는 다시 라디오 프레임을 형성하여 출력시킨다.

도 12에서, i는 가상의 인터리버의 열 번호이고, j는 실제 인터리버의 열 번호이고, b는 $y^{1'}$의 비트가 입력되면 2,$y^{2'}$의 비트가 입력되면 1로 정의하고, Fi는 실제 인터리버의 열수로 2,4,8로 정할 수 있다.

가령, 가상 인터리버(501)의 C'_2에 저장되어 있는y^1_2의 비트가 입력될 경우 실제 인터리버(503)에 저장될 위치(j)를 도 12의 알고리즘에 의거하여 구해보면 다음과 같다.

y^1_2는 2번째의 열에 있으므로 i=2이고,y^1 비트이므로 b=2이고, 인터리버가 8비트용 이므로 Fi=8이 된다.

따라서,$j=(2\times i+(b-\lfloor 2\times i/Fi \rfloor)\%2)\%Fi$ 에 적용해보면,

$j=(2\times 2(2-\lfloor 2\times 2/8 \rfloor)\%2)\%8$ 가 된다.

여기서 $\lfloor 2\times 2/8 \rfloor$ 을 계산하면 0.5이므로 소수점 이하를 버리면 0가 되므로, 2%2 모듈러 연산을 수행하면 이 또한 0가 된다. 따라서 4%8 모듈러 연산을 하게 되면 4가 나온다.

그러면 이는 실제 인터리버(503)의 C_4 자리에 저장하면 된다.

그리고, 가상 인터리버(502)의 C'_3에 저장되어 있는 y_3^2의 비트가 입력될 경우 실제 인터리버(503)에 저장될 위치(j)를 도 12의 알고리즘에 의거하여 구해보면 다음과 같다.

y_3^2는 3번째의 열에 있으므로 i=3이고, y^2 비트이므로 b=1이고, 인터리버가 8비트용 이므로 Fi=8이 된다.

따라서, $j=(2\times i+(b-\lfloor 2\times i/Fi \rfloor)\%2)\%Fi$ 에 적용해보면,

$j=(2\times 3(1-\lfloor 2\times 3/8 \rfloor)\%2)\%8$ 가 된다.

여기서 $\lfloor 2\times 3/8 \rfloor$ 을 계산하면 0.75이므로 소수점 이하를 버리면 0가 되므로, 1%2 모듈러 연산을 수행하면 1이 된다. 그러면 마지막으로 7%8 모듈러 연산을 하게 되면 7가 나온다.

그러면 이는 실제 인터리버(503)의 C_7 자리에 저장하면 된다.

이상에서와 같은 방법으로 도 13에 도시한 가상의 인터리버(501)(502)로부터 각각 비트들이 입력되면 비트 수집부(205b)는 도 12의 알고리즘을 이용하여 실제 인터리버(503)의 위치를 찾아 저장하고 이렇게 저장한 비트들을 열 방향으로 출력한다.

기지국에서 단말기로 데이터를 전송할 경우 가장 효율적인 전송순서는, 도 15에서와 같이, 부호기, 매칭부, 블록 인터리버 및 라디오 프레임 세그먼트와 같은 순서이다.

그런데 단말기에서 기지국으로 도 15에서와 같은 순서로 데이터를 전송하게 되면 효율적인 전송이 이루어지지 않게 되므로, 도 2에서와 같이 부호기, 인터리버, 라디오 프레임 세그먼트, 매칭부와 같은 순으로 전송하여야 효율적인 전송이 이루어진다.

그러면 기지국에서 단말기로의 데이터 전송(Down link)과 단말기에서 기지국으로의 데이터 전송(Up link)시 성능을 비교한 그래프를 살펴보면 다음과 같다.

도 3의 스위칭 알고리즘과 도 12의 알고리즘을 적용한 Up link와 도 15의 Down link의 성능 비교 곡선은 도 16에서와 같이, 윗쪽의 비트 에러율(BER)과 아래쪽의 프레임 에러율(FER)이 거의 같게 나타남을 보여준다.

도 16 내지 도 30에서 실험한 정보들은 그래프의 우측 상단에 나타낸 것으로, Up은 단말기에서 기지국으로의 전송 상태를 나타내고, Down은 기지국에서 단말기로의 데이터 전송 상태를 나타내고, It은 반복되는 복호 과정 횟수, TTI는 전송 시간 구간, RMI는 전송 매칭율, BER은 비트 에러율, FER은 프레임 에러율을 각각 나타낸다.

도 16 ~ 도 20, 도 28~도 30은 데이터 블록단 입력 비트 수가 322, 인터리버 크기가 486일 경우의 실험 결과이고, 도 21 ~ 도 27은 데이터 블록단 입력 비트 수가 324, 인터리버 크기가 489일 경우의 실험 결과이다.

발명의 효과

이상에서 상세히 설명한 바와 같이 본 발명은 오류 정정 부호화 과정을 통하여 발생한 부호어를 블록 인터리버를 통하여 인터리빙한 후 전송 매칭 과정을 수행할 때, 블록 인터리버의 각 열에 포함되어 전송되는 데이터 비트를 균일하게 분포시켜 효율적인 전송 매칭이 이루어지도록 하고, 시스템에 부가되는 하드웨어적 복잡도가 거의 없이 비트 에러율과 프레임 에러율 상의 성능을 향상시키도록 한다. 그리고 성능 향상에 의해 전송 전력이나 싯템 성능, 사용자 용량 면에서 이득이 발생하도록 하는 효과가 있다.

(57) 청구의 범위

청구항 1.

입력 비트열을 부호화하는 부호기의 부호 비트들을 인터리빙하여 출력하는 블록 인터리버와, 상기 부호 비트들을 상기 블록 인터리버로 스위칭하여주는 스위치와, 인터리빙된 비트들을 라디오 프레임 단위로 잘라서 출력하는 라디오 프레임 세그먼트와, 상기 라디오 프레임에 포함되어 있는 비트들을 전송율에 맞추어 매칭시켜 출력하는 매칭부로 이루어진 통신 시스템에 있어서, 상기 블록 인터리버는 부호 비트들을 저장하기 위한 열수 만큼의 메모리 버퍼와, 상기 메모리 버퍼로 가상 비트가 들어오면 카운트동작을 수행하지 않고 가상 비트 이외의 신호가 들어오면 카운트하는 카운터를 포함한 것을 특징으로 하는 차세대 이동 통신 시스템의 전송율 매칭 장치.

청구항 2.

제1항에 있어서, 블록 인터리버는 입력에 대하여 {0,3,2,1,6,5,4,7}의 열로 치환한 후 출력하여 인터리빙 효율을 증대시키도록 한 것을 특징으로 하는 차세대 이동 통신 시스템의 전송율 매칭 장치.

청구항 3.

제1항에 있어서, 매칭부는 라디오 프레임 단위로 입력되는 비트열을 종류별로 분리하는 비트 분리기와, 상기에서 종류별로 분리된 비트들을 받아 전송율 매칭을 위해 지우거나 반복시켜 생성하고, 이 생성된 비트들을 종류별로 가상 인터리버에 저장해두는 전송매칭 알고리즘과, 상기 가상 인터리버로부터 출력되는 비트들을 받아 상기 비트 분리기로 입력되는 비트들의 입력 순서와 맞게 조절하여 해당 위치에 저장하기 위한 비트 수집부로 이루어진 것을 특징으로 하는 차세대 이동 통신 시스템의 전송율 매칭 장치.

청구항 4.

제3항에 있어서, 비트 수집부는 블록 인터리버인 것을 특징으로 하는 차세대 이동 통신 시스템의 전송율 매칭 장치.

청구항 5.

제4항에 있어서, 가상 인터리버에서 출력되는 비트를 블록 인터리버로 입력하는 제어는 인터리버 저장위치 알고리즘에 의한 것임을 특징으로 하는 차세대 이동 통신 시스템의 전송율 매칭 장치.

청구항 6.

제5항에 있어서, 인터리버 저장위치 알고리즘은 아래에서와 같도록 한 것을 특징으로 하는 차세대 이동 통신시스템의 전송율 매칭 장치.

$$j=(2\times i+(b-\lfloor 2\times i/Fi\rfloor)\%2)\%Fi$$

단, j는 실제 인터리버의 열 번호이고, i는 가상 인터리버의 열 번호이고, Fi는 실제 인터리버의 열수이고, b는 비트 종류에 따라 정해지는 상수이다.

청구항 7.

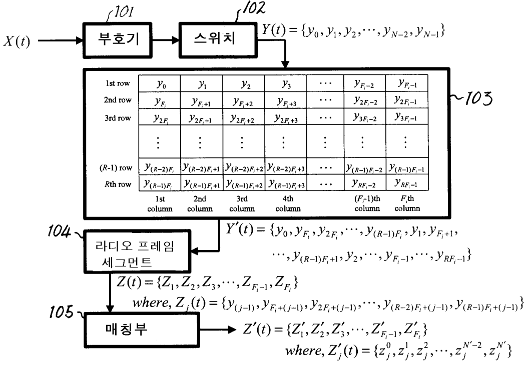
부호기를 거쳐 부호화 과정 수행 후에 블록 인터리버에서 인터리빙을 수행하고, 인터리버의 각 열에 대하여 전송 매칭을 수행한 비트들을 전송하는 통신 시스템에 있어서, 상기 부호기의 출력 비트수(n)와 블록 인터리버의 열수(Fi)가 서로 소인 경우인지 소가 아닌 경우인지를 판단하는 제1단계와, 상기에서 서로 소인 경우 부호기의 출력을 순차적으로 블록 인터리버로 입력되도록 스위칭하는 제2단계와, 상기 제1단계에서 서로 소가 아닌 경우 부호기의 출력 순서를 교차시켜 상기 블록 인터리버로 입력되도록 스위칭하여 인터리버의 각 열에 부호어 비트들을 균일하게 분포시키는 제3단계로 이루어진 것을 특징으로 하는 차세대 이동 시스템의 전송율 매칭 방법.

청구항 8.

부호기를 거쳐 부호화 과정 수행 후에 블록 인터리버에서 인터리빙을 수행하고, 인터리버의 각 열에 대하여 전송 매칭을 수행한 비트들을 전송하는 통신 시스템에 있어서 , 상기 부호기의 출력 비트수(n)와 블록 인터리버의 열수(Fi)가 서로 소인 경우인지 소가 아닌 경우인지를 판단하는 제1단계와, 상기에서 서로 소인 경우 부호기의 출력을 순차적으로 블록 인터리버로 입력되도록 스위칭하는 제2단계와, 상기 제1단계에서 서로 소가 아닌 경우 부호기에서 순차적으로 출력되는 출력비트를 한번 스위칭한 후 가상비트가 삽입되도록 스위칭하는 동작을 반복하여 블록 인터리버로 입력시켜 인터리버의 각 열에 부호어 비트들을 균일하게 분포시키는 제3단계로 이루어진 것을 특징으로 하는 차세대 이동 통신 시스템의 전송율 매칭 방법.

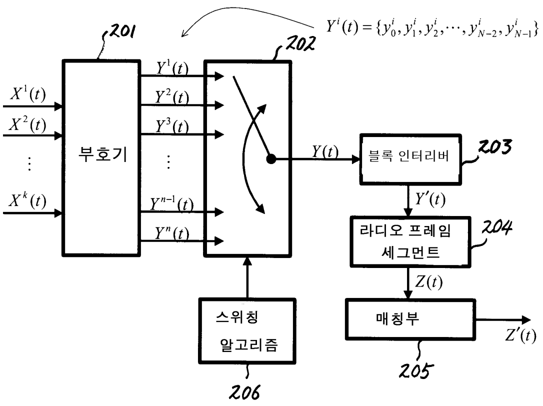
도면

도면 1


101
부호기
102
스위치
$X(t)$
$Y(t)=\{y_0,y_1,y_2,\cdots,y_{N-2},y_{N-1}\}$
103
104
라디오 프레임 세그먼트
$Y'(t)=\{y_0,y_{F_i},y_{2F_i},\cdots,y_{(R-1)F_i},y_1,y_{F_i+1},\cdots,y_{(R-1)F_i+1},y_2,\cdots,y_{F_i-1},\cdots,y_{RF_i-1}\}$
$Z(t)=\{Z_1,Z_2,Z_3,\cdots,Z_{F_i-1},Z_{F_i}\}$
$where,\ Z_j(t)=\{y_{(j-1)},y_{F_i+(j-1)},y_{2F_i+(j-1)},\cdots,y_{(R-2)F_i+(j-1)},y_{(R-1)F_i+(j-1)}\}$
105
매칭부
$Z'(t)=\{Z'_1,Z'_2,Z'_3,\cdots,Z'_{F_i-1},Z'_{F_i}\}$
$where,\ Z'_j(t)=\{z_j^0,z_j^1,z_j^2,\cdots,z_j^{N'-2},z_j^{N'}\}$

	1st column	2nd column	3rd column	4th column		(F_i-1)th column	F_ith column
1st row	y_0	y_1	y_2	y_3	$\cdots$	y_{F_i-2}	y_{F_i-1}
2nd row	y_{F_i}	y_{F_i+1}	y_{F_i+2}	y_{F_i+3}	$\cdots$	y_{2F_i-2}	y_{2F_i-1}
3rd row	y_{2F_i}	y_{2F_i+1}	y_{2F_i+2}	y_{2F_i+3}	$\cdots$	y_{3F_i-2}	y_{3F_i-1}
	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
(R-1) row	$y_{(R-2)F_i}$	$y_{(R-2)F_i+1}$	$y_{(R-2)F_i+2}$	$y_{(R-2)F_i+3}$	$\cdots$	$y_{(R-1)F_i-2}$	$y_{(R-1)F_i-1}$
Rth row	$y_{(R-1)F_i}$	$y_{(R-1)F_i+1}$	$y_{(R-1)F_i+2}$	$y_{(R-1)F_i+3}$	$\cdots$	y_{RF_i-2}	y_{RF_i-1}

도면 2

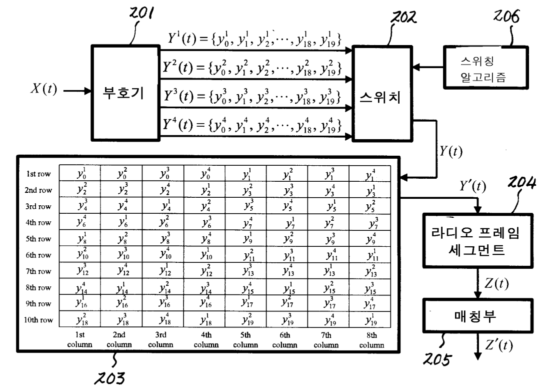

$Y^i(t)=\{y_0^i, y_1^i, y_2^i, \cdots, y_{N-2}^i, y_{N-1}^i\}$
201
202
$X^1(t)$
$X^2(t)$
$X^k(t)$
부호기
$Y^1(t)$
$Y^2(t)$
$Y^3(t)$
$Y^{n-1}(t)$
$Y^n(t)$
$Y(t)$
블록 인터리버
203
$Y'(t)$
라디오 프레임 세그먼트
204
$Z(t)$
매칭부
$Z'(t)$
205
스위칭 알고리즘
206

도면 3

```
if (GCM(n,F_i) = 1) {                //n과 Fi가 서로 소인 경우
        index=0;
        do while (index < index_Limit)
                k=(index % n) + 1;
                switching at Y^k;
                index=index+1;
        end do;
} else {                             //n과 Fi가 서로 소가 아닌 경우
        index=0;
        m=0;
        do while (index < index_Limit)
                k=( (index % n) + m) % n +1);
                switching at Y^k;
                index=index+1;
                if (  (index % (n×F_i))=0  )  {m=0;}
                else
                        if (  (index % LCM(n,F_i))=0  )  {m=m+1;}
        end do;
}
```

LCM(a,b): a와 b의 최소 공배수
GCM(a,b): a와 b의 최대 공약수
%: 모듈러 연산

도면 4


201
202
206
$X(t)$
부호기
$Y^1(t)=\{y_0^1, y_1^1, y_2^1, \cdots, y_{18}^1, y_{19}^1\}$
$Y^2(t)=\{y_0^2, y_1^2, y_2^2, \cdots, y_{18}^2, y_{19}^2\}$
$Y^3(t)=\{y_0^3, y_1^3, y_2^3, \cdots, y_{18}^3, y_{19}^3\}$
$Y^4(t)=\{y_0^4, y_1^4, y_2^4, \cdots, y_{18}^4, y_{19}^4\}$
스위치
스위칭 알고리즘
$Y(t)$
$Y'(t)$
204
라디오 프레임 세그먼트
$Z(t)$
매칭부
$Z'(t)$
205
203

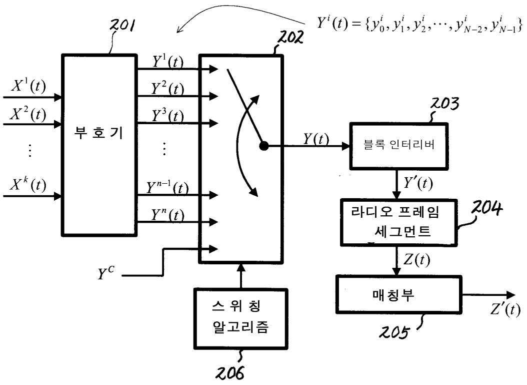
	1st column	2nd column	3rd column	4th column	5th column	6th column	7th column	8th column
1st row	y_0^1	y_0^2	y_0^3	y_0^4	y_1^1	y_1^2	y_1^3	y_1^4
2nd row	y_2^2	y_2^3	y_2^4	y_2^1	y_3^2	y_3^3	y_3^4	y_3^1
3rd row	y_4^3	y_4^4	y_4^1	y_4^2	y_5^3	y_5^4	y_5^1	y_5^2
4th row	y_6^4	y_6^1	y_6^2	y_6^3	y_7^4	y_7^1	y_7^2	y_7^3
5th row	y_8^1	y_8^2	y_8^3	y_8^4	y_9^1	y_9^2	y_9^3	y_9^4
6th row	y_{10}^2	y_{10}^3	y_{10}^4	y_{10}^1	y_{11}^2	y_{11}^3	y_{11}^4	y_{11}^1
7th row	y_{12}^3	y_{12}^4	y_{12}^1	y_{12}^2	y_{13}^3	y_{13}^4	y_{13}^1	y_{13}^2
8th row	y_{14}^4	y_{14}^1	y_{14}^2	y_{14}^3	y_{15}^4	y_{15}^1	y_{15}^2	y_{15}^3
9th row	y_{16}^1	y_{16}^2	y_{16}^3	y_{16}^4	y_{17}^1	y_{17}^2	y_{17}^3	y_{17}^4
10th row	y_{18}^2	y_{18}^3	y_{18}^4	y_{18}^1	y_{19}^2	y_{19}^3	y_{19}^4	y_{19}^1

도면 5

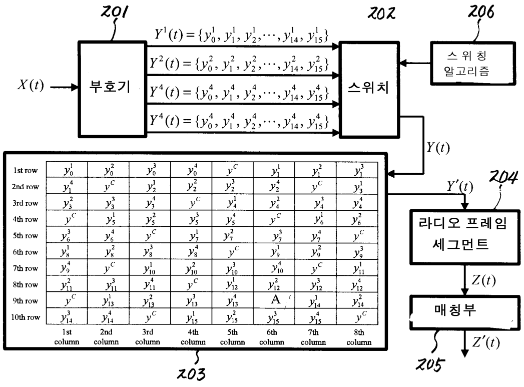
```
if (GCM(n,Fi) = 1) {                    //n과 Fi가 서로 소인 경우
        index=0;
        do while (index < index_Limit)
                k=(index % n) + 1;
                switching at Y^k;
                index=index+1;
        end do;
} else {                                //n과 Fi가 서로 소가 아닌 경우
        index=0;
        do while (index < index_Limit)
                k=(index % n) + 1;
                Switching at Y^k;
                index=index+1;
                if (index%n = 0)        {switching at Y^C;}
        end do;
}
```

LCM(a,b): a와 b의 최소 공배수
GCM(a,b): a와 b의 최대 공약수
%: 모듈러 연산

도면 6

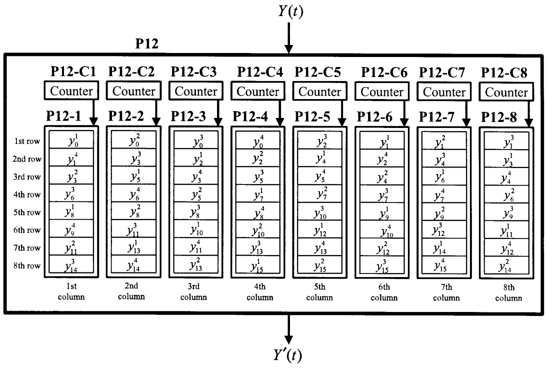

$Y^i(t)=\{y_0^i, y_1^i, y_2^i, \cdots, y_{N-2}^i, y_{N-1}^i\}$
201
202
$X^1(t)$
$X^2(t)$
$X^k(t)$
부 호 기
$Y^1(t)$
$Y^2(t)$
$Y^3(t)$
$Y^{n-1}(t)$
$Y^n(t)$
Y^C
$Y(t)$
203
블록 인터리버
$Y'(t)$
204
라디오 프레임 세그먼트
$Z(t)$
매칭부
$Z'(t)$
205
스 위 칭 알고리즘
206

도면 7


201
202
206
$X(t)$
부호기
$Y^1(t)=\{y_0^1, y_1^1, y_2^1, \cdots, y_{14}^1, y_{15}^1\}$
$Y^2(t)=\{y_0^2, y_1^2, y_2^2, \cdots, y_{14}^2, y_{15}^2\}$
$Y^4(t)=\{y_0^4, y_1^4, y_2^4, \cdots, y_{14}^4, y_{15}^4\}$
$Y^4(t)=\{y_0^4, y_1^4, y_2^4, \cdots, y_{14}^4, y_{15}^4\}$
스위치
스 위 칭 알고리즘
$Y(t)$
$Y'(t)$
204
라디오 프레임 세그먼트
$Z(t)$
매칭부
$Z'(t)$
205
203

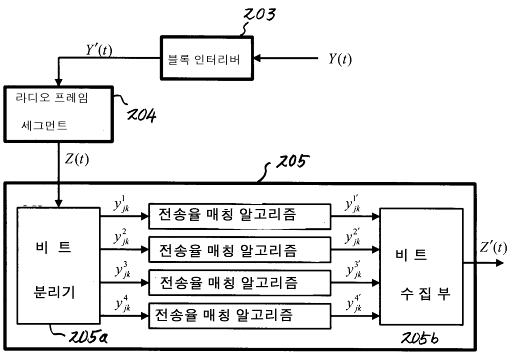
	1st column	2nd column	3rd column	4th column	5th column	6th column	7th column	8th column
1st row	y_0^1	y_0^2	y_0^3	y_0^4	y^C	y_1^1	y_1^2	y_1^3
2nd row	y_1^4	y^C	y_2^1	y_2^2	y_2^3	y_2^4	y^C	y_3^1
3rd row	y_3^2	y_3^3	y_3^4	y^C	y_4^1	y_4^2	y_4^3	y_4^4
4th row	y^C	y_5^1	y_5^2	y_5^3	y_5^4	y^C	y_6^1	y_6^2
5th row	y_6^3	y_6^4	y^C	y_7^1	y_7^2	y_7^3	y_7^4	y^C
6th row	y_8^1	y_8^2	y_8^3	y_8^4	y^C	y_9^1	y_9^2	y_9^3
7th row	y_9^4	y^C	y_{10}^1	y_{10}^2	y_{10}^3	y_{10}^4	y^C	y_{11}^1
8th row	y_{11}^2	y_{11}^3	y_{11}^4	y^C	y_{12}^1	y_{12}^2	y_{12}^3	y_{12}^4
9th row	y^C	y_{13}^1	y_{13}^2	y_{13}^3	y_{13}^4	A	y_{14}^1	y_{14}^2
10th row	y_{14}^3	y_{14}^4	y^C	y_{15}^1	y_{15}^2	y_{15}^3	y_{15}^4	y^C

도면 8

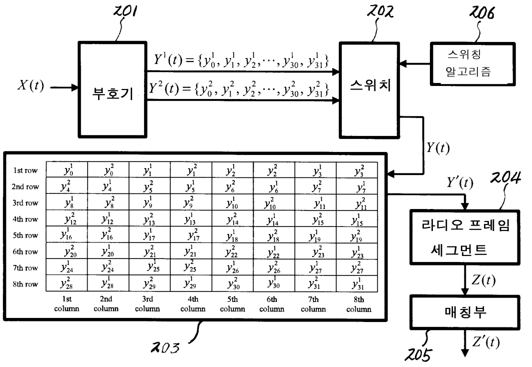

$Y(t)$
P12
P12-C1
P12-C2
P12-C3
P12-C4
P12-C5
P12-C6
P12-C7
P12-C8
Counter
Counter
Counter
Counter
Counter
Counter
Counter
Counter
P12-1
P12-2
P12-3
P12-4
P12-5
P12-6
P12-7
P12-8
$Y'(t)$

	1st column	2nd column	3rd column	4th column	5th column	6th column	7th column	8th column
1st row	y_0^1	y_0^2	y_0^3	y_0^4	y_2^3	y_1^1	y_1^2	y_1^3
2nd row	y_1^4	y_3^3	y_2^1	y_2^2	y_4^1	y_2^4	y_4^3	y_3^1
3rd row	y_3^2	y_5^1	y_3^4	y_5^3	y_5^4	y_4^2	y_6^1	y_4^4
4th row	y_6^3	y_6^4	y_5^2	y_7^1	y_7^2	y_7^3	y_7^4	y_6^2
5th row	y_8^1	y_8^2	y_8^3	y_8^4	y_{10}^3	y_9^1	y_9^2	y_9^3
6th row	y_9^4	y_{11}^3	y_{10}^1	y_{10}^2	y_{12}^1	y_{10}^4	y_{12}^3	y_{11}^1
7th row	y_{11}^2	y_{13}^1	y_{11}^4	y_{13}^3	y_{13}^4	y_{12}^2	y_{14}^1	y_{12}^4
8th row	y_{14}^3	y_{14}^4	y_{13}^2	y_{15}^1	y_{15}^2	y_{15}^3	y_{15}^4	y_{14}^2

도면 9


203
블록 인터리버
$Y'(t)$
$Y(t)$
라디오 프레임 세그먼트
204
$Z(t)$
205
비 트 분리기
y_{jk}^1
y_{jk}^2
y_{jk}^3
y_{jk}^4
전송율 매칭 알고리즘
전송율 매칭 알고리즘
전송율 매칭 알고리즘
전송율 매칭 알고리즘
$y_{jk}^{1'}$
$y_{jk}^{2'}$
$y_{jk}^{3'}$
$y_{jk}^{4'}$
비 트 수 집 부
$Z'(t)$
205a
205b

도면 10

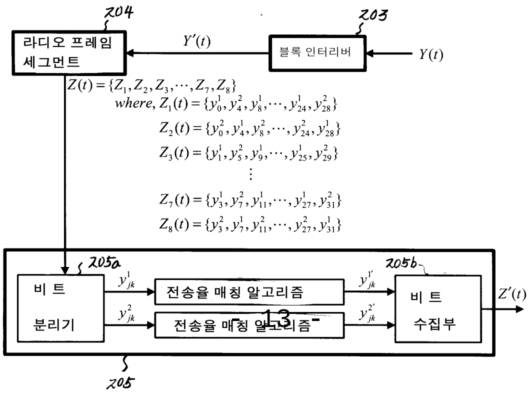

201
202
206
$X(t)$
부호기
스위치
스위칭 알고리즘
$Y(t)$
$Y'(t)$
204
라디오 프레임 세그먼트
$Z(t)$
매칭부
$Z'(t)$
205
203

$Y^1(t)=\{y_0^1, y_1^1, y_2^1, \cdots, y_{30}^1, y_{31}^1\}$

$Y^2(t)=\{y_0^2, y_1^2, y_2^2, \cdots, y_{30}^2, y_{31}^2\}$

	1st column	2nd column	3rd column	4th column	5th column	6th column	7th column	8th column
1st row	y_0^1	y_0^2	y_1^1	y_1^2	y_2^1	y_2^2	y_3^1	y_3^2
2nd row	y_4^2	y_4^1	y_5^2	y_5^1	y_6^2	y_6^1	y_7^2	y_7^1
3rd row	y_8^1	y_8^2	y_9^1	y_9^2	y_{10}^1	y_{10}^2	y_{11}^1	y_{11}^2
4th row	y_{12}^2	y_{12}^1	y_{13}^2	y_{13}^1	y_{14}^2	y_{14}^1	y_{15}^2	y_{15}^1
5th row	y_{16}^1	y_{16}^2	y_{17}^1	y_{17}^2	y_{18}^1	y_{18}^2	y_{19}^1	y_{19}^2
6th row	y_{20}^2	y_{20}^1	y_{21}^2	y_{21}^1	y_{22}^2	y_{22}^1	y_{23}^2	y_{23}^1
7th row	y_{24}^1	y_{24}^2	y_{25}^1	y_{25}^2	y_{26}^1	y_{26}^2	y_{27}^1	y_{27}^2
8th row	y_{28}^2	y_{28}^1	y_{29}^2	y_{29}^1	y_{30}^2	y_{30}^1	y_{31}^2	y_{31}^1

도면 11


204
203
라디오 프레임 세그먼트
$Y'(t)$
블록 인터리버
$Y(t)$
205a
205b
비 트 분리기
y_{jk}^1
y_{jk}^2
전송율 매칭 알고리즘
전송율 매칭 알고리즘
$y_{jk}^{1'}$
$y_{jk}^{2'}$
비 트 수집부
$Z'(t)$
205

$Z(t)=\{Z_1, Z_2, Z_3, \cdots, Z_7, Z_8\}$

$where, Z_1(t)=\{y_0^1, y_4^2, y_8^1, \cdots, y_{24}^1, y_{28}^2\}$

$Z_2(t)=\{y_0^2, y_4^1, y_8^2, \cdots, y_{24}^2, y_{28}^1\}$

$Z_3(t)=\{y_1^1, y_5^2, y_9^1, \cdots, y_{25}^1, y_{29}^2\}$

$\vdots$

$Z_7(t)=\{y_3^1, y_7^2, y_{11}^1, \cdots, y_{27}^1, y_{31}^2\}$

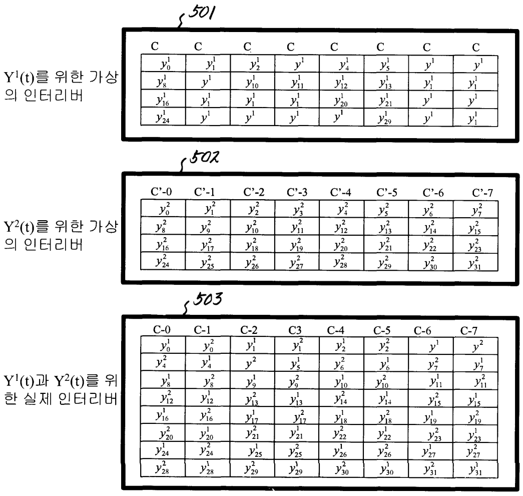
$Z_8(t)=\{y_3^2, y_7^1, y_{11}^2, \cdots, y_{27}^2, y_{31}^1\}$

도면 12

for (i=0; $i<F_i$; i++) $j=(\,2\times i+(b-\lfloor 2\times i/Fi\rfloor)\%2\,)\ \%\ F_i$

where, index b is used to indicate Y^1's bit (b=2) and Y^2's bit (b=1), and $F_i \in \{2, 4, 8\}$.

도면 13

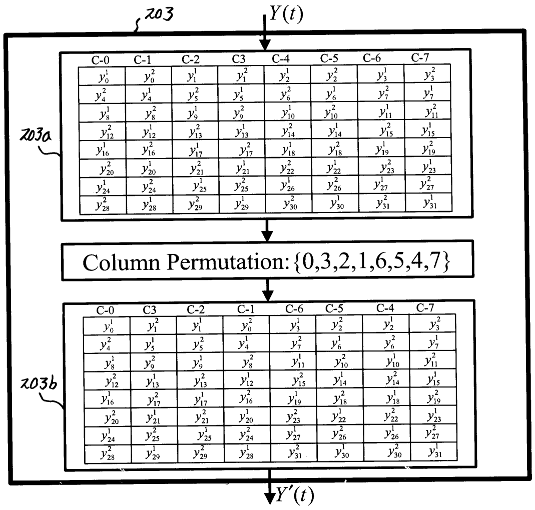

501
Y¹(t)를 위한 가상의 인터리버
502
Y²(t)를 위한 가상의 인터리버
503
Y¹(t)과 Y²(t)를 위한 실제 인터리버

C	C	C	C	C	C	C	C
y_0^1	y_1^1	y_2^1	y^1	y_4^1	y_5^1	y^1	y^1
y_8^1	y^1	y_{10}^1	y_{11}^1	y_{12}^1	y_{13}^1	y_1^1	y_1^1
y_{16}^1	y_1^1	y_1^1	y_1^1	y_{20}^1	y_{21}^1	y^1	y^1
y_{24}^1	y^1	y^1	y^1	y^1	y_{29}^1	y^1	y_1^1

C'-0	C'-1	C'-2	C'-3	C'-4	C'-5	C'-6	C'-7
y_0^2	y_1^2	y_2^2	y_3^2	y_4^2	y_5^2	y_6^2	y_7^2
y_8^2	y_9^2	y_{10}^2	y_{11}^2	y_{12}^2	y_{13}^2	y_{14}^2	y_{15}^2
y_{16}^2	y_{17}^2	y_{18}^2	y_{19}^2	y_{20}^2	y_{21}^2	y_{22}^2	y_{23}^2
y_{24}^2	y_{25}^2	y_{26}^2	y_{27}^2	y_{28}^2	y_{29}^2	y_{30}^2	y_{31}^2

C-0	C-1	C-2	C3	C-4	C-5	C-6	C-7
y_0^1	y_0^2	y_1^1	y_1^2	y_2^1	y_2^2	y^1	y^2
y_4^2	y_4^1	y^2	y_5^1	y_6^2	y_6^1	y_7^2	y_7^1
y_8^1	y_8^2	y_9^1	y_9^2	y_{10}^1	y_{10}^2	y_{11}^1	y_{11}^2
y_{12}^2	y_{12}^1	y_{13}^2	y_{13}^1	y_{14}^2	y_{14}^1	y_{15}^2	y_{15}^1
y_{16}^1	y_{16}^2	y_{17}^1	y_{17}^2	y_{18}^1	y_{18}^2	y_{19}^1	y_{19}^2
y_{20}^2	y_{20}^1	y_{21}^2	y_{21}^1	y_{22}^2	y_{22}^1	y_{23}^2	y_{23}^1
y_{24}^1	y_{24}^2	y_{25}^1	y_{25}^2	y_{26}^1	y_{26}^2	y_{27}^1	y_{27}^2
y_{28}^2	y_{28}^1	y_{29}^2	y_{29}^1	y_{30}^2	y_{30}^1	y_{31}^2	y_{31}^1

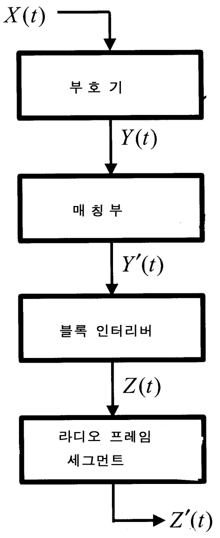
도면 14



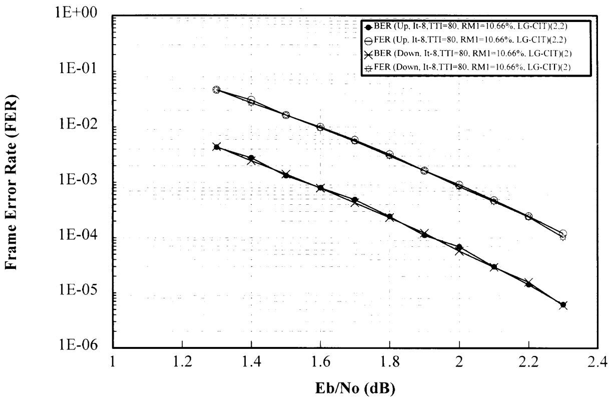
C-0	C-1	C-2	C3	C-4	C-5	C-6	C-7
y_0^1	y_0^2	y_1^1	y_1^2	y_2^1	y_2^2	y_3^1	y_3^2
y_4^2	y_4^1	y_5^2	y_5^1	y_6^2	y_6^1	y_7^2	y_7^1
y_8^1	y_8^2	y_9^1	y_9^2	y_{10}^1	y_{10}^2	y_{11}^1	y_{11}^2
y_{12}^2	y_{12}^1	y_{13}^2	y_{13}^1	y_{14}^2	y_{14}^1	y_{15}^2	y_{15}^1
y_{16}^1	y_{16}^2	y_{17}^1	y_{17}^2	y_{18}^1	y_{18}^2	y_{19}^1	y_{19}^2
y_{20}^2	y_{20}^1	y_{21}^2	y_{21}^1	y_{22}^2	y_{22}^1	y_{23}^2	y_{23}^1
y_{24}^1	y_{24}^2	y_{25}^1	y_{25}^2	y_{26}^1	y_{26}^2	y_{27}^1	y_{27}^2
y_{28}^2	y_{28}^1	y_{29}^2	y_{29}^1	y_{30}^2	y_{30}^1	y_{31}^2	y_{31}^1

C-0	C3	C-2	C-1	C-6	C-5	C-4	C-7
y_0^1	y_1^2	y_1^1	y_0^2	y_3^1	y_2^2	y_2^1	y_3^2
y_4^2	y_5^1	y_5^2	y_4^1	y_7^2	y_6^1	y_6^2	y_7^1
y_8^1	y_9^2	y_9^1	y_8^2	y_{11}^1	y_{10}^2	y_{10}^1	y_{11}^2
y_{12}^2	y_{13}^1	y_{13}^2	y_{12}^1	y_{15}^2	y_{14}^1	y_{14}^2	y_{15}^1
y_{16}^1	y_{17}^2	y_{17}^1	y_{16}^2	y_{19}^1	y_{18}^2	y_{18}^1	y_{19}^2
y_{20}^2	y_{21}^1	y_{21}^2	y_{20}^1	y_{23}^2	y_{22}^1	y_{22}^2	y_{23}^1
y_{24}^1	y_{25}^2	y_{25}^1	y_{24}^2	y_{27}^1	y_{26}^2	y_{26}^1	y_{27}^2
y_{28}^2	y_{29}^1	y_{29}^2	y_{28}^1	y_{31}^2	y_{30}^1	y_{30}^2	y_{31}^1

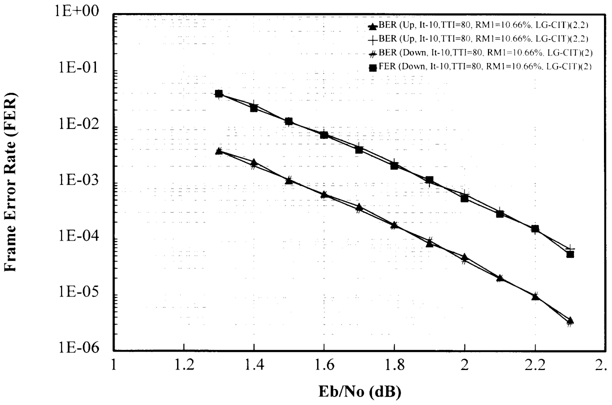
도면 15


$X(t)$
부 호 기
$Y(t)$
매 칭 부
$Y'(t)$
블록 인터리버
$Z(t)$
라디오 프레임
세그먼트
$Z'(t)$

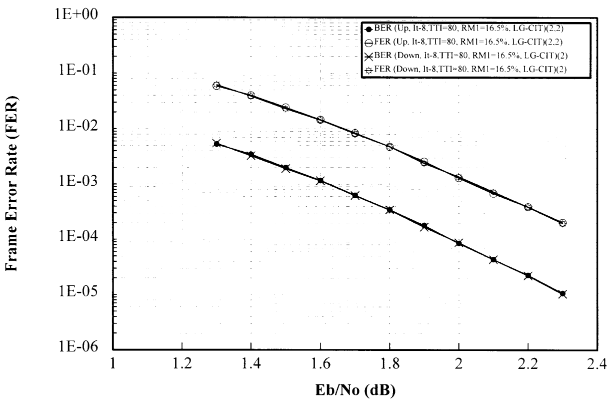
도면 16



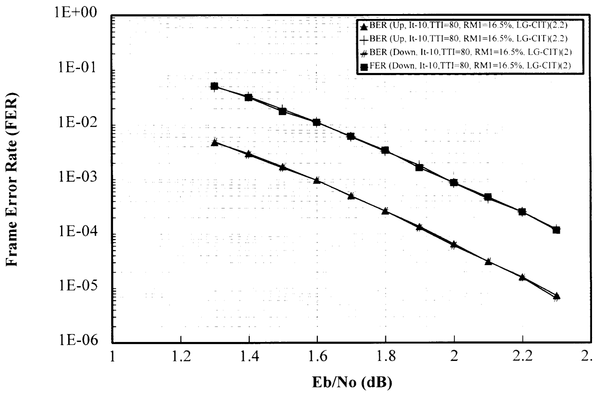
도면 17



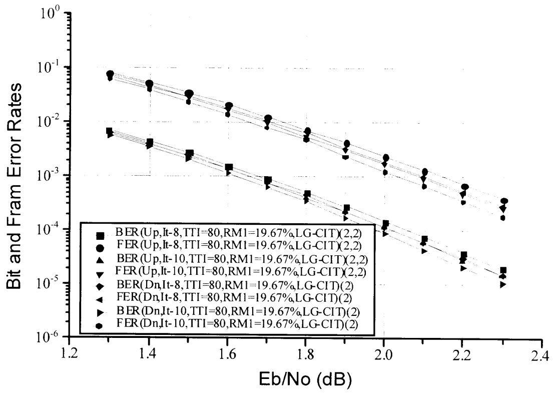
도면 18



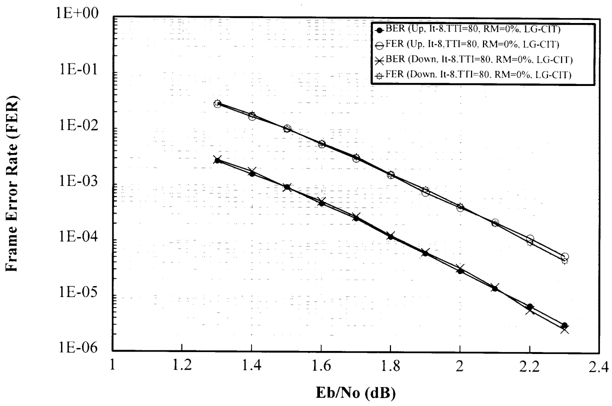
도면 19



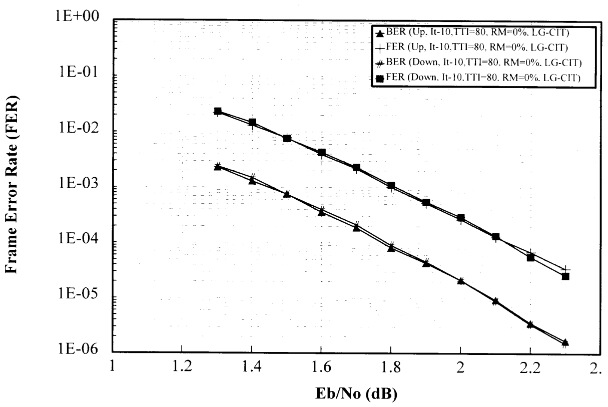
도면 20



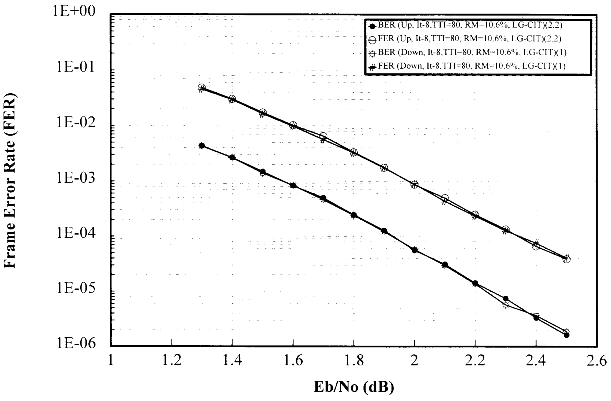
도면 21



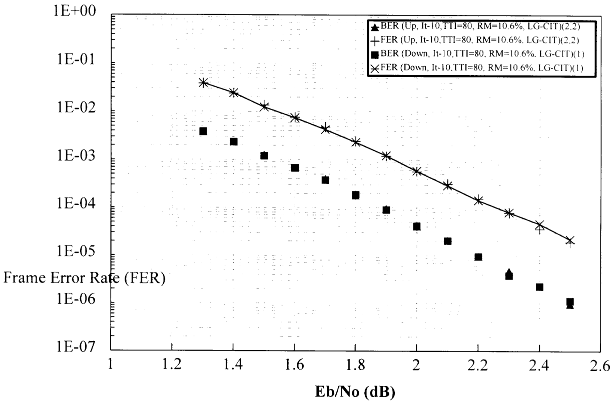
도면 22



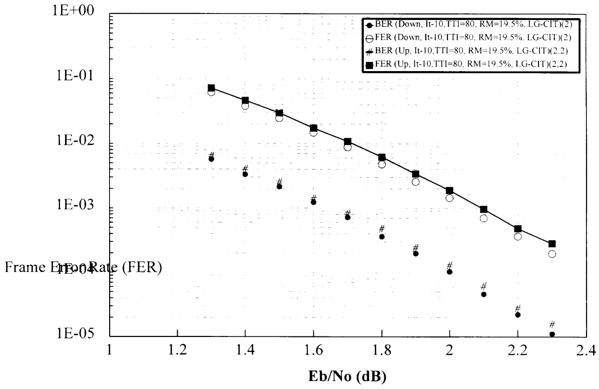
도면 23



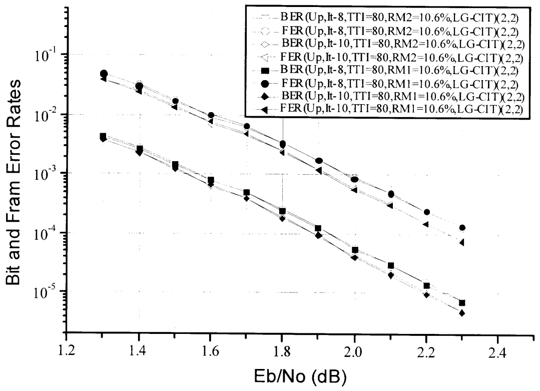
도면 24



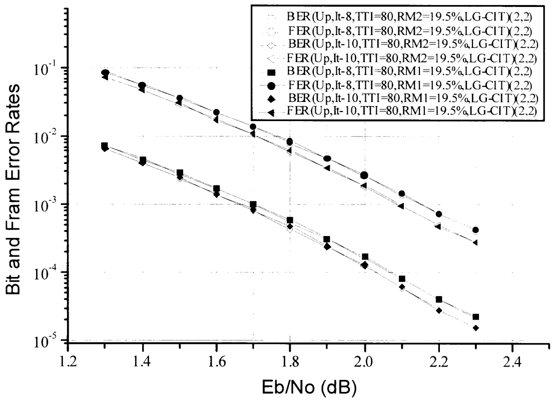
도면 25



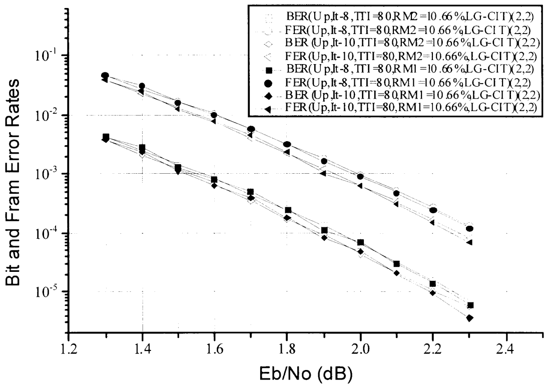
도면 26



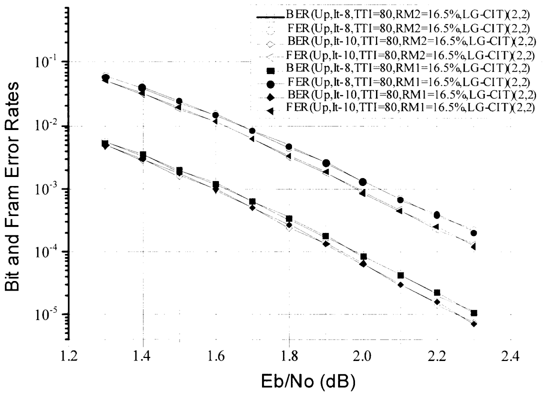
도면 27



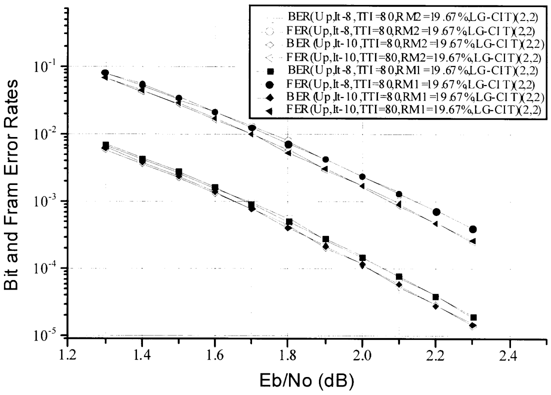
도면 28



도면 29



도면 30


BER(Up,It-8,TTI=80,RM2=19.67%,LG-CIT)(2,2)
FER(Up,It-8,TTI=80,RM2=19.67%,LG-CIT)(2,2)
BER(Up,It-10,TTI=80,RM2=19.67%,LG-CIT)(2,2)
FER(Up,It-10,TTI=80,RM2=19.67%,LG-CIT)(2,2)
BER(Up,It-8,TTI=80,RM1=19.67%,LG-CIT)(2,2)
FER(Up,It-8,TTI=80,RM1=19.67%,LG-CIT)(2,2)
BER(Up,It-10,TTI=80,RM1=19.67%,LG-CIT)(2,2)
FER(Up,It-10,TTI=80,RM1=19.67%,LG-CIT)(2,2)
Bit and Fram Error Rates
10^{-1}
10^{-2}
10^{-3}
10^{-4}
10^{-5}
1.2
1.4
1.6
1.8
2.0
2.2
2.4
Eb/No (dB)