

MATLAB GUI 기반의 ATSC 3.0 시뮬레이터 개발

김시문, 김정창¹, 박성익*, 김홍묵*

한국해양대학교

*한국전자통신연구원

{csp_mong, jchkim}@kmou.ac.kr, {psi76, hmkim}@etri.re.kr,

Development of ATSC 3.0 Simulator Based on MATLAB GUI

Si Moon Kim, Jeongchang Kim, Sung Ik Park*, and Heung Mook Kim*

Korea Maritime and Ocean University

*Electronics and Telecommunications Research Institute

요 약

본 논문에서는 ATSC(Avanced Television Systems Committee) 3.0 물리계층에서 다양한 서비스 시나리오별 가능한 프로파일을 도출하기 위한 MATLAB GUI(graphical user interface)기반의 ATSC 3.0 시뮬레이터를 구현한다. 개발된 시뮬레이터는 ATSC 3.0 에서 지원하는 다양한 종류의 BICM(bit interleaved and coded modulation) 및 OFDM(orthogonal frequency division multiplexing) 파라미터 조합과 다중화 방식을 지원할 수 있고, 그에 따른 프레임 길이와 전송률을 계산할 수 있다. 개발된 시뮬레이터에서는 4 개의 PLP(physical layer pipe)까지 지원할 수 있고, 최대 4 개의 부프레임(subframe)까지 구성이 가능하다.

1. 서론

ATSC 3.0 은 기존 표준들과는 역호환성을 제공하지 않는 차세대 지상파 디지털 TV 방송 표준 규격이다[1]. 또한 ATSC 3.0 물리계층은 UHDTV 서비스를 제공하기 위해 보다 향상된 시스템 전송 용량을 제공하고, 다양하게 요구되는 서비스에 효율적으로 대응하기 위하여 많은 동작 모드와 전송 기술을 제공한다.

ATSC 3.0 물리계층은 OFDM(orthogonal frequency division multiplexing)을 기반으로, 64800 혹은 16200 길이를 가지는 LDPC(low-density parity check) 코드를 사용한다. 또한, 균등(uniform) QPSK 와 16QAM 부터 4096QAM 까지 다섯 가지 비균일 성상(non-uniform constellation)을 사용하고 있다. 다중화 기법으로는 시간을 분할하여 사용하는 TDM(time division multiplexing), 주파수를 분할하여 사용하는 FDM(frequency division multiplexing)을 사용할 뿐만 아니라 송신 전력을 나누어 사용하는 LDM(layered division multiplexing) 방식이 사용될 수 있다. ATSC 3.0 은 이러한 기술들을 이용하여 넓은 동작 SNR(signal to noise ratio) 영역을 제공하는 장점을 가지고 있다. 방송사들은 이러한 장점을 이용하여 서비스를 하고자 하는 대상에 따라 달라질 수 있는 여러 가지 파라미터들을 적절히 선택하여 사용할 수 있다[2].

본 논문에서는 ATSC 3.0 물리계층에서 다양한 서비스

시나리오별 가능한 프로파일을 도출하기 위하여 MATLAB GUI 기반의 시뮬레이터를 개발한다.

2. ATSC 3.0 시뮬레이터 구성

ATSC 3.0 프레임은 부트스트랩, 프리앰블, 부프레임들로 구성된다. 가장 기본적인 프리앰블의 정보는 부트스트랩에 들어 있으며, 부트스트랩은 프리앰블 심볼을 복조하는데 필요한 파라미터를 포함하고 있다. 첫 번째 프리앰블 심볼은 L1-Basic 시그널링을 포함하고 있으며, L1-Detail 시그널링은 L1-Basic 시그널링 이후에 나타나므로 첫 번째 프리앰블 혹은 그 이후 프리앰블 심볼에 걸쳐서 나타날 수 있다. L1-Basic 시그널링은 첫 번째 부프레임의 OFDM 파라미터를 포함하고, L1-Detail 은 나머지 부 프레임의 OFDM 파라미터 및 모든 PLP 에 대한 파라미터를 포함한다.

본 논문에서 개발한 MATLAB GUI 기반의 ATSC 3.0 시뮬레이터는 크게 부트스트랩, 프리앰블, 부프레임의 시그널링 정보를 입력하는 부분과 입력한 정보를 바탕으로 필요한 시그널링 정보 혹은 파라미터를 자동으로 계산하는 부분으로 구성된다. 개발한 MATLAB GUI 기반의 ATSC 3.0 시뮬레이터는 전체 11 개의 탭으로 구성되어 있다. 부트스트랩과 관련된 파라미터를 설정하는 부트스트랩 탭(Bootstrap), 프리앰블에 관련된 파라미터를 설정하는 탭(Preamble), 부프레임과 관련된 파라미터를 설정하는 9 개의 탭(Subframe, Sub1-PLP INFO 1,

¹ Corresponding Author

Sub1-PLP INFO 2, Sub2-PLP INFO 1, Sub2-PLP INFO 2, Sub3-PLP INFO 1, Sub3-PLP INFO 2, Sub4-PLP INFO 1, Sub4-PLP INFO 2)으로 구성되어 있다.

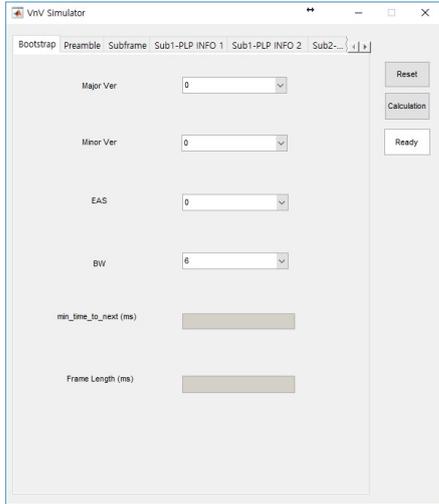


그림 1. 시뮬레이터의 부트스트랩 항목

그림 1 은 개발한 시뮬레이터의 부트스트랩 파라미터 설정 항목을 나타낸다. 부트스트랩의 ‘Major ver’과 ‘Minor ver’은 부트스트랩의 주 버전과 부 버전에 대한 설정이다. 예를 들어 주 버전이 0 이고, 부 버전이 0 이면 부트스트랩의 심볼 수는 4 이다. ‘EAS’는 긴급상황 존재 여부를 나타내는 파라미터이고 ‘BW’는 시스템 대역폭에 대한 파라미터이다.

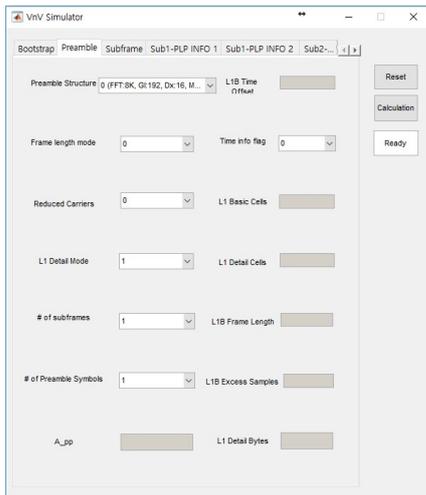


그림 2. 시뮬레이터의 프리앰블 항목

그림 2 는 개발한 시뮬레이터의 프리앰블 파라미터 설정 항목을 나타낸다. 프리앰블은 L1 시그널링 정보와 데이터 페이로드의 프레임 구성에 대한 정보를 담고 있다. L1 시그널링은 매 프레임 마다 물리계층 파라미터 구성을 위해 필요한 정보를 제공하며 L1-Basic 과 L1-Detail 로 구성된다. L1-Basic 의 길이는 200 비트로 고정되어 있으며 L1-Detail 에 필요한 정보가 담겨있다. L1-Detail 의 길이는 설정한 부프레임 및 PLP 의 구성에 따라 다양한 길이를 가질 수 있으며, 부프레임

의 데이터 페이로드를 디코딩하기 위해 필요한 정보를 제공한다. L1-Detail 의 길이는 시뮬레이터에서 계산되어 ‘L1 Detail Bytes’항목에 표시된다. ‘Preamble Structure’는 프리앰블의 FFT(fast Fourier transform) 크기, GI(guard interval) 길이, 프리앰블 파일럿 패턴, L1-Basic FEC(forward error correction) 모드에 따라 다르게 시그널링되고 이는 부트스트랩의 마지막 심볼을 통해 시그널링된다. 그 외에도 시간 정렬(time-aligned) 방식과 심볼 정렬(symbol-aligned) 방식을 결정할 수 있으며, 부프레임의 개수와 프리앰블 심볼의 개수 등도 설정할 수 있다.

그림 3 은 개발한 시뮬레이터의 부프레임 파라미터 설정 항목을 나타낸다. 부프레임의 FFT 크기는 8K(8192), 16K(16384), 32K(32768)로 설정할 수 있고 ‘GI’는 해당 부프레임의 OFDM 심볼에 사용되는 GI 길이를 설정하는 파라미터 값이다. ‘Reduced carrier’는 반송파의 최대 개수를 감소시키기 위한 컨트롤 유닛값을 나타낸다. ‘# of Payload’는 경계 심볼을 포함한 모든 OFDM 심볼의 개수를 나타낸다. ‘SP_Pattern’은 해당 부프레임의 분산 파일럿 패턴을 표시하고 ‘Pilot Boost’는 해당 부프레임에 사용되는 분산 파일럿의 크기를 표시한다. ‘First SBS’(subframe boundary symbol), Last SBS 는 첫 번째와 마지막 심볼이 부프레임 경계 심볼 여부를 나타낸다. ‘# of PLP’는 해당 부프레임의 PLP 개수를 나타내며 본 시뮬레이터에서는 최대 4 개까지 설정할 수 있다.

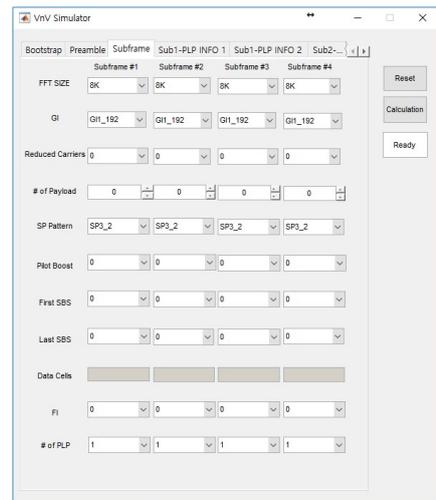


그림 3. 시뮬레이터의 부프레임 항목

그림 4 와 5 는 시뮬레이터의 부프레임에 해당하는 PLP 와 관련된 파라미터값을 설정하는 Sub1-PLP INFO 1, 2 를 나타낸다. 각 PLP 를 구분할 수 있는 ‘PLP ID’는 0 부터 63 까지 설정할 수 있다. 시간 인터리빙과 관련된 설정은 ‘TI_mode’ 항목에서 선택할 수 있으며, CTI(convolutional time interleaver), HTI(hybrid time interleaver), None 중 선택할 수 있다. 컨볼루션 시간 인터리빙(CTI) 을 사용하는 경우 ‘CTI Depth’를 통해 수행하는 인터리버의 길이를 설정할 수 있다. 하이브리드 시간 인터리빙(HTI) 을 사용하는 경우 프레임 당 TI 블록의 개수와 FEC 블록의 최대 개수, FEC 블록의 개수를 설정해야 한다. BICM 과 관련된 파라미터로서 ‘Outer Code’ 항목은 BCH(Bose- Chaudhuri- Hocquenghem), CRC(cyclic redundancy check), None 중에서 선택 가능하며, LDPC 와 관련된 파라미터로는 ‘Codeword’항목에서 코드 길이를 16k(16200), 64k(64800) 중에 선택 가능하다. 부호율은 ‘Code Rate’ 항목을 통해 2/15 에서 13/15 까지 설정할 수 있

다. LDPC 부호는 코드길이 및 부호율에 따라 타입 A 혹은 타입 B의 구조를 갖는다.

구되는 SNR[dB]이 계산된다.

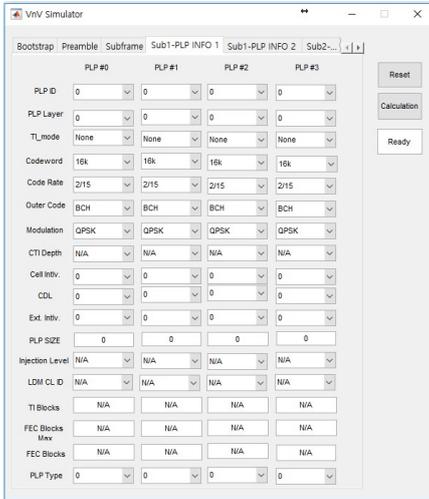


그림 4. 시뮬레이터의 Sub1-PLP INFO 1 항목

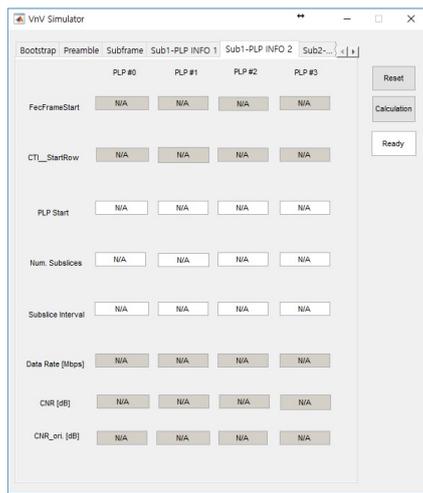


그림 5. 시뮬레이터의 Sub1-PLP INFO 2 항목

그 외에도 HTI의 경우 셀 인터리빙과 CDL(convolutional delay line) 여부를 결정할 수 있다. 또한, 확장 셀 인터리빙 여부 등도 파라미터 설정값으로 입력해줄 수 있다. 셀 다중화의 경우 PLP의 구성에 따라 다양한 방법으로 다중화될 수 있다. 2개 이상의 PLP를 사용하는 경우 TDM, FDM, LDM, TFDm 등 다양한 조합이 가능하고 LDM을 사용하는 경우 ‘PLP Layer’ 항목을 통해 높은 강인성을 요구하는 서비스를 제공하기 위하여 사용되는 CL(core layer)과 높은 전송률을 요구하는 서비스를 제공하기 위해 사용되는 EL(enhanced layer)를 구분할 수 있다 [3]. LDM에서는 ‘Injection Level’ 항목을 통해 삽입 레벨(0dB~25dB)을 설정하여 CL과 EL을 결합한다. 여기서, 여러 개의 PLP가 사용될 경우, EL에 해당하는 PLP가 어떤 CL과 결합되었는지 구분하기 위하여 ‘LDM CL ID’를 사용한다. 그림 5의 Sub1-PLP INFO 2 탭은 셀 다중화에 대한 정보인 ‘PLP Start’, ‘Num. Subslice’, ‘Subslice interval’ 항목들이 설정 가능하고, 부트스트랩 항목부터 부프레임까지 설정한 모든 파라미터에 따른 전송률[Mbps] 및 AWGN 채널에서

3. 결론

본 논문에서는 ATSC 3.0 물리계층에서 다양한 서비스 시나리오별 가능한 프로파일을 도출하기 위한 MATLAB GUI 기반의 시뮬레이터를 개발하였다. 물리계층 단말 프로파일의 경우 시스템 파라미터에 따라 다양한 전송률, 에러 성능, 방송망 커버리지 등의 조합이 가능하다. 개발된 시뮬레이터에서는 부프레임당 최대 4개의 PLP까지 지원할 수 있고, LDM까지 적용할 수 있어 다양한 서비스에 따른 적합한 단말 프로파일을 도출하는데 활용할 수 있다.

ACKNOWLEDGMENT

본 연구는 미래창조과학부 및 정보통신기술진흥센터의 정보통신·방송 연구개발 사업의 일환으로 수행되었음. [Development of Transmission Technology for Ultra High Quality UHD, 2017-0-00081].

참고문헌

- [1] ATSC, “ATSC Standard: Physical Layer Protocol,” Doc.A/322, Feb. 2017.
- [2] 권선형, 박성익, 이재영, 임보미, 김홍목, "ATSC 3.0 물리계층 표준기술," 방송과 미디어 vol. 20, no. 4, pp. 17-27, Oct. 2015.
- [3] 김세우, 오현오, 배병준, 김정창, 김규현, “ATSC 3.0 UHD 지상파 전송방식 규격 고찰” KBS 논문집 R&D, Vol. 20, no. 2, Dec. 2015.