



เทคนิคการมอดูเลตแบบปรับตัวสำหรับเทคโนโลยี WiMAX

Adaptive Modulation in WiMAX Technology

โดย

นายกิตติกร ศุภเสถียรกุล รหัสนักศึกษา B4706212
นายศุภกิจ ดักขวุธ รหัสนักศึกษา B4711971

รายงานนี้เป็นส่วนหนึ่งของการศึกษาวิชา 427499 โครงการวิศวกรรมโทรคมนาคม
และวิชา 427494 โครงการวิศวกรรมโทรคมนาคม
หลักสูตรวิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมโทรคมนาคม หลักสูตรปรับปรุง พ.ศ. 2545
สำนักวิชาวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีสุรนารี
ประจำภาคการศึกษาที่ 2 ปีการศึกษา 2550 และประจำภาคการศึกษาที่ 1 ปีการศึกษา 2551

แบบเสนอโครงการวิชา 427494 โครงการวิศวกรรมโทรคมนาคม
ประจำภาคการศึกษาที่ 1/2551
เรื่อง เทคนิคการมอดูเลตแบบปรับตัวสำหรับเทคโนโลยี WiMAX

ผู้เสนอโครงการ	1. นาย กิตติกร สุภเสถียรกุล	รหัสประจำตัว	B4706212
	2. นาย สุภกิจ ลักษวุธ	รหัสประจำตัว	B4711971
อาจารย์ที่ปรึกษา	อ.ดร. พีระพงษ์ อุซารสกุล		

บทคัดย่อ

ปัจจุบันนี้ความต้องการที่เพิ่มขึ้นในด้านความเร็วของการส่งข้อมูล และความเร็วของผู้ใช้บริการขณะเคลื่อนที่ รวมถึงระยะทางจากสถานีแม่ข่าย เป็นสิ่งผลักดันให้มีการพัฒนาเทคโนโลยีด้านระบบไร้สายแบบแถบความถี่กว้าง หรือที่รู้จักในชื่อ **WiMAX** ในโครงการนี้จึงได้ศึกษา และจำลองการทำงานของเทคโนโลยี WiMAX ด้วยโปรแกรม MATLAB โดยมุ่งเน้นที่จะวิเคราะห์เทคนิคการมอดูเลตแบบปรับตัว ซึ่งจะสามารถพัฒนาสมรรถนะของระบบได้ในที่สุด

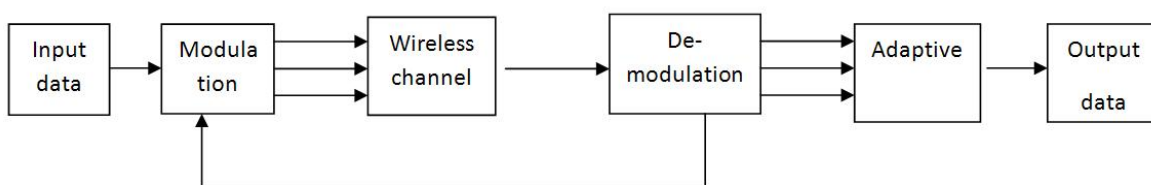
1. หลักการ และเหตุผล

การออกแบบอุปกรณ์WiMAXย่านความถี่2-11 Gbps ได้รับความสนใจอย่างต่อเนื่องจากหลายหน่วยงานทั้งภาครัฐและเอกชน โดยมีความมุ่งหวังที่จะสร้างอุปกรณ์เฉพาะทางที่สามารถใช้งานได้ตามความต้องการของผู้บริโภค สำหรับเทคโนโลยีการสื่อสารไร้สายในอนาคต ปัจจัยหนึ่งที่ต้องพิจารณาคือคุณภาพของสัญญาณที่ได้ในสภาพแวดล้อมต่างๆ รวมทั้งแบบอัตราส่งข้อมูลความเร็วสูง ดังนั้นโครงการนี้จึงต้องการศึกษา และจำลองการทำงานของเทคโนโลยี WiMAX ด้วยโปรแกรม MATLAB โดยมุ่งเน้นที่จะวิเคราะห์เทคนิคการมอดูเลตแบบปรับตัว ซึ่งจะสามารถพัฒนาสมรรถนะของระบบได้ในที่สุด

สำหรับการออกแบบในปัจจุบัน การใช้โปรแกรมสำเร็จรูปถือว่าได้รับความนิยมมาก เพราะค่าที่ได้จากโปรแกรมให้ผลใกล้เคียงกับผลจากการวัดจริงมาก เนื่องจากโปรแกรม MatLab มีในส่วน

ของโปรแกรม Simulink อยู่ และโปรแกรม Simulink นี้มีฟังก์ชันการหาค่าที่เหมาะสมซึ่งทำให้สามารถ ออกแบบอุปกรณ์ที่ซับซ้อนได้ ดังนั้นผู้จัดทำโครงการจึงนำโปรแกรมนี้มาศึกษาการใช้งาน และรวมไป ถึงนำมาออกแบบเพื่อสร้างเป็นอุปกรณ์จริงด้วย

เทคนิคการมอดูเลต Orthogonal Frequency Division Multiplex(OFDM) เป็นเทคนิคการมอดูเลตแบบหลายคลื่นพาห์ (Multiple Carrier Modulation) ซึ่งเป็นรูปแบบของการสื่อสารแบบขนาน นั่นเอง เนื่องจากการสื่อสารแบบไร้สายมักจะประสบกับปัญหาการจางหายของสัญญาณ (Fading) อันมีสาเหตุมาจากการแพร่กระจายคลื่นสัญญาณเป็นหลายวิถี (Multipath propagation) นอกจากนั้นคลื่นหลายวิถียังทำให้เกิดการรบกวนแบบแทรกสอดระหว่างสัญลักษณ์ (Inter Symbol Interference) หรือ ISI ของสัญญาณข้อมูลขึ้นที่เครื่องรับอีกด้วย ซึ่งถ้าเป็นระบบการมอดูเลตและส่งสัญญาณแบบเก่าๆ (การส่งข้อมูลแบบอนุกรม) จะแก้ไขปัญหาลำบากได้ค่อนข้างยาก โดยเฉพาะเมื่อเป็นการสื่อสารที่มี อัตราการส่งข้อมูลสูงๆ แต่สำหรับ OFDM แล้วปัญหาลำบากนี้สามารถแก้ไขได้ง่ายกว่าและยังสามารถส่ง ข้อมูลที่มีอัตราสูงๆ ได้อย่างสบาย เนื่องจากในระบบ OFDM ข้อมูลที่เป็นอนุกรมความเร็วสูงจะถูก แปลงให้เป็นข้อมูลแบบขนานความเร็วต่ำเสียก่อน แล้วจึงส่งออกไปยังเครื่องรับพร้อมๆ กัน ซึ่งสามารถ ลดปัญหาลำบากนี้ลงได้



รูปที่ 1 ไลอะแกรมแสดงการทำงานอย่างง่ายของแบบจำลอง WiMAX

2. วัตถุประสงค์

1. เพื่อศึกษาการใช้งานของ โปรแกรม Simulink
2. เพื่อศึกษาและจำลองแบบตามในมาตรฐาน WiMAX
3. เพื่อศึกษารูปแบบการส่งสัญญาณแบบ Orthogonal Frequency Division Multiplex(OFDM)
4. เพื่อพัฒนาเทคนิคการมอดูเลตแบบปรับตัว สำหรับ WiMAX

3. ขอบเขตงาน

1. ศึกษาและออกแบบจำลองระบบ WiMAX ด้วยโปรแกรม Simulink
2. ออกแบบเทคนิคมอดูเลตแบบปรับตัวสำหรับระบบ WiMAX
3. พัฒนาและวิเคราะห์ระบบ WiMAX เมื่อใช้และไม่ใช้เทคนิคการมอดูเลตแบบปรับตัว
4. เปรียบเทียบผลที่ได้กับทางทฤษฎี และทำการแก้ไขปรับปรุงการออกแบบเพื่อให้ได้ผลดีขึ้น

5.งบประมาณ

1.อุปกรณ์ที่ใช้ในการสร้าง	5500 บาท
2.ค่าใช้จ่ายอื่นๆ	500 บาท
รวม	6000 บาท

6.ผลที่คาดว่าจะได้รับ

1. ได้แบบจำลองเทคนิคการมอดูเลตแบบปรับตัวที่สามารถใช้กับระบบ WiMAX
2. สามารถนำความรู้จากโครงการนี้ไปประกอบอาชีพในอนาคตได้
3. สามารถนำความรู้ทางทฤษฎีมาประยุกต์ใช้ในทางปฏิบัติ
4. สามารถทำงานร่วมกับเพื่อนร่วมงานได้

7.การประเมินผลงาน

- 1.รายงานผลการทำโครงการต่ออาจารย์ที่ปรึกษาตามขั้นตอนที่กำหนดในแผนการปฏิบัติงาน
- 2.การสอบสัมภาษณ์จากอาจารย์ประจำสาขาวิชาวิศวกรรมโทรคมนาคม

ลงชื่อ

(นาย กิตติกร ศุภเสถียรกุล)

ผู้เสนอโครงการ

ลงชื่อ.....

(นาย ศุภกิจ ลักษุฐ)

ผู้เสนอโครงการ

ลงชื่อ (อ. ดร. พีระพงษ์ อุฑารสกุล) อาจารย์ที่ปรึกษาโครงการ	ลงชื่อ..... (ผศ. ดร. ริงสรณ์ ทองทา) หัวหน้าสาขาวิชาวิศวกรรมโทรคมนาคม
------------------------------------------------------------------------	----------------------------------------------------------------------------

โครงการงาน	เทคนิคการมอดูเลตแบบปรับตัวสำหรับเทคโนโลยี WiMAX
โดย	นายกิตติกร ศุภเสถียรกุล นายศุภกิจ ลักษวุธ
อาจารย์ที่ปรึกษา	อาจารย์ ดร.พีระพงษ์ อุฑารสกุล
ภาคการศึกษา	2/2550,1/2551

บทคัดย่อ

โครงการนี้ได้นำเสนอเทคนิคการมอดูเลตแบบปรับตัวสำหรับเทคโนโลยี WiMAX เนื่องจากปัจจุบันนี้ความต้องการที่เพิ่มขึ้นในด้านความเร็วของการส่งข้อมูลและความเร็วของผู้ใช้บริการขณะเคลื่อนที่รวมถึงระยะทางจากสถานีแม่ข่ายเป็นสิ่งผลักดันให้มีการพัฒนาเทคโนโลยีด้านระบบไร้สายแบบแถบความถี่กว้างหรือที่รู้จักในชื่อ **WiMAX** ในโครงการนี้จึงได้ศึกษาและจำลองการทำงานของเทคโนโลยี WiMAX ด้วยโปรแกรมMATLABโดยมุ่งเน้นที่จะวิเคราะห์เทคนิคการมอดูเลตแบบปรับตัว ซึ่งจะสามารถพัฒนาสมรรถนะของระบบได้ในที่สุด

กิตติกรรมประกาศ

โครงการนี้สำเร็จลุล่วงไปได้ด้วยดีส่งผลให้คณะผู้จัดทำโครงการได้รับความรู้และประสบการณ์ต่างๆมากมายในระหว่างที่ศึกษาและจัดทำโครงการนี้ คณะผู้จัดทำได้รับความช่วยเหลือ คำแนะนำและความช่วยเหลืออย่างดีเสมอมา ได้แก่

อาจารย์ ดร.พีระพงษ์ อุฑารสกุล สาขาวิชาวิศวกรรมโทรคมนาคม อาจารย์ที่ปรึกษาโครงการที่คอยให้คำปรึกษาและคำแนะนำในด้านต่างๆ

สุดท้ายนี้คณะผู้จัดทำใคร่ขอขอบพระคุณบุคคลทุกท่านที่ได้ให้โอกาสให้การสนับสนุนและคอยให้กำลังใจกับคณะผู้จัดทำโครงการด้วยดีตลอดมา

นาย กิตติกร สุภเสถียรกุล

นาย นายศุภกิจ ลักษวุธ

สาขาวิชาวิศวกรรมโทรคมนาคม

สำนักวิชาวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีสุรนารี

สารบัญ

หัวข้อ	หน้า
บทคัดย่อ	ก
กิตติกรรมประกาศ	ข
สารบัญ	ค
สารบัญรูปภาพ	จ
บทที่ 1 บทนำ	
1.1 ความเป็นมา	1
1.2 วัตถุประสงค์	2
1.3 ขอบเขตการทำงาน	2
1.4 ขั้นตอนการดำเนินงาน	2
1.5 ผลที่คาดว่าจะได้รับ	3
บทที่ 2 ทฤษฎี	
2.1 การเข้ารหัสช่องสัญญาณ	4
2.2 Physical Layer(PHY)	5
2.3 FDD (Frequency Division Duplex)	21
2.4 การมอดูเลต(Modulation)	22
บทที่ 3 การใช้งานโปรแกรม Simulink	
3.1 ที่มาของโปรแกรม Simulink	34
3.2 แหล่งจ่าย	35
3.3 ภาคการเข้ารหัส	36
3.4 ภาคการถอดรหัส	47
3.5 ภาคส่ง	51
3.6 ภาครับ	60
3.7 ภาคการปรับตัว	66
3.8 ภาคการแสดงผล	70
บทที่ 4 ผลการทดลอง	
4.1 ผลการทดลองการวัด Bit Error Rate กับ SNR	73
4.2 ผลการทดลองการกำหนดเป้าหมายของจำนวนบิตที่ผิดพลาด	74

สารบัญ(ต่อ)

หัวข้อ	หน้า
4.3 ผลการทดลองการวัด Bit Error Rate กับ SNRของแบบจำลองที่มีการปรับตัวที่เป้าหมายของจำนวนบิตที่ผิดพลาดที่ 10^{-2} และแบบจำลองที่ไม่มีการปรับตัว	77
4.4 ผลการทดลองการวัดจำนวนข้อมูลที่ส่งออกของแบบจำลองที่มีการปรับตัวและไม่มีการปรับตัว	78
บทที่ 5 สรุปผลการทดลอง	
สรุปผลการทดลอง	80
ประวัติผู้เขียน	82
บรรณานุกรม	83

สารบัญรูปภาพ

เนื้อหา	หน้า
รูปที่ 1.1 ไดอะแกรมแสดงการทำงานอย่างง่ายของแบบจำลอง WiMAX	3
รูปที่ 2.1 แผนภาพการทำงานมาตรฐาน IEEE 802.16a หรือ WiMAX	4
รูปที่ 2.2 โครงสร้าง PHY Layer และ MAC Layer ของ 802.16e	5
รูปที่ 2.3 โครงสร้าง PHY Layer	6
รูปที่ 2.4 Reed-Solomon Codes	7
รูปที่ 2.5 การเข้ารหัสแบบ Convolution code $K=3$	9
รูปที่ 2.6 State Diagram	12
รูปที่ 2.7 Tree Diagram	13
รูปที่ 2.8 Trellis Diagram	14
รูปที่ 2.9 การใช้งาน Trellis Diagram	15
รูปที่ 2.10 การถอดรหัสแบบ Viterbi Decoding (1)	18
รูปที่ 2.11 การถอดรหัสแบบ Viterbi Decoding (2)	18
รูปที่ 2.12 การถอดรหัสแบบ Viterbi Decoding (3)	19
รูปที่ 2.13 การถอดรหัสแบบ Viterbi Decoding (4)	20
รูปที่ 2.14 แสดงการส่งข้อมูลขาขึ้น(Uplink) และขาลง (Downlink) โดยใช้ FDD	21
รูปที่ 2.15 แสดงเฟรมย่อยของ FDD ในข้อมูลขาลง (Down link)	22
รูปที่ 2.16 แสดงการแปลงฟูเรียร์อย่างรวดเร็วของสัญญาณที่ทำการมอดูเลตกับสัญญาณพาหะย่อย(หนังสือ OFDM เทคโนโลยีการมัลติเพล็กซ์ในระบบสื่อสารยุคใหม่)	25
รูปที่ 2.17 แสดงการมัลติเพล็กซ์ชอนุกรมของสัญญาณบนแกนความถี่	25
รูปที่ 2.18 แสดงบล็อกไดอะแกรมตัวส่ง OFDM	27
รูปที่ 2.19 แสดงบล็อกไดอะแกรมรับ OFDM	27
รูปที่ 2.20 แสดงการเสริมสร้างไซคลิก	28
รูปที่ 2.21 Constellation Diagram ของ 4-QAM (ซ้าย) และ 8-QAM (ขวา) ความแตกต่างของ Phase จะมากกว่า Amplitude	30
รูปที่ 2.22 การผสมสัญญาณของข้อมูลจำนวน 24 บิต ด้วยวิธีการ QAM ในกรณีนี้ Bit Rate มีค่าเป็น 3 เท่าของ Baud Rate	31
รูปที่ 2.23 โครงสร้าง 16-QAM ที่แตกต่างกัน	32

สารบัญรูปภาพ

เนื้อหา	หน้า
รูปที่3.1 บล็อกโคดอะแกรม Bernoulli Binary Generator	35
รูปที่3.2การตั้งค่าบล็อกโคดอะแกรม Bernoulli Binary Generator	35
รูปที่3.3แสดงแผนภาพโคดอะแกรมการเข้ารหัส(Encode)	37
รูปที่3.4แสดงแผนภาพโคดอะแกรมการเข้ารหัสReed-Solomon	38
รูปที่3.5 บล็อกโคดอะแกรม Bit to integer Converter	39
รูปที่3.6การตั้งค่าบล็อกโคดอะแกรม Bit to integer Converter-block	39
รูปที่3.7 บล็อกโคดอะแกรม Zero Pad	40
รูปที่3.8การตั้งค่าบล็อกโคดอะแกรม	40
รูปที่3.9 บล็อกโคดอะแกรม Reed –Solomon encode	41
รูปที่3.10การตั้งค่าบล็อกโคดอะแกรม Reed –Solomon encode	41
รูปที่3.11 บล็อกโคดอะแกรม Selectrows	42
รูปที่3.12การตั้งค่าบล็อกโคดอะแกรม Selectrows	42
รูปที่3.13 บล็อกโคดอะแกรม Integer to bit Converter	43
รูปที่3.14การตั้งค่าบล็อกโคดอะแกรม	43
รูปที่3.15การทำงานของ Convolution encode	44
รูปที่3.16 บล็อกโคดอะแกรม Convolution encode	44
รูปที่3.17การตั้งค่าบล็อกโคดอะแกรม Convolution encode	44
รูปที่3.18 บล็อกโคดอะแกรมPuncture	45
รูปที่3.19การตั้งค่าบล็อกโคดอะแกรมPuncture	45
รูปที่3.20 บล็อกโคดอะแกรม Matrix Interleaver	46
รูปที่3.21การตั้งค่าบล็อกโคดอะแกรม Matrix Interleaver	46
รูปที่ 3.22แสดงขั้นตอนการถอดรหัส	47
รูปที่3.23บล็อกโคดอะแกรม Deinterleaving	47
รูปที่3.24การตั้งค่าบล็อกโคดอะแกรม Deinterleaving	47
รูปที่3.25 บล็อกโคดอะแกรม Inserting zero	48
รูปที่3.26การตั้งค่าบล็อกโคดอะแกรม Inserting zero	48
รูปที่3.27 บล็อกโคดอะแกรม Viterbi decoder	49
รูปที่3.28การตั้งค่าบล็อกโคดอะแกรม Viterbi decoder	49

สารบัญรูปภาพ

เนื้อหา	หน้า
รูปที่ 3.29 แสดงขั้นตอนการถอดรหัสของ Reed-Solomon decoder	50
รูปที่ 3.30 บล็อกไดอะแกรม Modulator	51
รูปที่ 3.31 บล็อกไดอะแกรม PAM-Modulator	51
รูปที่ 3.32 บล็อกไดอะแกรม QAM-Modulator(QPSK)	52
รูปที่ 3.33 บล็อกไดอะแกรม QAM-Modulator	52
รูปที่ 3.34 บล็อกไดอะแกรม PAM-Modulator	53
รูปที่ 3.35 บล็อกไดอะแกรม OFDM Transmitter	53
รูปที่ 3.36 บล็อกไดอะแกรม Multiplex selector	54
รูปที่ 3.37 บล็อกไดอะแกรม Generator polynomial	54
รูปที่ 3.38 บล็อกไดอะแกรม PN sequence generator	55
รูปที่ 3.39 บล็อกไดอะแกรม Unipolar to Bipolar converter	55
รูปที่ 3.40 บล็อกไดอะแกรม Dc carrier	56
รูปที่ 3.41 บล็อกไดอะแกรม Matrix concatenation	56
รูปที่ 3.42 บล็อกไดอะแกรม Zero pad(1)	57
รูปที่ 3.43 บล็อกไดอะแกรม Zero pad(2)	57
รูปที่ 3.44 บล็อกไดอะแกรม Selector	57
รูปที่ 3.45 บล็อกไดอะแกรม IFFT	58
รูปที่ 3.46 บล็อกไดอะแกรม Add cyclic prefix	59
รูปที่ 3.47 บล็อกไดอะแกรม AWGN	59
รูปที่ 3.48 บล็อกไดอะแกรม OFDM Receiver	60
รูปที่ 3.49 บล็อกไดอะแกรม Remove cyclic	60
รูปที่ 3.50 บล็อกไดอะแกรม FFT	61
รูปที่ 3.51 บล็อกไดอะแกรม Frame Status Conversion	61
รูปที่ 3.52 บล็อกไดอะแกรม Remove Zero pad	62
รูปที่ 3.53 บล็อกไดอะแกรม Multiplex selector	62
รูปที่ 3.54 บล็อกไดอะแกรม Demodulator	63
รูปที่ 3.55 บล็อกไดอะแกรม PAM-Demodulator	63
รูปที่ 3.56 บล็อกไดอะแกรม QAM-Demodulator	64

สารบัญรูปภาพ

เนื้อหา	หน้า
รูปที่ 3.57 บล็อกไดอะแกรม QAM-Demodulator(1)	65
รูปที่ 3.58 บล็อกไดอะแกรม QAM-Demodulator(2)	65
รูปที่ 3.59 บล็อกไดอะแกรม Constant	66
รูปที่ 3.60 บล็อกไดอะแกรม AMC	67
รูปที่ 3.61 บล็อกไดอะแกรม Adapt	67
รูปที่ 3.62 บล็อกไดอะแกรม Inverse dB	68
รูปที่ 3.63 บล็อกไดอะแกรม Gain	68
รูปที่ 3.64 บล็อกไดอะแกรม Math	69
รูปที่ 3.65 บล็อกไดอะแกรม Error rate calculation	70
รูปที่ 3.66 บล็อกไดอะแกรม Discrete-Time Scatter Plot Scope(1)	70
รูปที่ 3.67 บล็อกไดอะแกรม Discrete-Time Scatter Plot Scope(2)	70
รูปที่ 3.68 บล็อกไดอะแกรม Spectrum Scope(1)	71
รูปที่ 3.69 บล็อกไดอะแกรม Spectrum Scope(2)	71
รูปที่ 4.1 แบบจำลอง WiMAX จากการใช้โปรแกรม Simulink	72
รูปที่ 4.2 กราฟแสดงความสัมพันธ์ระหว่าง SNR กับ Bit Error Rate	73
รูปที่ 4.3 กราฟแสดงความสัมพันธ์ระหว่าง SNR กับ BER ของการกำหนดเป้าหมายที่ 10^{-2}	74
รูปที่ 4.4 กราฟแสดงความสัมพันธ์ระหว่าง SNR กับ BER ของการกำหนดเป้าหมายที่ 10^{-1}	75
รูปที่ 4.5 กราฟแสดงความสัมพันธ์ระหว่าง SNR กับ BER ของการกำหนดเป้าหมายเฉลี่ย ระหว่าง 10^{-1} และ 10^{-2}	76
รูปที่ 4.6 กราฟแสดงความสัมพันธ์ระหว่าง SNR กับ Bit Error Rate ของแบบจำลองที่มีการ การปรับตัวที่เป้าหมายของจำนวนบิตที่ผิดพลาดที่ 10^{-2} และแบบจำลองที่ไม่มี การปรับตัว	77
รูปที่ 4.7 กราฟแสดงความสัมพันธ์ระหว่าง SNR กับ throughput เมื่อจำนวนบิตเท่ากัน	78
รูปที่ 4.8 กราฟแสดงความสัมพันธ์ระหว่าง SNR กับ throughput เมื่อคิดเป็นเปอร์เซ็นต์	79
รูปที่ 5.1 แสดงการใช้การมอดูเลตแบบต่างๆเมื่อเทียบกับระยะห่างจากสถานีฐาน	81

บทที่ 1

บทนำ

1.1 ความเป็นมา

ปัจจุบันนี้ความต้องการที่เพิ่มขึ้นในด้านความเร็วของการส่งข้อมูล และความเร็วของผู้ใช้บริการขณะเคลื่อนที่ รวมถึงระยะทางจากสถานีแม่ข่าย เป็นสิ่งผลักดันให้มีการพัฒนาเทคโนโลยีด้านระบบไร้สายแบบแถบความถี่กว้าง หรือที่รู้จักในชื่อ WiMAX เป็นชื่อเรียกเทคโนโลยีไร้สายรุ่นใหม่ล่าสุดที่คาดหมายกันว่า จะถูกนำมาใช้งานในอนาคตอันใกล้นี้ โดย WiMAX เป็นชื่อย่อของ Worldwide Interoperability for Microwave Access ซึ่งเป็นเทคโนโลยี Broadband ไร้สายความเร็วสูงรุ่นใหม่ที่ถูกพัฒนาขึ้นมาจากมาตรฐาน IEEE 802.16 ซึ่งมาก็ได้พัฒนา มาตรฐาน IEEE 802.16a ขึ้น โดยได้การอนุมัติออกมาเมื่อเดือนมกราคม 2004 โดยสถาบันวิศวกรรมไฟฟ้าและอิเล็กทรอนิกส์ หรือ IEEE (Institute of Electrical and Electronics Engineers)

WiMAX จะเปลี่ยนโฉมหน้าของการติดต่อสื่อสาร Wimax ตอบสนองความต้องการของการเชื่อมต่อเครือข่ายอินเทอร์เน็ตให้กับพื้นที่ที่ห่างไกล ที่สายเคเบิลไม่สามารถลากไปไม่ถึงได้เป็นอย่างดี ตลอดจนเพิ่มความสะดวกสบาย และประหยัดสำหรับการขยายเครือข่ายในเมืองที่มีอยู่แล้วได้ เนื่องจากไม่ต้องลงทุน ขุดถนนเพื่อวางสายเคเบิลใยแก้วใหม่ นอกจากนี้ WiMAX หรือ Broadband ไร้สาย มาตรฐาน IEEE 802.16a ยังได้รับการปรับปรุงประสิทธิภาพของคุณภาพในการให้บริการ (QoS) ซึ่งสามารถรองรับการใช้ การใช้งานภาพ (video) หรือการใช้งานเสียง (voice) ซึ่งไม่จำเป็นต้องใช้ทรัพยากรของเครือข่ายมากอย่างเท่า (low-latency network) อีกทั้งในเรื่องของความปลอดภัยยังได้เพิ่มคุณสมบัติของความเป็นส่วนตัว (privacy)

WiMAX ในโครงการนี้จึงได้ศึกษา และจำลองการทำงานของเทคโนโลยี WiMAX ด้วยโปรแกรม MATLAB โดยมุ่งเน้นที่จะวิเคราะห์เทคนิคการมอดูเลตแบบปรับตัว ซึ่งจะสามารถพัฒนาสมรรถนะของระบบได้ในที่สุด

1.2. วัตถุประสงค์

1. เพื่อศึกษาการใช้งานของโปรแกรม Simulink
2. เพื่อศึกษาและจำลองแบบตามในมาตรฐาน WiMAX
3. เพื่อศึกษารูปแบบการเข้ารหัสสัญญาณแบบ รหัสแบบรีด - โซโลมอน (Reed Solomon Encoder) และการเข้ารหัสแบบ คอนวลูโวลชัน (Convolution)
4. เพื่อพัฒนาเทคนิคการมอดูเลตแบบปรับตัว สำหรับ WiMAX

1.3 ขอบเขตงาน

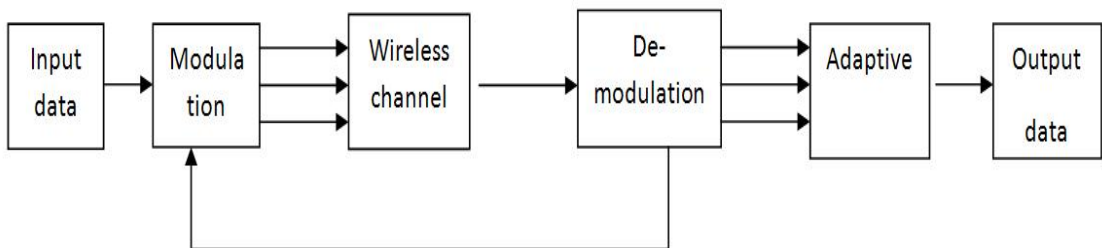
1. ศึกษาและออกแบบจำลองระบบ WiMAX ด้วยโปรแกรม Simulink
2. ออกแบบเทคนิคมอดูเลตแบบปรับตัวสำหรับระบบ WiMAX
3. พัฒนาและวิเคราะห์ระบบ WiMAX เมื่อใช้และไม่ใช้เทคนิคการมอดูเลตแบบปรับตัว
4. เปรียบเทียบผลที่ได้กับทางทฤษฎี

1.4 ขั้นตอนการดำเนินงาน

1. ศึกษาการใช้โปรแกรม Simulink
2. ศึกษาเทคนิคการมอดูเลตของ WiMAX
3. ออกแบบวงจรแบบจำลอง WiMAX ขั้นต้น
4. ศึกษารูปแบบไดอะแกรมต่างๆในการใช้งาน
5. นำไดอะแกรมมาต่อเป็นระบบการส่งสัญญาณอย่างง่าย
6. ต่อไดอะแกรมแบบการมอดูเลตสัญญาณต่างๆที่เกี่ยวข้องกับ WiMAX
7. นำไดอะแกรมของการมอดูเลตแบบ OFDM มาใช้ในการพัฒนา
8. ทำการวัดผลจากโปรแกรม Simulink
9. จัดทำคู่มือในการสร้างไดอะแกรม WiMAX และวิธีใช้งานโปรแกรม simulink
10. นำเสนอโครงการ

1.5.ผลที่คาดว่าจะได้รับ

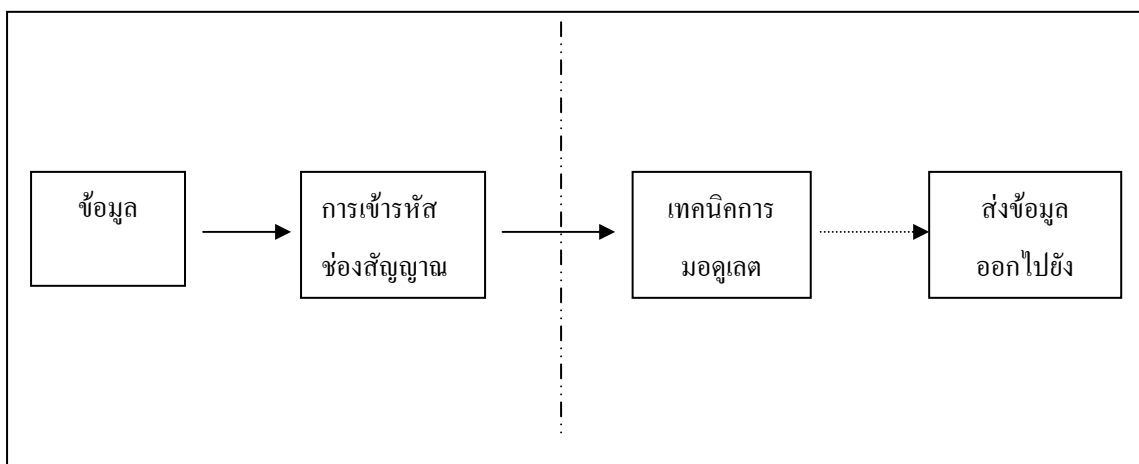
1. ได้แบบจำลองเทคนิคการมอดูเลตแบบปรับตัวที่สามารถใช้กับระบบ WiMAX
2. สามารถนำความรู้จากโครงการนี้ไปประกอบอาชีพในอนาคตได้
3. สามารถนำความรู้ทางทฤษฎีมาประยุกต์ใช้ในทางปฏิบัติ
4. สามารถทำงานร่วมกับเพื่อนร่วมงานได้



รูปที่ 1.1 ไคอะแกรมแสดงการทำงานอย่างง่ายของแบบจำลอง WiMAX

บทที่ 2 ทฤษฎี

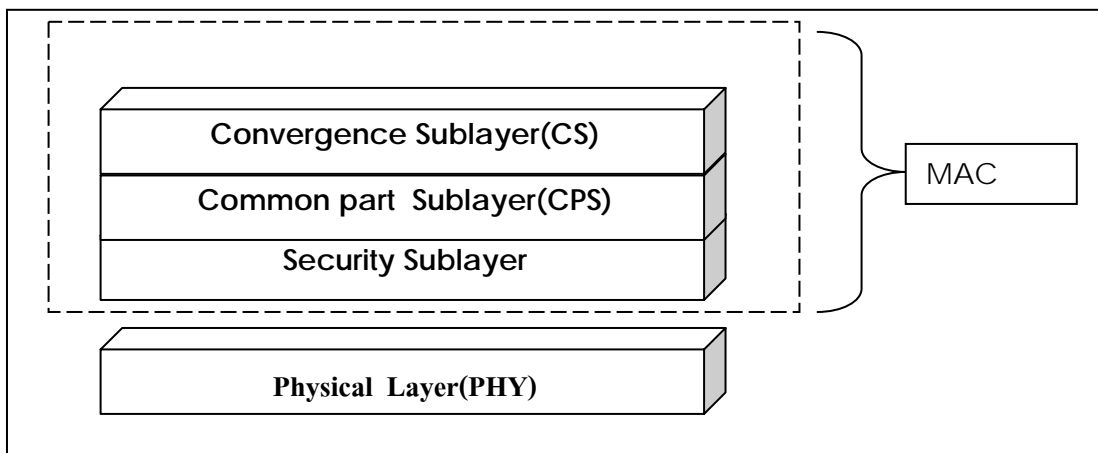
โดยมาตรฐาน IEEE 802.16a หรือ WiMAX โดยสถาบันวิศวกรรมไฟฟ้าและอิเล็กทรอนิกส์ หรือ IEEE (Institute of Electrical and Electronics Engineers) โดยมาตรฐาน IEEE 802.16a หรือ WiMAX มีความสามารถในการส่งกระจายสัญญาณในลักษณะจากจุดเดียวไปยังหลายจุด (Point-to-multipoint) ได้พร้อมๆ กัน โดยมีความสามารถรองรับการทำงานในแบบ Non-Line-of-Sight ได้ สามารถทำงานได้แม้กระทั่งมีสิ่งกีดขวาง โดยจะสามารถแบ่งออกได้เป็น 2 ชั้นตอนคือ



รูปที่ 2.1 แผนภาพการทำงานมาตรฐาน IEEE 802.16a หรือ WiMAX

2.1. การเข้ารหัสช่องสัญญาณ(Channel Coding)

จากพัฒนาการของ IEEE 802.16 จนกระทั่งมาถึง 802.16e ดังที่ได้กล่าวถึงไปแล้วนั้น โปรโตคอล 802.16 ได้ถูกพัฒนาขึ้นมาด้วยตามลำดับ อย่างไรก็ตามโปรโตคอลหลัก ๆ ที่ได้กำหนดไว้ใน IEEE 802.16 ก็ยังคงมีอยู่เพียง 2 เลเยอร์ ก็คือ Physical Layer หรือที่นิยมเรียกกันสั้น ๆ ว่า PHY และ Media Access Control Layer หรือ MAC Layer เท่านั้นทั้ง สองเลเยอร์จะมีหน้าที่ต่างกัน อย่างที่เราพอจะทราบกันคร่าว ๆ นั่นคือ ชั้น PHY จะเป็นชั้นที่ว่าด้วยรายละเอียดทางกายภาพ ลักษณะการรับส่งสัญญาณต่าง ๆ ไม่ว่าจะเป็นเรื่องการควบคุมกำลังการรับส่ง การมอดูเลชัน การทำมัลติเพล็กซ์สำหรับหลายยูสเซอร์ การเข้ารหัสต่าง ๆ ลักษณะของเสาอากาศที่ใช้ เป็นต้น ส่วนชั้น MAC จะว่าด้วยเรื่องการเข้าถึงระบบ การควบคุมรักษาและการตรวจสอบความปลอดภัยต่าง ๆ การเชื่อมโยงเข้ากับโปรโตคอลต่าง ๆ ที่สูงกว่า เป็นต้น



รูปที่ 2.2 โครงสร้าง PHY Layer และ MAC Layer ของ 802.16e

2.2 Physical Layer(PHY)

PHY เป็นเลเยอร์ที่เทียบเท่ากับเลเยอร์ Physical ของ OSI โดยในมาตรฐานของ 802.16 ทั้งหมดนี้จะมีการกำหนด PHY เลเยอร์ทั้งหมด 5 แบบด้วยกันดังต่อไปนี้

1) **Wireless MAN SC** จะเป็นลักษณะของ Single Carrier ซึ่งเป็น 802.16 ตัวแรกสุดที่ได้กำหนดขึ้น ความถี่ที่ใช้งานจะอยู่ในช่วงที่สูงกว่า 11 GHz สามารถที่จะทำงานได้ทั้งแบบ FDD หรือ TDD และไม่มีการใช้งาน OFDM

2) **Wireless MAN SCa** ได้ปรากฏอยู่ใน 802.16a เป็นการใช้งานแบบ Single Carrier เช่นกัน จึงไม่มีการใช้งาน OFDM โดยใช้งานกับความถี่ 2-11 GHz ที่ให้บริการแบบ point to Multipoint และมีทั้งแบบ FDD และ TDD นับเป็นจุดเริ่มต้นของการใช้ไวแมกซ์แบบ Last Mile เป็นครั้งแรกเพื่อรองรับผู้ใช้งานทั่วไป

3) **Wireless MAN OFDM** เป็นการเพิ่มความสามารถของ OFDM เข้าไปใน 802.16a และใช้เป็นรากฐานจนถึงปัจจุบันที่เป็น 802.16e โดยการใช้งาน FFT ขนาด 256 เพื่อรองรับการใช้งานแบบ NLOS และแบบ Point to Multipoint ที่ความถี่ 2-11 GHz มันสามารถที่จะใช้ได้ทั้ง FDD หรือ TDD และออกมาใช้งานกันใน 802.16d เป็นครั้งแรก จึงอาจจะเป็นที่รู้จักกันในชื่อ Fixed WiMAX เพราะ 802.16d ยังให้บริการแบบไม่เคลื่อนที่อยู่นั่นเอง แต่จริง ๆ แล้วมันก็สามารถที่จะให้บริการแบบเคลื่อนที่ได้ ดังจะพบใน 802.16e และรองรับการเชื่อมต่อแบบ Mesh ได้อีกด้วย

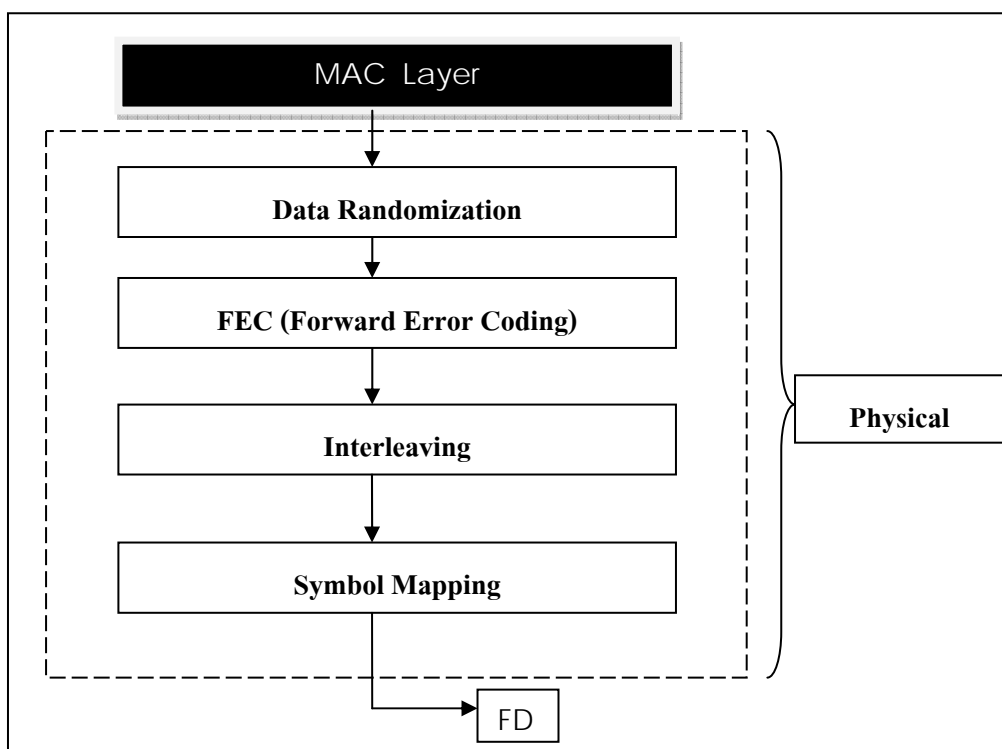
4) **Wireless MAN OFDMA** เป็นผลงานของ 802.16a อีกเช่นกัน โดยจะมีขนาด FFT เท่ากับ 2048 ใช้งานความถี่ 2-11 GHz รองรับได้ทั้ง FDD และ TDD และรองรับการใช้งานแบบ

เคลื่อนที่ได้ด้วย และในมาตรฐาน 802.16e ของปี ค.ศ. 2005 นั้นก็ได้มีการปรับเปลี่ยนรูปแบบจาก OFDMA ปกติเป็น SOFDMA นั่นคือ สามารถปรับเปลี่ยนขนาด FFT ได้ตั้งแต่ 256, 512, 1024 ไปจนถึง 2048 โดยการกำหนดช่วงห่างระหว่าง subcarrier ไว้ให้คงที่นั่นเอง ซึ่งจะทำให้มีความยืดหยุ่นไปตามสภาพของระบบ แบนด์วิดท์ที่มีให้ และสภาวะแวดล้อมต่าง ๆ ได้ดีขึ้น เหมาะสมกับการใช้งานแบบเคลื่อนที่ได้ และนั่นจึงทำให้ 802.16e ได้รับความสนใจเป็นอย่างสูงนั่นเอง

5) **Wireless MAN Human** เป็นผลงานของ 802.16b ซึ่งไม่ได้มีการใช้งานในปัจจุบัน มันทำงานได้ในแบบ TDD เท่านั้น และใช้งาน OFDM หรือ OFDMA ได้ด้วย

โดยในการศึกษาครั้งนี้เราจะเน้นไปที่ **Wireless MAN OFDM** เป็นการเพิ่มความสามารถของ OFDM เข้าไปใน 802.16a โดยการใช้งาน FFT ขนาด 256 โดยจะใช้ FDD เป็นหลัก

2.2.1 ขั้นตอนการสร้าง Physical Layer



รูปที่ 2.3 โครงสร้าง PHY Layer

1) Data Randomization

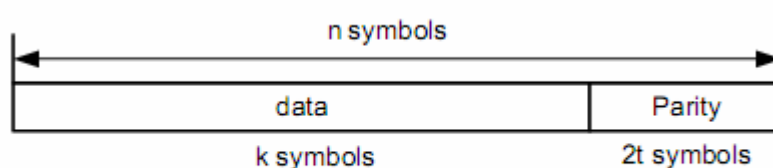
ในส่วน นี้จะเปรียบเสมือนเป็นการเข้ารหัส(Encryption) ในระดับเลเยอร์แรกของข้อมูลทั้ง คิวแวลู (จากสถานีฐานไปยังเครื่องลูกข่าย) และอัปลิงค์ (จากเครื่องลูกข่ายไปที่สถานีฐาน) เพื่อ สร้างความปลอดภัยจากการดักฟังข้อมูลต่าง ๆ หลังจากได้ข้อมูลจากการ Randomization แล้วก็จะ ส่งต่อไปยังส่วนของ Channel Coding หรือ FEC

2) FEC (Forward Error Coding)

เพื่อ เป็นการเพิ่มความน่าเชื่อถือและประสิทธิภาพของการรับส่งข้อมูล จึงต้องมีการเพิ่ม ความสามารถที่จะตรวจจับปัญหาหรือแก้ไขความผิดพลาดในการรับ ส่งข้อมูล และใน 802.16e นั้น ก็ได้ใช้การเข้ารหัสมาช่วยเหลือในส่วนนี้ ซึ่งในส่วนของ Channel Coding นั้นจะประกอบด้วย การเข้ารหัสหลาย ๆ แบบที่มีวัตถุประสงค์และการใช้งานแตกต่างกัน โดยการใช้งานหลักคือ

2.1) รหัสแบบรีด - โซโลมอน (Reed Solomon Encoder)

รหัสนี้เป็นรหัสส่วนที่ไม่เป็นไบนารี(NonbinaryCyclicCodes)ในส่วนสัญลักษณ์ (Symbols) ทำขึ้น บนลำดับ m-bit โดยที่ m เป็นจำนวนเต็มบวกมีค่ามากกว่า 2 รหัส R-S(n, k) กระทำบน สัญลักษณ์ m-bit สำหรับทุก n และ k จะมีค่า $0 < k < n < 2^m + 2$ ที่ k คือ จำนวนสัญลักษณ์ของข้อมูลเริ่มต้นในการเข้ารหัส และ n คือ จำนวนรวมของสัญลักษณ์รหัสในบล็อกการเข้ารหัส สำหรับ รูปแบบการเข้ารหัส R-S(n, k) ที่ได้ เป็น $(n, k) = (2^m - 1, 2^m - 1 - 2t)$ ที่ t คือ สัญลักษณ์ความผิดพลาดที่สามารถแก้ไขได้ของรหัส และ $n - k = 2t$ เป็นจำนวนสัญลักษณ์ของรหัสแก้ไข (Parity)



R-S (n, k) code

รูปที่ 2.4 Reed-Solomon Codes

สำหรับรหัส R-S นี้มีระยะห่าง (Distance) ระหว่าง 2 Codeword จะกำหนดที่จำนวนสัญลักษณ์ที่แตกต่างในลำดับ จะได้ระยะห่างต่ำสุด (Minimum Distance) เป็น $d_{\min} = n - k + 1$ รหัสเหมาะแก่การตรวจ แก้ไขความผิดพลาดเท่ากับ t หรือข้อผิดพลาดที่น้อยกว่า t

สามารถแสดงดังนี้

$$t = \left\lceil \frac{d_{\min} - 1}{2} \right\rceil = \left\lceil \frac{n - k}{2} \right\rceil$$

2.2) Convolution Code

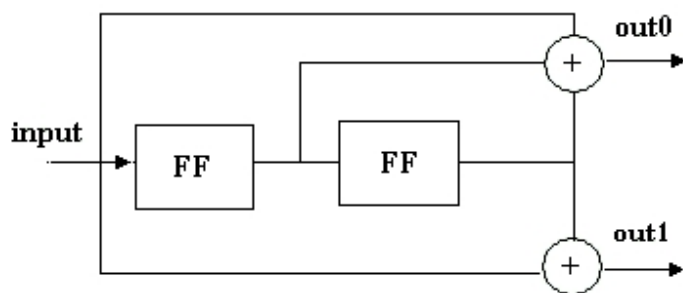
การเข้ารหัสแบบ Convolution ได้รับการพัฒนาขึ้นเมื่อปี 1955 โดย Elias ซึ่งเป็นวิธีการป้องกันการผิดพลาดล่วงหน้า (Forward Error Correction : FEC) เป็นเทคนิครูปแบบหนึ่งที่ใช้แก้ไขการผิดพลาดของข้อมูลที่ได้รับได้จากสัญญาณรบกวนให้ถูกต้องที่มีประสิทธิภาพ ทำให้ข้อมูลที่ถูส่งไปในระบบสื่อสารนั้นมีความน่าเชื่อถือมากขึ้น โดยโครงสร้างพื้นฐานของการเข้ารหัสแบบ Convolution มีคุณลักษณะที่แตกต่างไปจากรหัสบล็อกที่ได้อธิบายไว้ข้างต้น ในกรณีของรหัสบล็อก บิตข้อมูลที่จะนำมาเข้ารหัสจะได้รับการแบ่งออกเป็นบล็อกที่มีขนาดตายตัวเท่ากันทุกบล็อกก่อน จากนั้นจึงนำบิตข้อมูลของแต่ละบล็อกไปผ่านกระบวนการเข้ารหัสเพื่อให้ได้เป็นรหัสที่มีความยาวมากขึ้น โดยความยาวของรหัสที่ได้ จะมีขนาดคงที่ ตายตัวเท่ากันหมดเช่นกัน ส่วนเข้ารหัสแบบ Convolution ไม่จำเป็น ต้องมีการแบ่งบิตข้อมูลเป็นบล็อก แต่อย่างไร เราสามารถป้อนชุดบิตข้อมูลขนาดความยาวเท่าใดก็ได้เข้าสู่วงจรเข้ารหัสแบบ Convolution และผลลัพธ์ที่ได้จากการเข้ารหัสจะเป็นรหัสที่มีความยาวเป็น จำนวนเท่าของบิตข้อมูลที่ป้อนเข้า

ส่วนวิธีการถอดรหัส Convolution มีอยู่หลายแบบด้วย แต่วิธีการถอดรหัสแบบ Viterbi ซึ่งเป็นวิธีที่มีการนิยมใช้งานมากที่สุด การถอดรหัสแบบ Viterbi ค้นพบโดย Viterbi ในปี 1967 ซึ่งจะมีลักษณะ การทำงานของ Viterbi เป็น แบบ Maximum-likelihood decoding algorithm ซึ่งหมายถึงในการทำงาน ของการถอดรหัสนั้น จะทำการค้นหาเส้นทางของข้อมูลใน Trellis Diagram ที่มีลักษณะที่ใกล้เคียงกับ ข้อมูลที่ส่งมามากที่สุด

การเข้ารหัสแบบ Convolution และ การถอดรหัสแบบ Viterbi นี้ได้ถูกนำมาใช้งานในระบบสื่อสารแบบดิจิทัลในปัจจุบันอย่างแพร่หลาย เช่น ระบบสื่อสารดาวเทียม ระบบโทรศัพท์เคลื่อนที่ และระบบ Wireless LAN หรือในระบบจัดเก็บข้อมูล เช่น ฮาร์ดดิสก์ เป็นต้น

2.2.1) การเข้ารหัสแบบ Convolution

การเข้ารหัสข้อมูลแบบ Convolution เป็นการความสัมพันธ์ระหว่างข้อมูลอินพุตเรียงลำดับอย่างต่อเนื่อง โดยข้อมูลเข้ามาผ่านตัว shift register (flip-flop) และ Modulo-2 adder (exclusive or) การหาเอาต์พุตของภาคเข้ารหัสจะทำโดยนำข้อมูลที่อยู่ใน shift register บวกแบบ Modulo-2 adder ซึ่งจะมีลักษณะของวงจรแสดงดังรูปที่ 2.5 จากวงจรเข้ารหัสจะมีอัตราการเข้ารหัส (Rate) เท่ากับ $\frac{1}{2}$ และค่า Constraint Length (K) เท่ากับ 3 โดยจะใช้ Generator polynomial เพื่อแสดงตำแหน่งของใน shift - register ที่จะนำมาหาค่าเอาต์พุตโดยการบวกแบบ Modulo-2 คือ g_0 เท่ากับ 58 และ g_1 เท่ากับ 78



รูปที่ 2.5 การเข้ารหัสแบบ Convolution code K=3

จากรูปที่ 2.5 จะเป็นตัวอย่างของวงจรเข้ารหัสข้อมูลแบบ Convolution code โดยในการทำงาน นั้นจะมีการดึงข้อมูลที่จะทำการเข้ารหัสมาครั้งละ 1 บิต ($k = 1$) เข้ามาภายในวงจร ซึ่งจะทำให้ข้อมูลที่อยู่ในตำแหน่งต่างๆ ของ shift-register จากนั้นถูกเลื่อนไปอยู่ในตำแหน่งถัดไป ต่อจากนั้นจะมีการนำ ข้อมูลทั้งหมดที่เก็บไว้ใน shift-register มาทำการคำนวณเพื่อหาเอาต์พุตจำนวน 2 บิต ($n = 2$) ซึ่งอัตรา การเข้ารหัสเท่ากับ $\frac{1}{2}$ ($Rate = k/n = 1/2$) ที่จะเป็นผลลัพธ์ส่งออกไปจากภาคเข้ารหัสข้อมูล ส่วนค่า จำนวนของ shift register และอินพุตข้อมูลที่จะนำมาบวกแบบ Modulo-2 เรียกว่าค่า Constraint Length (K) โดย จะใช้ Generator polynomial เพื่อแสดงตำแหน่งของใน shift register ที่จะนำมาหาค่าเอาต์พุตโดยการ บวกแบบ Modulo-2 ในการแสดงลักษณะของวงจรเข้ารหัส โดยสำหรับกรณีของวงจรในรูปที่ 2.5 นั้น จะมี Constraint Length = 3 และ Generator polynomial $g_0 = 58$ และ $g_1 = 78$

การวิเคราะห์การทำงานของวงจรเข้ารหัส

จากวงจรเข้ารหัสแบบ Convolution ที่มีการรับข้อมูลที่จะทำการเข้ารหัสเข้ามาภายในวงจรครั้งละ 1 บิต จากนั้นจึงทำการคำนวณหาค่าของ Codeword ที่จะถูกส่งออกไปเป็นผลลัพธ์ของภาคเข้ารหัสจำนวน 2 บิต โดยที่ในการคำนวณหาค่า Codeword ในแต่ละครั้งนั้นจะมีการนำข้อมูลที่อยู่ใน shift register จำนวน 3 บิต (K) มาใช้ในการคำนวณ ซึ่งในการวิเคราะห์การทำงานของวงจรเข้ารหัสนั้น จะมีการแทนการทำงานต่างๆ ของวงจรด้วยตัวแปรที่เรียกว่า Generator polynomial ซึ่งจะเป็นตัวแปรที่ใช้แสดงถึงลักษณะของการคำนวณหาผลลัพธ์ในการเข้ารหัสของ O/P แต่ละตัว โดยในกรณีของวงจรเข้ารหัสตัวอย่างในรูปที่ 2.5 นั้น จะสามารถแสดงการทำงานต่างๆ ของวงจรได้ด้วย Generator polynomial ต่อไปนี้

$$g1 = [1 \ 1 \ 1]$$

$$g2 = [1 \ 0 \ 1]$$

ซึ่งในการคำนวณหาค่าของข้อมูลที่ได้หลังจากการเข้ารหัสนั้น จะสามารถนำข้อมูลที่ได้จากค่าของ Generator polynomial มาทำการคำนวณหาค่าของผลลัพธ์ที่ได้จากการเข้ารหัส โดยการนำข้อมูลที่จะทำการเข้ารหัสมาทำการ Convolution กับค่าของ Generator polynomial ดังสมการ

$$O/P = [\text{input}] * g_i$$

$$O/P1 = [1 \ 0 \ 0 \ 1] * [1 \ 0 \ 1] = 1 \ 0 \ 0 \ 1 \ 0 \ 0$$

$$O/P1 = [1 \ 0 \ 0 \ 1] * [1 \ 1 \ 1] = 1 \ 1 \ 1 \ 1 \ 1 \ 1$$

ดังนั้นค่าของ Codeword ที่ได้จากการเข้ารหัสข้อมูลนั้น จะเกิดจากการคูณข้อมูลที่คำนวณได้จาก O/P1 และตามด้วย O/P2 ซึ่งจะมีค่าเท่ากับ 11 01 01 11 01 01 ซึ่งจากการที่การคำนวณต่างๆ ในการหาผลลัพธ์ของวงจรเข้ารหัสนั้น สามารถที่จะคำนวณได้จากการหาค่า Convolution ดังนั้นจึงมีการเรียก รูปแบบในการเข้ารหัสข้อมูลในลักษณะนี้ว่าเป็นการเข้ารหัสแบบ Convolution Codes และเมื่อนำมาเขียนรวมกันจะได้เป็น Generator polynomial ของวงจรเข้ารหัสจะมีลักษณะดังนี้

$$G = \begin{pmatrix} 1 & 0 & 1 \\ 1 & 1 & 1 \end{pmatrix}$$

โดยในบางกรณีนั้น จะมีการแทนค่าของ Generator polynomial ด้วยตัวเลขฐาน 8 เพื่อแสดงถึง ข้อมูลที่อยู่ใน Generator polynomial โดยในกรณีของวงจรตัวอย่างนั้นจะมีการแทนค่าของ Generator polynomial ด้วยตัวเลข 58 และ 78 และสำหรับข้อมูลในการเข้ารหัสข้อมูลนั้น จะมีการแบ่งข้อมูลที่จะทำการเข้ารหัสออกเป็น n ชุด โดยที่แต่ละชุดนั้นมีขนาดเท่ากับ k_0 บิต ซึ่งในการทำงานนั้นจะมีการดึง ข้อมูลเข้ามาภายในวงจรในแต่ละครั้งนั้น จะมีการนำข้อมูลจำนวน k_0 บิตมาทำการประมวลผลเพื่อหา ค่า codeword จำนวน n_0 บิตเพื่อส่งออกไปจากวงจร โดยที่จะมีการทำงานในลักษณะนี้เรื่อยๆ จนกระทั่งข้อมูลทุกบิตถูกนำมาเข้ารหัสทั้งหมดและผลลัพธ์ที่ออกจากวงจรเข้ารหัสจะกลับเข้าสู่สถานะ ข้อมูลทั้งหมดเป็น 0 จึงจะถือว่าเป็นการสิ้นสุดการเข้ารหัสข้อมูล ดังนั้นสำหรับการเข้ารหัสแบบ Convolution codes นั้นจะมีค่าอัตราการเข้ารหัส (Code Rate) เท่ากับ

$$R = \frac{nk_0}{(n+L-1)n_p}$$

และในกรณีที่ค่า n มีค่ามากกว่าค่า L มากจะได้ว่าค่าอัตราการเข้ารหัสนั้นจะมีค่าเท่ากับ

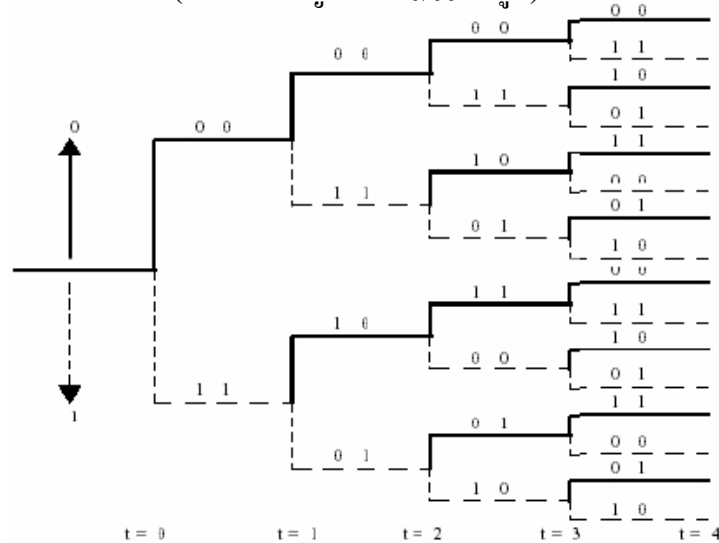
$$R = \frac{k_0}{n_0}$$

ในการวิเคราะห์การทำงานต่างๆของวงจรเข้ารหัสแบบ Convolution นั้น นอกจากจะมีการใช้สมการของ Generator polynomial ในการแสดงลักษณะของวงจรเข้ารหัสแล้ว จะมีวิธีการแสดงการทำงานต่าง ๆ ของวงจรเข้ารหัส โดยการใช้รูปภาพแสดงความสัมพันธ์ระหว่างข้อมูลที่ถูกเก็บไว้ในวงจรเข้ารหัสข้อมูลที่ถูกป้อนเข้ามา ณ เวลานั้น และ ค่าของ Codeword ที่จะถูกส่งออกไป เมื่อมีข้อมูลในกรณีต่าง ๆ ป้อนเข้ามา ซึ่งรูปแบบของภาพที่ใช้แสดงการทำงานของวงจรเข้ารหัสนั้น จะ

มีอยู่ด้วยกัน 3 รูปแบบด้วยกัน ได้แก่ State Diagram, Tree Diagram และ Trellis Diagram ซึ่งจะมีรายละเอียด ดังต่อไปนี้

ก) State Diagram

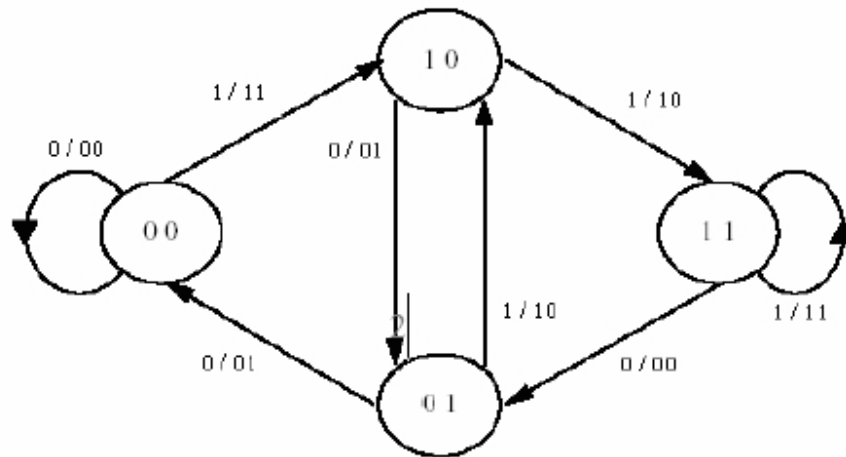
State Diagram จะแสดงค่าของข้อมูลใน shift register และเอาที่พู่ทงของตัวเข้ารหัส ซึ่งแสดงดัง รูปที่ 2.6 ตัวเลขที่อยู่ในวงกลมแต่ละวงนั้น จะหมายถึงสถานะต่างๆ ของข้อมูลที่ถูกรับไว้ใน Shift register ซึ่งในกรณีของวงจรเข้ารหัสที่ใช้เป็นตัวอย่างนั้น จำนวนสถานะ(State) ทั้งหมดเท่ากับ 4 สถานะ และสำหรับลูกศรที่ถูกแสดงไว้ในรูปนั้นจะแสดงถึงลักษณะของการเปลี่ยนแปลงการทำงานจากสถานะหนึ่งไปเป็นอีกสถานะหนึ่ง ซึ่งจะขึ้นอยู่กับข้อมูลที่ป้อนเข้ามา ณ เวลานั้นๆ เช่น กรณีที่ข้อมูลที่เก็บไว้มีค่าเป็น 00 เมื่อมีข้อมูล 1 ป้อนเข้ามาจะมีผลทำให้ข้อมูลที่เก็บไว้ถูกเปลี่ยนไปเป็น 10 และจะมีการส่งค่า 11 ออกไปจากวงจร (แทนด้วยสัญลักษณ์ 1 / 11 ดังรูป) แต่ถ้าหากว่ามีการป้อนข้อมูล 0 เข้ามา วงจรก็จะยังคงมีสถานะเป็น 00 เหมือนเดิมและจะมีการส่งค่า 00 ออกไปจากวงจร (แทนด้วยสัญลักษณ์ 0/00 ดังรูป)



รูปที่ 2.6 State Diagram

ข) Tree Diagram

สำหรับ Tree Diagram นั้น จะเป็นการพิจารณาถึงลักษณะของการทำงานของวงจรเข้ารหัสข้อมูล โดยที่จะมีการพิจารณาถึงค่าของผลลัพธ์ที่ได้หลังจากการป้อนข้อมูลต่างๆ เข้าไปในวงจรเข้ารหัส เป็นหลัก ซึ่งจะมีลักษณะของ Tree Diagram ดังรูปที่ 2.7

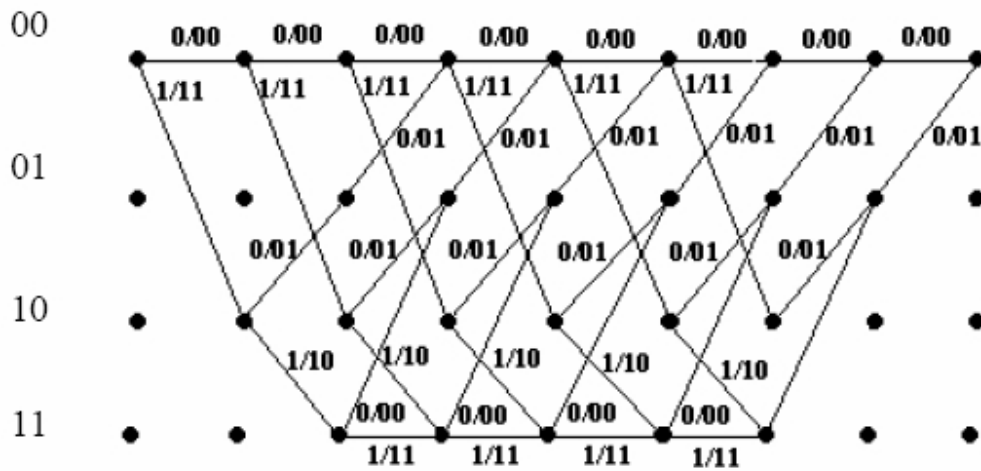


รูปที่ 2.7 Tree Diagram

สำหรับการแสดงการทำงานของวงจรเข้ารหัสแบบ Convolution Codes โดยใช้ Tree Diagram นั้นจะเป็นการพิจารณาการทำงาน โดยการคำนึงถึงข้อมูลที่ป้อนเข้ามาและจะถูกส่งออกไปจากภาคเข้ารหัสเป็นหลัก ซึ่งในการพิจารณานั้น จะเริ่มต้น ณ ตำแหน่งรากของ Tree Diagram ซึ่งในกรณีของรูปที่ 2.8 นั้น จะอยู่ในตำแหน่งซ้ายมือสุดของรูป ซึ่งจะมีการนำข้อมูลที่ถูกป้อนเข้ามาภายในวงจรเข้ารหัส เป็นตัวกำหนดทิศทางการเดินทางของข้อมูลใน Tree Diagram โดยในกรณีของ Tree Diagram ตัวอย่างนั้น จะกำหนดให้มีการเลื่อนตำแหน่งไปทางข้างบนเมื่อมีการรับข้อมูล 0 เข้า มาและจะเลื่อนตำแหน่งลงล่างเมื่อมีการรับข้อมูล 1 เข้ามา ซึ่งหลังจากที่มีการเลื่อนตำแหน่งที่ใช้พิจารณาแล้ว จะมีการพิจารณาถึง ข้อมูลที่จะถูกส่งออกไปจากภาคเข้ารหัส ณ เวลานั้นๆ จากข้อมูล ที่อยู่เหนือเส้นทางในตำแหน่งที่มีการพิจารณา

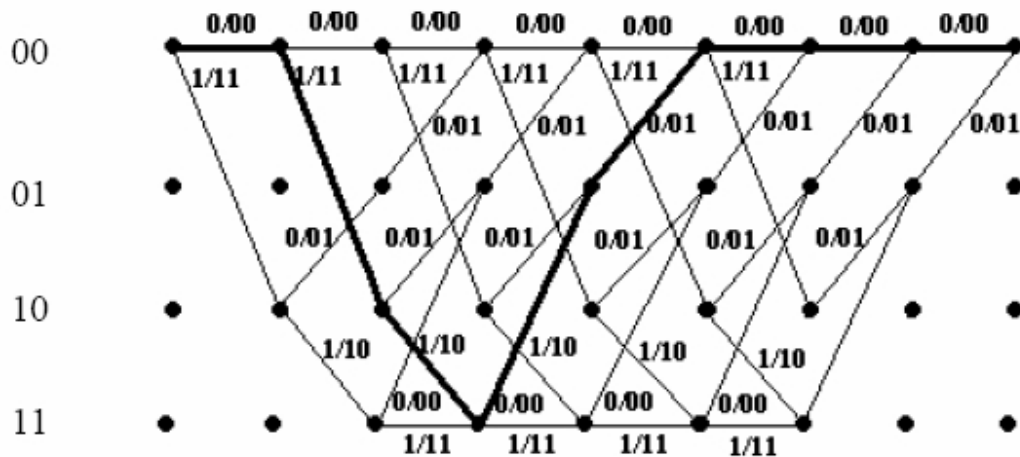
ค) Trellis Diagram

การแสดงการทำงานของวงจรเข้ารหัสโดยใช้ Trellis Diagram นั้น จะเป็นการแสดงการทำงาน ของวงจรเข้ารหัสโดยใช้ State Diagram มาทำการเปลี่ยนแปลงรูปแบบให้อยู่ในอีกลักษณะหนึ่ง ที่แสดงถึงการเปลี่ยนแปลงของข้อมูลต่างๆ ภายในวงจรเข้ารหัส, ข้อมูลที่ป้อนเข้ามาและ Codeword ที่จะถูก ส่งออกไป ณ เวลาต่างๆ โดยที่จะมีลักษณะของ Trellis Diagram ดังรูปต่อไปนี้



รูปที่ 2.8 Trellis Diagram

จากรูปที่ 2.8 จะเป็นการแสดงการทำงานของวงจรเข้ารหัสข้อมูลแบบ Convolution Codes ในรูปที่ 2.5 ที่มีการนำข้อมูลในอดีตจำนวน 2 บิต มาทำการประมวลผลร่วมกับข้อมูล ณ เวลานั้น (จำนวน state ทั้งหมดใน trellis diagram จะมีค่าเท่ากับ $2^2 = 4$ state) และจะมีข้อมูลป้อนเข้ามาภายในวงจรครั้งละ 1 บิต ($k=1$) ซึ่งในกรณีนี้ จะมีข้อมูลที่จะเข้ารหัสทั้งหมด 6 บิต ซึ่งเส้นทางต่างๆ ที่อยู่ใน Trellis Diagram นั้น จะแสดงถึงลักษณะการเปลี่ยนแปลง สถานะของวงจรและตัวเลข x/y ที่อยู่เหนือทางเดิน ในแต่ละเส้นทางนั้น จะแสดงถึงข้อมูลที่ป้อนเข้ามา และ Codeword ที่จะถูกส่งออกไปเมื่อมีการป้อนข้อมูลนั้นเข้ามา ซึ่งเมื่อพิจารณาลักษณะของ Trellis Diagram แล้ว จะพบว่ารูปแบบของ Trellis Diagram ในแต่ละ State การทำงานนั้น จะมีลักษณะที่คล้ายกันแต่จะมีความแตกต่างกันเฉพาะส่วนหัวและท้าย ซึ่งเป็นผลมาจากข้อมูลที่เก็บอยู่ในวงจรมีค่าที่เริ่มต้นจากสถานะที่มีข้อมูลเป็น 0 ทั้งหมด และจะจบลงที่สถานะข้อมูลเป็น 0 เช่นกัน ดังนั้นเส้นทางอื่นๆ ที่ไม่ผ่านจุดที่มีข้อมูลเป็น 0 ทั้งหมด ณ จุดเริ่มต้นและจุดสุดท้ายนั้นจะไม่ถูกนำมาพิจารณา ดังนั้นขนาดของความยาวใน Trellis Diagram นั้น จึงขึ้นอยู่กับข้อมูลที่ถูกลำมาเข้ารหัส ตัวอย่างในการใช้งาน Trellis Diagram เช่น กรณีที่มีการป้อน ข้อมูล 0 1 1 0 0 0 0 เข้ามาภายในวงจร จะสามารถใช้ Trellis Diagram ในการหาลักษณะของ Codeword ได้ดังรูปที่ 2.9



รูปที่ 2.9 การใช้งาน Trellis Diagram

ดังนั้นข้อมูลที่ได้หลังจากการเข้ารหัสจะมีค่าเท่ากับ 00 11 10 00 01 00 00 00

2.2.2) การถอดรหัสแบบ Viterbi Decoder

การถอดรหัส Convolution นั้นมีขั้นตอนที่ซับซ้อนกว่าวิธีการเข้ารหัสมาก ซึ่งการถอดรหัสมีหลายวิธีแต่การถอดรหัสด้วย Viterbi Algorithm นั้นเป็นวิธีที่ให้ประสิทธิภาพสูง ซึ่งลักษณะการทำงานของ Viterbi Algorithm เป็นแบบ Maximum Likelihood Decoding โดยผลลัพธ์ที่ได้จากการถอดรหัสจะเป็นเส้นทางเพียงเส้นทางเดียวที่มีความน่าจะเป็นสูงสุดจากเส้นทางทั้งหมดใน Trellis Diagram ซึ่งมีลักษณะเหมือนกับข้อมูลที่ถูกลบมากที่สุด

รูปแบบสำหรับการถอดรหัสแบบ Viterbi Decoder

สำหรับการถอดรหัสแบบ Viterbi Decoding นั้น จะมีรูปแบบสำหรับการถอดรหัสที่ใช้งานอยู่ 2 ลักษณะด้วยกัน ได้แก่ แบบ Hard-Decision และ Soft-Decision

ก) Hard Decision สำหรับการทำงานของวงจรถอดรหัสที่ใช้กระบวนการตัดสินใจแบบ Hard Decision นั้น จะเป็นการพิจารณาข้อมูลที่รับเข้ามา โดยการพิจารณาว่าข้อมูลที่รับเข้ามาในแต่ละบิตนั้น มีค่าของข้อมูลเป็น 0 หรือ 1 เท่านั้น

ข) Soft Decision การถอดรหัสที่มีการใช้กระบวนการตัดสินใจแบบ Soft Decision จะ เป็นการพิจารณาถึงข้อมูลที่รับเข้ามาได้โดยการทำการตัดสินใจระดับของข้อมูลที่รับเข้ามาได้โดยการแบ่ง ระดับของสัญญาณที่ใช้ในการ คำนวณค่า metric ที่มากกว่า 2 ระดับ ซึ่งผลลัพธ์ที่ได้นั้น จะได้ข้อมูล รายละเอียดของข้อมูลที่ส่งมาที่มากกว่ากรณีของ Hard-Decision ซึ่งข้อมูลที่ได้รับการตัดสินใจ (softoutput) นั้น จะถูกนำมาใช้ในการคำนวณค่า metrics เพื่อ

เปรียบเทียบข้อมูลที่รับ เข้ามา ณ เวลานั้นๆ กับข้อมูลที่อยู่ในเส้นทางต่างๆ ณ เวลานั้น ซึ่งจะมีรูปแบบที่ใช้ในการคำนวณที่แตกต่างกันไป

ก) Viterbi Algorithm โดยในการทำงานต่างๆนั้นจะต้องมีการคำนวณหาความแตกต่าง ระหว่างข้อมูลที่รับเข้ามาและค่าที่อยู่ในเส้นทางต่างๆ เพื่อใช้ในกระบวนการตัดสินใจ โดยกระบวนการ ที่ใช้ในการทำการหาเส้นทางที่ดีที่สุดนั้น จะใช้วิธีการทำงานที่มีชื่อว่า Viterbi Algorithm ซึ่งจะเป็น กระบวนการที่ใช้ในการค้นหาเส้นทางที่อยู่ใน Trellis Diagram ที่มีลักษณะที่ใกล้เคียงกับข้อมูลที่รับ ได้มากที่สุดเพื่อที่จะนำข้อมูลในเส้นทางนั้นมาคำนวณค่าของข้อมูลที่ ถูกส่งมา โดยที่ในกระบวนการค้นหาเส้นทางที่เหมาะสมที่สุดโดยใช้ Viterbi Algorithm นั้น จะมีขั้นตอนในการทำงานดังต่อไปนี้

1) พิจารณาแบ่งข้อมูลที่รับเข้ามาออกเป็นข้อมูลย่อยๆ จำนวน m ช่วง ซึ่งแต่ละช่วงนั้นมีขนาด ของข้อมูลเท่ากับ n_0 บิต

2) ทำการวาด Trellis diagram ที่มีจำนวน state ในการทำงานเท่ากับ m state โดยจะมีการพิจารณาเฉพาะเส้นทางที่มีความเป็นไปได้ว่าจะถูกส่งมาเท่านั้น โดยสำหรับที่ state ของTrellis diagram ตั้งแต่ $L-1$ ขึ้นไปนั้น ให้วาดเฉพาะเส้นทางที่จะพุ่งเข้าหาสถานะของวงจรที่มีข้อมูลเป็น 0 ทั้งหมด

3) กำหนดค่าตัวแปร $1 = 1$ และทำการกำหนดค่าเริ่มต้นของตัวแปร metric ในสถานะเริ่มต้นที่มีข้อมูลเป็น 0 ทั้งหมด ให้มีค่าของ metric เท่ากับ 0

4) ทำการคำนวณหาค่าความแตกต่างของข้อมูล (distance) ระหว่างข้อมูลที่รับ ได้ ชุดที่ 1 กับ ข้อมูลในเส้นทางในการเปลี่ยนแปลงสถานะใน Trellis diagram จาก state ที่ 1 ไปเป็น $1+1$

5) นำค่าที่คำนวณได้นั้นไปบวกกับค่า metric สะสมของstate 1 เพื่อคำนวณหา ค่าของ metric สะสมใน state ที่ $1+1$ เพื่อใช้ในการตัดสินใจเลือกเส้นทางที่เหมาะสมที่สุด ในการเปลี่ยนแปลงข้อมูลไปยัง state นั้นๆ โดยในแต่ละ state นั้น จะมีจำนวนเส้นทางทั้งหมด จำนวน $2k_0$ เส้นทางที่จะพุ่งเข้า state เดียวกัน

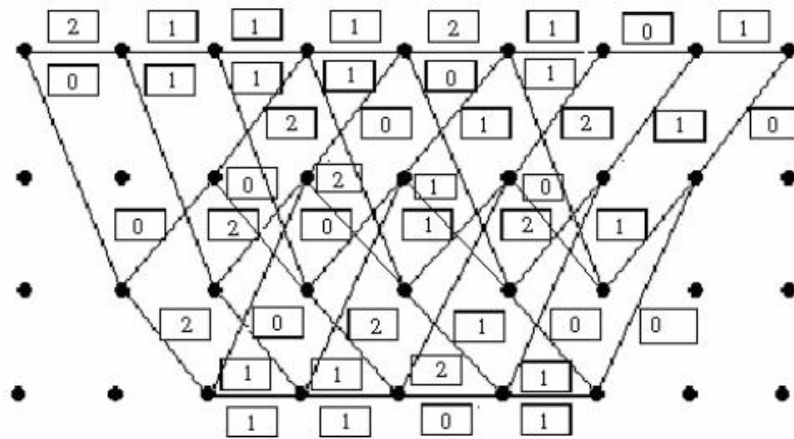
6) พิจารณา ณ ตำแหน่งใน state ที่ $1+1$ ในแต่ละstateนั้น ทำการเลือกเส้นทางที่มี ค่า metric สะสมที่มีค่าน้อยที่สุดที่พุ่งเข้าหาในแต่ละ state โดยที่เส้นทางที่ถูกเลือกนั้น จะ

ถูกเรียกว่า "Survivor" ซึ่งจะเป็นเส้นทางที่ถูกเก็บไว้ทำการคำนวณใน state ต่อไป และสำหรับเส้นทางอื่นๆ ที่ไม่ได้ถูกเลือกนั้น จะถูกเรียกว่า "Forgetting" "โดยจะถูกลบทิ้งออกไปจากระบบการตัดสินใจ

7) ถ้าหากว่า 1 นั้นมีค่าเท่ากับ m แล้วให้ทำงานในขั้นตอนต่อไปได้ แต่ถ้ายังมีค่าน้อยกว่า จะมีต้องมีการเพิ่มค่า 1 ขึ้นอีก 1 จากนั้นจึงกลับไปทำงานที่ขั้นตอนที่ 4 ใหม่

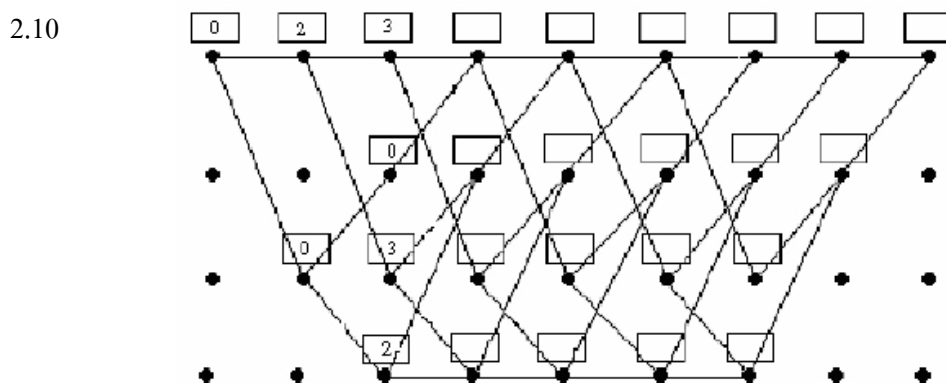
8) เริ่มต้นพิจารณา ณ state ที่ $m+1$ ที่มีสถานะของข้อมูลสถานะเป็น 0 ทั้งหมด ทำการเลือกเส้นทางที่เป็น "Survival" ซึ่งเป็นเส้นทางที่ถูกเลือกที่เหลืออยู่ย้อนกลับไปจนกระทั่งถึงสถานะเริ่มต้น ของการทำงานที่มีสถานะในการทำงานเป็น 0 ทั้งหมด ซึ่งเส้นทางที่ได้นั้น จะเป็นเส้นทางที่มีลักษณะที่ ใกล้เคียงกับข้อมูลที่รับเข้ามามากที่สุด ซึ่งจะถูกนำไปใช้ในการคำนวณหาข้อมูลข่าวสารที่ถูกส่งมา โดยข้อมูลข่าวสารที่จะถูกส่งออกไปจากภาคถอดรหัส นั้น จะเป็นการส่งข้อมูลทั้งหมดที่อยู่ในเส้นทางส่งออกไป ยกเว้นข้อมูล 0 จำนวน $k_0(L-1)$ บิต ที่อยู่ท้ายสุดนั้น จะถูกตัดทิ้งไป

ตัวอย่างเช่น ในกรณีของวงจรถอดรหัสสำหรับข้อมูลที่ถูกเข้ารหัสด้วยวงจรในรูปที่ 2.6 ซึ่งมีกรป้อนข้อมูลดิบจำนวน 6 บิตมีค่าเท่ากับ 101011 ซึ่งเมื่อทำการเข้ารหัสแล้วจะได้ Codeword มีค่าเท่ากับ 1101100110100001 จากนั้นพิจารณาในกรณีที่ข้อมูลของ Codeword ที่รับได้ที่ปลายทางนั้นเกิดความผิดพลาดในการตีความขึ้นจำนวน 1 บิต จึงทำให้ข้อมูลที่รับได้มีค่าเท่ากับ 1101100111100001 ดังนั้นในการถอดรหัสโดยใช้วิธีการ Viterbi Decoding นั้นจะทำงานโดยเริ่มต้นจากการนำข้อมูลที่ได้นั้นมาทำการแบ่งออกเป็นชุดๆ ที่มีขนาดเท่ากับจำนวนของ Codeword ที่วงจรเข้ารหัสส่งออกมาใน แต่ละช่วงเวลาซึ่งในกรณีนี้มีค่าเท่ากับ 2 ดังนั้นข้อมูลต่างๆ จะถูกแบ่งออกเป็นชุดดังนี้ 11011001110001 ซึ่งในการทำงานนั้นจะมีการพิจารณาข้อมูลที่ละชุดเพื่อนำมาเปรียบเทียบกับข้อมูลที่อยู่ในเส้นทางต่างๆ ใน Trellis Diagram โดยการทำงานในช่วงแรกนั้นจะเป็นการหาความแตกต่างระหว่าง ข้อมูลที่รับได้และข้อมูลที่อยู่ในเส้นทางต่างๆ ซึ่งมีลักษณะการคำนวณดังรูปที่ 2.9



รูปที่ 2.10 การถอดรหัสแบบViterbi Decoding (1)

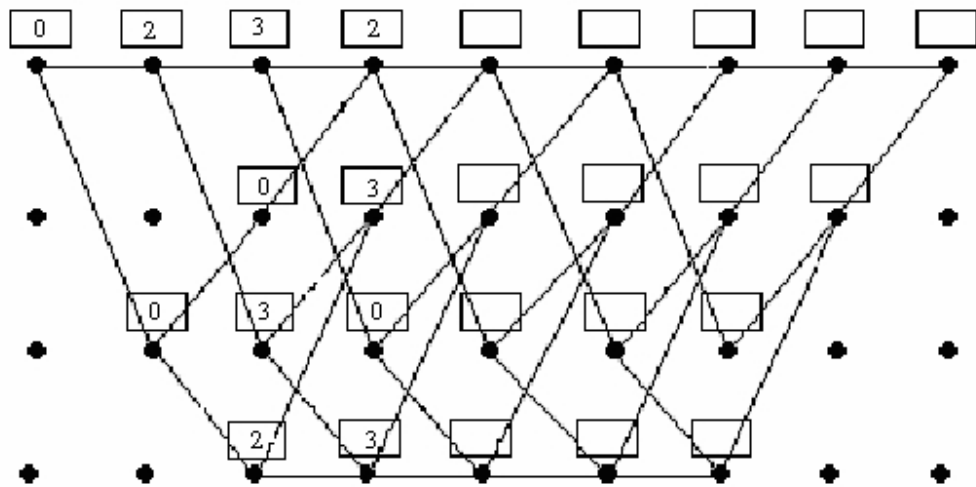
ซึ่งจะเป็นการนำข้อมูลที่รับได้ในแต่ละช่วงเวลาทำการเปรียบเทียบกับข้อมูลที่อยู่ในเส้นทางต่างๆ โดยในช่วงเวลาที่ 1 นั้น จะเป็นการนำข้อมูล 11 ที่รับได้มาทำการเปรียบเทียบเพื่อหาจำนวนบิต ของข้อมูลที่มีความแตกต่างกัน โดยเมื่อเปรียบเทียบกับเส้นทางที่ลากจาก Node เริ่มต้นที่มีข้อมูลเป็น 0 ทั้งหมดมายัง Node ที่มีข้อมูลเป็น 00 ในช่วงเวลาถัดไปซึ่งมีข้อมูลในเส้นทางเป็น 0/00 ซึ่งจะพบว่าจะมี ค่า Codeword ที่แตกต่างกันจำนวน 2 บิต และเมื่อเปรียบเทียบกับเส้นทางที่ลากไปยัง Node ที่มีข้อมูลเป็น 10 จะพบว่ามีความแตกต่างเท่ากับ 0 บิต และใน ช่วงเวลาที่ 2 จะเป็นการนำข้อมูล 01 ที่รับได้ ณ เวลานั้นมาทำการเปรียบเทียบกับเส้นทางต่างๆทั้งหมด ณ เวลานั้นๆ และจะมีการคำนวณเช่นนี้ไปเรื่อยๆจนครบทุกเส้นทาง จากนั้นในการทำงานขั้นตอนต่อไปจะเป็นการคำนวณหาค่า metric สะสมของเส้นทางโดยสำหรับ 2 ช่วงเวลาแรกนั้นจะมีการทำงานดังรูปที่



รูปที่ 2.11 การถอดรหัสแบบ Viterbi Decoding (2)

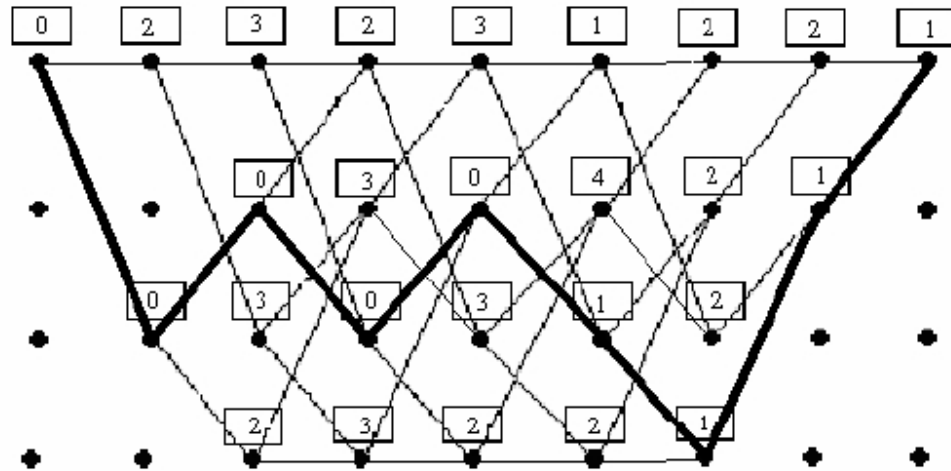
โดยในการทำงานนั้นจะเริ่มตั้งแต่ Node 00 ที่อยู่ซ้ายบนสุดของรูป ซึ่งกำหนดใหม่ค่า metric เป็น 0 จากนั้นจึงพิจารณา Node ถัดไปที่เชื่อมต่อกับ Node 00 ซึ่งมี 2 Node ด้วยกัน โดยจะทำการนำ metric ที่อยู่ในเส้นทางนั้นมาบวกกับ metric สะสมที่ถูกเก็บไว้ยัง Node ที่พิจารณา ซึ่ง ณ

เวลานี้คือ Node 00 ที่อยู่ซ้ายบนสุด และผลลัพธ์ที่ได้นั้นจะถูกเก็บไว้ใน Node ที่อยู่ปลายทางของเส้นทางนั้นๆ ซึ่ง จะมีการทำงานเช่นนี้ไปเรื่อยๆจนกระทั่งมีการพบว่า มีเส้นทางมากกว่า 1 เส้นทางที่พุ่งเข้าหา Node เดียวกันดังเช่นในช่วงเวลาที่ 3 จนถึงช่วงเวลาที่ 8 ซึ่งเมื่อเกิดเหตุการณ์นี้ขึ้นจะต้องมีกระบวนการ ตัดสินใจเพื่อเลือกเส้นทางที่ใช้ในการคำนวณเฉพาะ เส้นทางที่มีค่า metric สะสมน้อยที่สุดเท่านั้นมาพิจารณา โดยในช่วงเวลาที่ 3 นั้น จะมีการคำนวณดังรูปที่ 2.11



รูปที่ 2.12 การถอดรหัสแบบ Viterbi Decoding (3)

โดยสำหรับการพิจารณาค่า metric สะสมที่ Node 00 ในช่วงเวลาที่ 3 นั้น จะเป็นการตัดสินใจ เลือกเส้นทางระหว่างเส้นทางที่มาจาก Node 00 และ Node 01 ของช่วงเวลาที่ 2 ซึ่งจะทำให้การเลือกเส้นทางที่มีค่า metric สะสมที่น้อยที่สุดเท่านั้น โดยสำหรับเส้นทางที่มาจาก Node 00 จะมีค่า metric สะสมเท่ากับ $3+1$ เท่ากับ 4 และเส้นทางที่มาจาก Node 01 มีค่าเท่ากับ $0+2$ เท่ากับ 2 ดังนั้นจึงเลือก เส้นทางที่มาจาก Node 01 มาใช้ในการคำนวณ ซึ่งจะมีการทำงานในลักษณะนี้เรื่อยๆจนกระทั่งคำนวณ metric สะสมครบทุกช่วงเวลา จากนั้นจึงนำข้อมูลที่ได้นั้นมาใช้ในการค้นหาเส้นทางโดยการเริ่มต้นที่ Node 00 ที่อยู่ขวาสุดของรูป จากนั้นจึงทำการมองย้อนกลับใน Node ที่อยู่ซ้ายมือที่มีเส้นทางต่อกับ Node 00 โดยจะทำการเลือกเส้นทางที่ต่อกับ Node ที่มีค่า metric สะสมที่น้อยที่สุด เพื่อเลือกเป็นเส้นทางที่ใช้งาน จากนั้นจึงย้ายจุดที่พิจารณาไปยัง Node ถัดไปทางซ้ายมือที่อยู่ปลายทางของเส้นทางที่เลือกไว้ ซึ่งจะมีกระบวนการตัดสินใจเช่นเดิมซ้ำอีกครั้ง โดยจะมีการทำงานเช่นนี้ไปเรื่อยๆ จนมีการค้นหาเส้นทางครบทุกช่วงเวลา ซึ่งผลลัพธ์ที่ได้จากการทำงานจะมีลักษณะดังรูปที่ 2.12



รูปที่ 2.13 การถอดรหัสแบบ Viterbi Decoding (4)

หลังจากนั้น จะเป็นการนำข้อมูลที่อยู่ในเส้นทางที่ถูกเลือกมาใช้ในการหาข้อมูลดิบที่ส่งมา โดยการพิจารณาถึงข้อมูลที่อยู่ในเส้นทางส่วนต่างๆ ซึ่งจากรูปนั้นจะเห็นว่าข้อมูลดิบที่ถอดรหัสได้ จะมีค่าเท่ากับ 1 0 1 0 1 1 0 0 โดยใน 2 บิตสุดท้ายจะไม่ถูกนำมาพิจารณาเนื่องจากเป็นข้อมูลที่ถูกป้อนเข้ามาเพื่อใช้ในการคำนวณ ดังนั้นข้อมูลดิบที่ได้จากการถอดรหัสจะมีค่าเท่ากับ 1 0 1 0 1 1

3) Interleaving

เป็นกระบวนการในการสลับย้ายตำแหน่งของส่วนต่าง ๆ ของข้อมูลเพื่อลดโอกาสที่จะเกิดความผิดพลาดในการรับส่งข้อมูลโดยไม่ต้องสูญเสียแบนด์วิดท์หรือขีดความสามารถในการรับส่งข้อมูลแต่อย่างใด สำหรับไวแมกซ์หรือ 802.16e นั้นก็ได้ใช้วิธีนี้หลังการทำ Channel Coding ด้วยเพื่อประสิทธิภาพที่เพิ่มขึ้นดังกล่าว วิธีการใน 802.16e จะมีด้วยกัน 2 ขั้นตอนดังต่อไปนี้

3.1) ทำการสลับหรือย้าย ตำแหน่งบิตข้อมูลที่ติดกันไปไว้กับความถี่ subcarrier ที่ไม่ติดกันของ OFDMA ที่ใช้งานอยู่เพื่อให้โอกาสที่จะถูกรบกวนจากความถี่รบกวนลดลง เป็นการสร้าง Frequency Diversity ที่ดี เพิ่มประสิทธิภาพในการถอดรหัสให้มากขึ้น

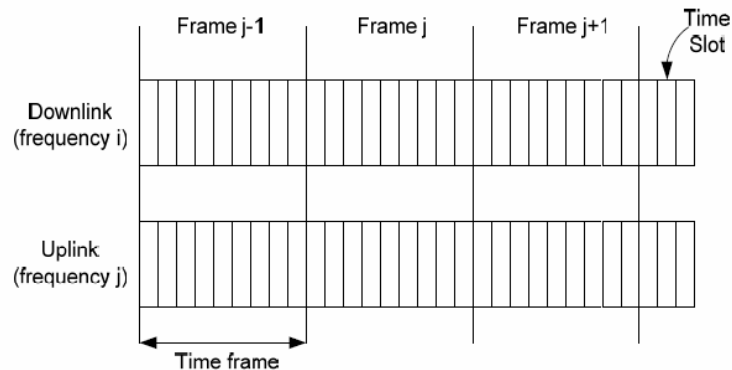
3.2) จากนั้นทำการย้าย ตำแหน่งบิตข้อมูลที่ติดกันให้ไปอยู่บนบิตที่ significant ต่างกันในการมอดูเลชันเพื่อให้กระจายความเสี่ยงที่จะเกิดจากการรบกวนสัญญาณ ที่ผ่านการมอดูเลต เพราะโอกาสที่จะเกิดความผิดพลาดในแต่ละบิตของ 16QAM และ 54QAM จะแตกต่างกันออกไป บิตที่เป็น Most significant จะมีโอกาสเกิดความผิดพลาดได้น้อยกว่าบิตที่เป็น Least significant นั่นเอง

4) Symbol Mapping

กระบวนการ การนี้เป็นกระบวนการที่เชื่อมโยงข้อมูลเข้ากับการมอดูเลชัน โดยจะจับกลุ่มของไบนารีบิตของข้อมูลเพื่อแสดงเป็น symbol ที่เป็นตัวแทนการมอดูเลชันค่าใน QPSK, 16QAM และ 64QAM

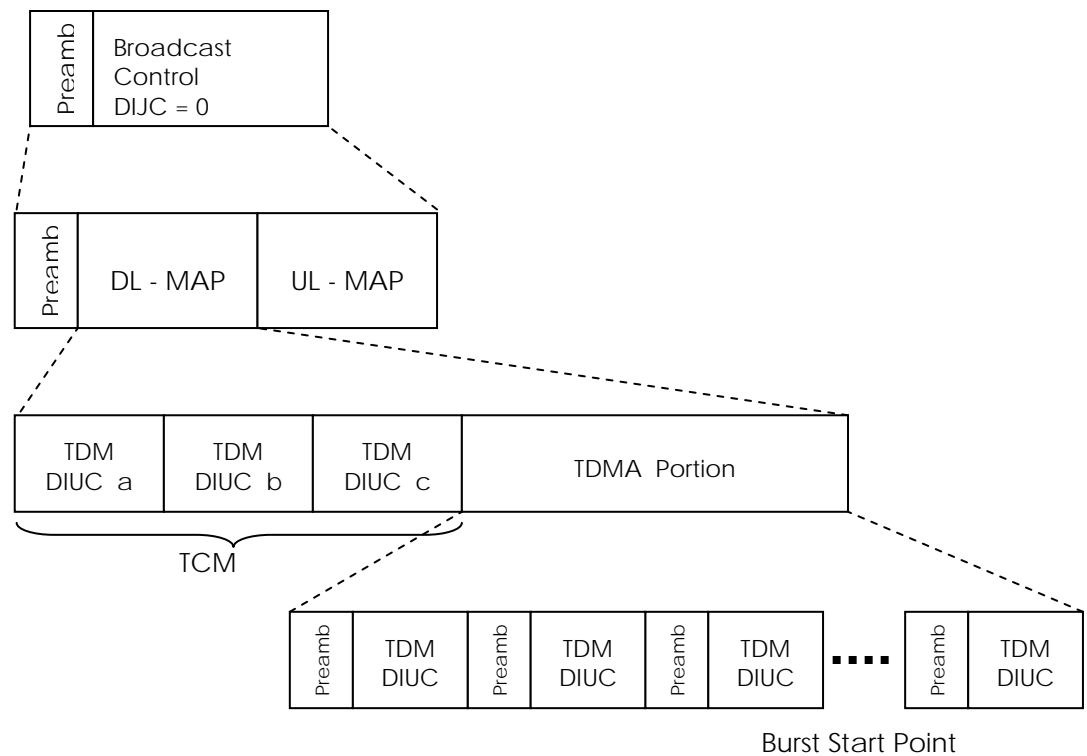
2.3 FDD (Frequency Division Duplex)

FDD (Frequency Division Duplex) การส่งข้อมูล uplink กับ downlink ส่งที่คนละความถี่ การส่งข้อมูล uplink กับ downlink จึงสามารถส่งได้ในเวลาเดียวกัน นอกจากนี้ FDD ยังรองรับ CPE ที่เป็นแบบ Half-FDD คือ อุปกรณ์ลูกข่ายจะทำการส่งข้อมูลและรับข้อมูลคนละเวลากัน คือขณะที่อุปกรณ์ลูกข่ายกำลังส่งข้อมูล จะไม่สามารถรับข้อมูลได้ แต่การส่งและรับข้อมูลยังใช้คนละความถี่ การใช้รูปแบบ Half-FDD อุปกรณ์จะมีราคาที่ถูกกว่า full FDD เพราะส่วนประกอบภาครับสัญญาณกับภาคส่งสัญญาณของอุปกรณ์สามารถใช้ชุดเดียวกันรวมกันได้



รูปที่ 2.14 แสดงการส่งข้อมูล ขาขึ้น (Uplink) และ ขาลง (Downlink) โดยใช้ FDD

โดยในเทคโนโลยี WiMAX นั้นเราจะสามารถจำแนกในส่วนการจัด Frame ของข้อมูลขาขึ้น (Uplink) กับขาลง (Downlink) ได้ดังนี้



รูปที่ 2.15 แสดงเฟรมย่อยของ FDD ในข้อมูลขาลง (Down link)

ในข้อมูลขาขึ้น (Up link) และข้อมูลขาลง (Down link) นั้นจะมีลักษณะการจัดเฟรมที่เหมือนกัน ซึ่งจะแตกต่างกันเพียงความถี่ที่ใช้งานเท่านั้น

2.4 การมอดูเลต (Modulation)

2.4.1 เทคนิคการมอดูเลต Orthogonal Frequency Division Multiplex (OFDM)

OFDM ย่อมาจาก Orthogonal Frequency Division Multiplex เป็นรูปแบบของ MCM สมัยใหม่ที่มีการเพิ่มความหนาแน่นของสัญญาณพาหะย่อยและการซ้อนทับสัญญาณพาหะที่อยู่ติดกัน ซึ่งเป็นเทคนิคการมอดูเลชันแบบหลายคลื่นพาหะ (Multiple Carrier Modulation) และเป็นรูปแบบของการสื่อสารแบบขนาน แตกต่างจากมาตรฐาน Spread Spectrum ทั่วไปที่ใช้แถบความถี่เดียวสำหรับรับส่งข้อมูลช่องเดียว

OFDM เป็นเทคโนโลยีการเข้ารหัสในชั้น Physical Layer สำหรับถ่ายทอด สัญญาณผ่านทางคลื่นวิทยุ ซึ่งกระบวนการนี้จะเป็นการแบ่งสัญญาณ ความถี่ออกเป็นหลาย ๆ เซ็กเมนต์ และในแต่ละเซ็กเมนต์จะมีความถี่ เฉพาะตัวด้วยรหัสที่ต่างกัน ซึ่งจะทำให้สามารถรองรับความจุได้

มากกว่า คลื่นวิทยุในคลื่นเดียวกัน ซึ่งโดยทั่วไปแล้ว OFDM ถูกใช้อยู่ในคลื่น ความถี่ 2.4 GHz ISM ซึ่งเป็นคลื่นความถี่มาตรฐานสำหรับมาตรฐาน เครือข่าย IEEE 802.11g ในขณะที่ IEEE 802.11a ถูกเลือกให้คลื่น ความถี่ที่ 5 GHz UNII (Unlicensed National Information Infrastructure)

OFDM สร้างขึ้นมาเพื่อใช้งานสำหรับระบบสื่อสารไร้สายแบบเคลื่อนที่แบนด์กว้าง (Broad band) มีอัตราการส่งข้อมูลสูงๆ เช่น ระบบ LAN แบบไร้สาย (Wireless LAN) ระบบอินเทอร์เน็ตความเร็วสูงแบบไร้สาย (Wireless high speed internet) ระบบกระจายเสียงแบบดิจิทัล (Digital Audio Broadcasting) หรือ DAB และระบบกระจายสัญญาณ โทรทัศน์แบบดิจิทัล (Digital Television) หรือ DTV เป็นต้น

2.4.2 ความตั้งฉากซึ่งกันและกัน ใน OFDM

ความตั้งฉากซึ่งกันและกัน (Orthogonality) คือ “การเกี่ยวข้งกันเป็นมุมฉากหรือมีส่วนประกอบเป็นมุมฉาก” ความตั้งฉากซึ่งกันและกันมีความสัมพันธ์กันของเวกเตอร์ 2 เส้นในช่องว่างอาจจะตัดกันเป็นมุมฉากเรขาคณิตก็ได้ ดังนั้นสัญญาณที่ตั้งฉากซึ่งกันและกันหมายความว่าสถานะของสัญญาณเป็นอิสระจากกัน ตัวอย่างเช่น การมอดูเลตสัญญาณทางความถี่ (FM) และทางขนาด (AM) บนสัญญาณพาหะเดียวกัน (ในระบบ FM สเตอริโอ) สามารถแยกสัญญาณทั้งสองออกจากกัน โดยปราศจากการรบกวนซึ่งกันและกัน ซึ่งเราสามารถเรียกสัญญาณทั้งสองนี้ว่า “ความตั้งฉากซึ่งกันและกัน”

สมมุติว่าในระบบสื่อสารใช้การมอดูเลตแบบเลื่อนความถี่ (FSK) โดยสัญญาณไบนารีกำหนดในบิต “0” แทนที่ด้วยความถี่หนึ่งที่เลื่อนออกไป สัญญาณการมอดูเลตนี้สามารถแยกออกจากกันด้วยวงจร

กรองความถี่ 2 ชุด ที่สามารถปรับเลือกความถี่ทั้งสองได้ วงจรกรองความถี่สำหรับบิต “1” จะไม่ตอบสนองกับสัญญาณบิต “0” และในทำนองเดียวกันในกรณีของวงจรกรองความถี่สำหรับบิต “0” เราเรียกวงจรมอดูเลตแบบเลื่อนความถี่ว่า การจัดการสัญญาณไบนารีให้ตั้งฉากซึ่งกันและกัน

ใน OFDM ความตั้งฉากซึ่งกันและกันเป็นการนำเอาสัญญาณพัลส์มาทำเป็นสัญญาณพาหะซึ่งทำให้สเปกตรัมของสัญญาณพาหะแต่ละตัวมีค่าสัญญาณเป็นศูนย์ที่กลางแถบความถี่ของแต่ละ

สัญญาณพาหะในระบบ ทั้งนี้ทำให้แต่ละสัญญาณพาหะย่อยสามารถแยกออกจากกลุ่มโดยปราศจากการสอดแทรกหรือรบกวนจากสัญญาณพาหะย่อยอื่นๆ ดังนั้นสัญญาณพาหะย่อยสามารถจัดวางไว้ใกล้มากที่สุดเท่าที่จะทำได้ในทางทฤษฎี

2.4.3 การแปลงฟูรีเยร์และการแปลงกลับฟูรีเยร์ ของ OFDM

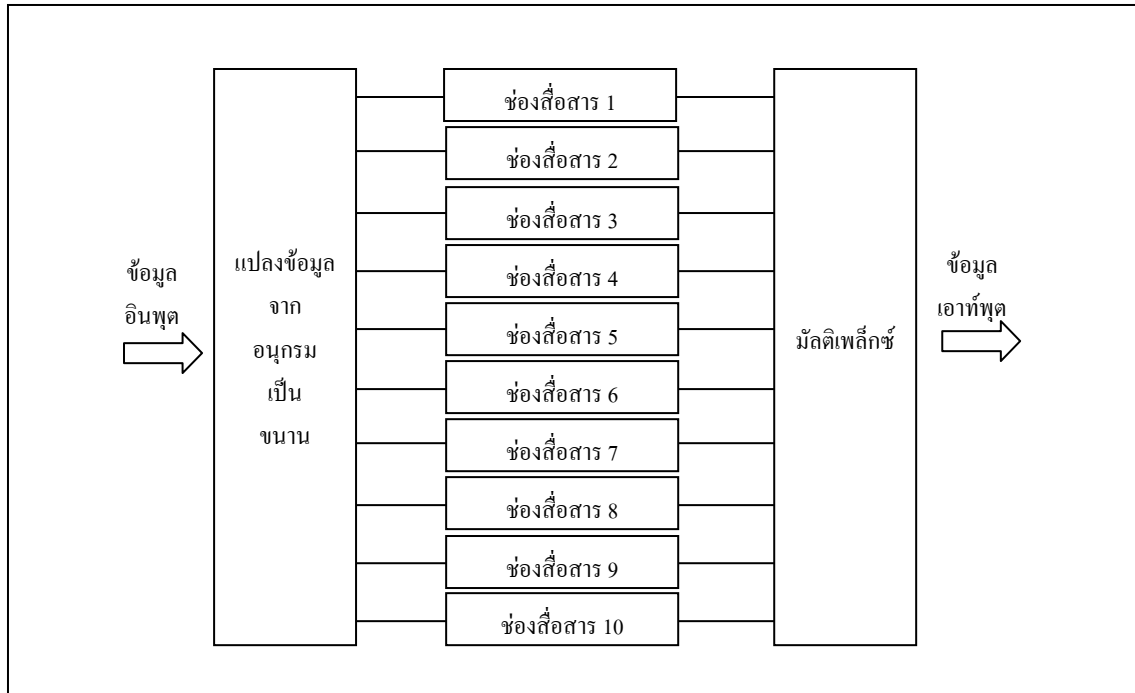
ฟังก์ชันทางคณิตศาสตร์ของการแปลงฟูรีเยร์เปรียบได้กับแท่งปริซึมที่กระจายแสงสีขาออกเป็นแถบสีที่เป็นส่วนประกอบของแสงสีขาว การแปลงฟูรีเยร์นั้นใช้สำหรับแยกสัญญาณออกเป็นส่วนประกอบทางความถี่ของสัญญาณนั้นๆ ซึ่งที่จริงแล้วการแปลงฟูรีเยร์สามารถใช้การแก้ปัญหาที่หลากหลายโดยการวิเคราะห์หองค์ประกอบทางความถี่ของสัญญาณในระบบนั้นๆ การจัดการกับสัญญาณดิจิทัลในระบบ OFDM เป็นวิธีการหนึ่งในการประยุกต์ใช้งานของการแปลงฟูรีเยร์

การแปลงฟูรีเยร์เป็นเทคนิคทางคณิตศาสตร์สำหรับแปลงฟังก์ชันที่มีแกนในเชิงเวลาเป็นฟังก์ชันที่มีแกนในเชิงความถี่โดยอยู่บนพื้นฐานที่ฟังก์ชันนั้นๆ จะต้องเป็นฟังก์ชัรยคาบด้วย ซึ่งจะแตกออกเป็นสมการแปลงฟูรีเยร์

การแปลงฟูรีเยร์เป็นการแยกหรือสลายรูปคลื่นหรือฟังก์ชันออกเป็นสัญญาณลักษณะซายซ์ที่มีความถี่ต่างกันและขนาดแรงดันต่างกัน ซึ่งผลรวมของสัญญาณซายซ์เหล่านี้เมื่อนำมารวมกันทางเฟสแล้วจะได้รูปคลื่นดั้งเดิมที่นำมาแยก ดังนั้นสัญญาณที่มีการเปลี่ยนแปลงตามเวลาอาจจะเปลี่ยนเป็นนิพจน์ของผลรวมรูปคลื่นซายซ์และโคซายซ์ได้ ตัวอย่างเช่น ในการส่งสัญญาณดิจิทัลกับการส่งอนุกรมของสัญญาณซายซ์จะมีผลลัพธ์เหมือนกัน

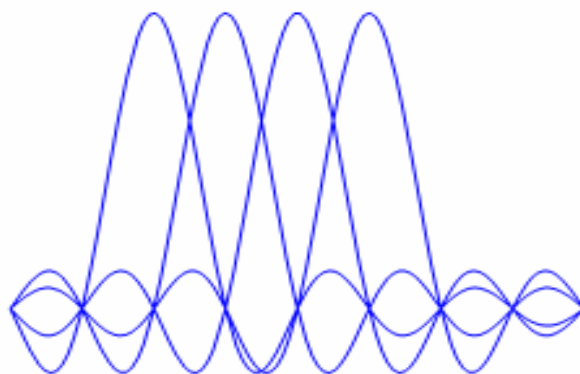
นั่นคือ การแปลงฟูรีเยร์เป็นการแปลงสัญญาณจากแกนเวลาเป็นแกนความถี่โดยจะจัดเป็นนิพจน์ของอนุกรมสัญญาณความถี่เป็นองค์ประกอบ ส่วนการแปลงกลับฟูรีเยร์เป็นกระบวนการที่ตรงข้ามกันคือแปลงสัญญาณที่อยู่ในแกนความถี่ให้อยู่ในรูปแบบแกนเวลา ส่วนการแปลงฟาสฟูรีเยร์ (FFT) จะทำให้การคำนวณการแปลงฟูรีเยร์และการแปลงกลับฟูรีเยร์ของกลุ่มข้อมูลทั้งค่าจริง (Real) และค่าเชิงซ้อน (complex) สามารถทำได้อย่างรวดเร็ว

สัญญาณซายซ์ที่มีช่วงเวลาที่มืค่าจำกัด (T) สามารถแปลงเป็นสัญญาณสมมูลซิง (Sinc Pulse: โดยสัญญาณซิงคือสัญญาณ $\text{sinc}(x)$) บนแกนความถี่ดังแสดงในรูป



รูปที่ 2.16 แสดงการแปลงฟูเรียร์อย่างรวดเร็วของสัญญาณที่ทำการมอดูเลตกับสัญญาณพาหะย่อย (หนังสือ OFDM เทคโนโลยีการมัลติเพล็กซ์ในระบบสื่อสารยุคใหม่)

ซึ่งสัญญาณซึ่งจะมีจุดตัดแกน x ทุกๆ ช่วงเวลา $1/T$ เมื่อ T เป็นช่วงเวลาของจำนวนสัญญาณบนแกนเวลา ดังนั้นถ้าสัญญาณพาหะย่อยมีช่วงห่างซึ่งกันและกันเท่ากับ $1/T$ ค่าสูงสุดจะเกิดขึ้นทุกๆ จุดตัดแกน x ทั้งนี้เนื่องจากการทับซ้อนกันของสัญญาณเหล่านี้จะไม่รบกวนซึ่งกันและกันหรือในอีกทำนองหนึ่งอาจกล่าวได้ว่าสัญญาณมีการรักษาความตั้งฉากซึ่งกันและกัน แนวคิดดังกล่าวแสดงไว้ในดังรูป



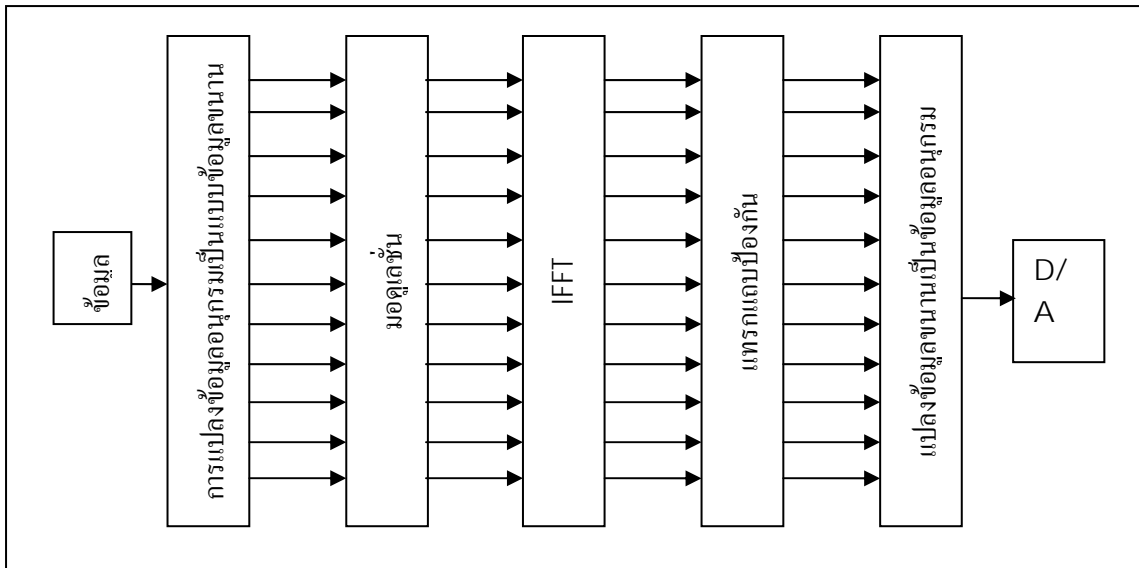
รูปที่ 2.17 แสดงการมัลติเพล็กซ์อนุกรมของสัญญาณบนแกนความถี่

2.4.4 หลักการทำงานของระบบ OFDM

โดยพื้นฐานแล้ว OFDM เป็นเทคนิคการมัลติเพล็กซ์โดยการแบ่งความถี่ เมื่อช่องความถี่ถูกแบ่งออกเป็นความถี่ขนาดเล็กลงๆ N ช่องแต่ละช่องมีขนาดเท่ากับขนาดของสัญลักษณ์ (bit rate) ดิจิตอล ซึ่งทำให้สัญญาณพาหะย่อยเหล่านี้มีความถี่ต่างกัน

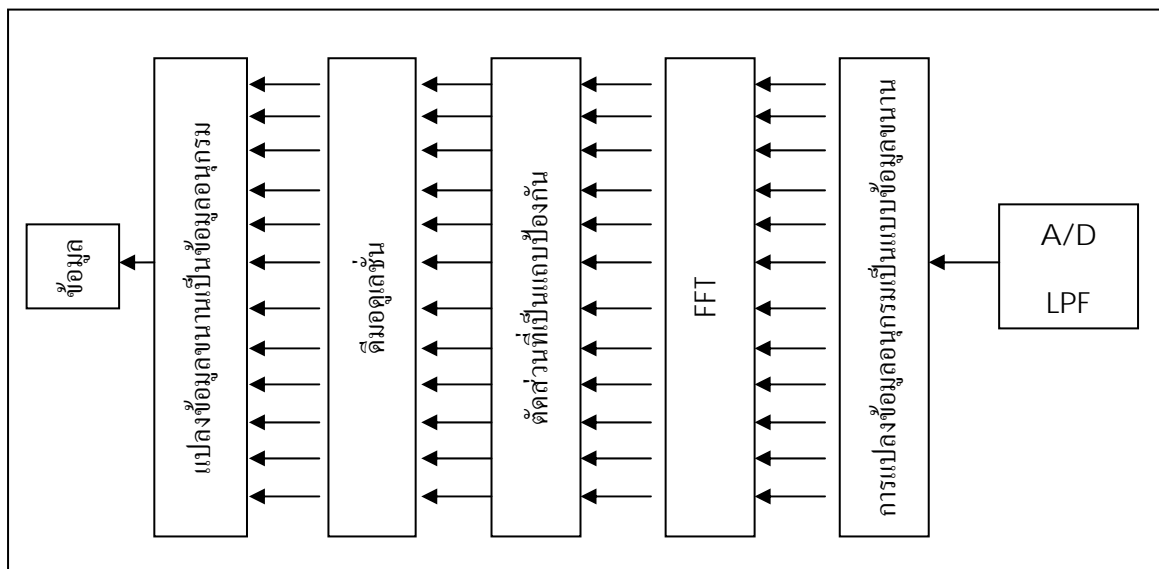
ทางด้านส่งสัญญาณดิจิตอลความเร็วสูงที่ถูกแบ่งออกเป็นกลุ่มข้อมูลที่ความเร็วต่ำลงจำนวนสิบหรือร้อยกลุ่ม ซึ่งแต่ละกลุ่มข้อมูลย่อยที่มีความเร็วต่ำกว่า จะถูกนำไปมอดูเลตกับสัญญาณพาหะย่อยทั้งหมดส่งขนาดกันออกไป รูปแบบในการมอดูเลตสัญญาณพาหะย่อยที่นิยมทั่วไปได้แก่ QAM, 16QAM หรือ 64QAM เป็นต้น ใน OFDM กลุ่มของข้อมูลจะถูกแปลงให้อยู่ในรูปแบบนูนกัน โดยการมอดูเลตกับสัญญาณพาหะย่อย ดังนั้นพวกมันจะกลายเป็นสัญญาณบนแกนความถี่ ในการที่จะแปลงกลับมาเป็นสัญลักษณ์บนแกนเวลาอีกครั้งนั้นทำได้โดย การใช้การแปลงกลับฟาส์ฟูเรียร์ (FFT) โดยที่สัญลักษณ์ทางแกนเวลาเหล่านี้จะถูกมัลติเพล็กซ์เข้าด้วยกันเป็นอนุกรมของสัญญาณ และสัญญาณที่ได้ทางเอาต์พุตของ IFFT จะถูกส่งบนความถี่ที่เหมาะสมในระบบสื่อสาร

ในระบบ OFDM แต่ละสัญญาณพาหะย่อยจะถูกทำให้รูปร่างมีความถี่ต่างกันเล็กน้อยกับสัญญาณพาหะย่อยอื่นๆ ด้วยวิธีการจัดการสัญญาณในลักษณะนี้ทำให้แม้แต่สัญญาณพาหะย่อยที่อยู่ใกล้กันซ้อนทับกันก็ไม่ก่อให้เกิดการรบกวนซึ่งกันและกัน ดังนั้นในระบบนี้จึงไม่ต้องการแถบป้องกัน (Guard Band) ระหว่างแต่ละช่องสื่อสารย่อย ทำให้ช่องสื่อสารระหว่างแต่ละสัญญาณพาหะย่อยมีความแคบที่สุดเท่าที่จะทำได้ ซึ่งในระบบนี้จะให้ประสิทธิภาพในการใช้แถบความถี่สูงสุด ความสัมพันธ์ระหว่างทุกสัญญาณพาหะย่อยในระบบ OFDM จะต้องถูกควบคุมอย่างระมัดระวังเพื่อรักษาให้มีลักษณะของความถี่ต่างกันและกันการแปลงกลับฟาส์ฟูเรียร์เป็นการแปลงที่มีประสิทธิภาพสูงมากและเป็นวิธีการขั้นต้นอย่างง่าย ๆ ที่สามารถทำให้เรามั่นใจได้ว่าสัญญาณพาหะที่สร้างขึ้นมามีความถี่ต่างกันและกัน หลังจากการมอดูเลตแบบ OFDM จะมีการสอดแทรกช่วงแถบป้องกันแคบๆ เพื่อลดสัญญาณรบกวนระหว่างสัญลักษณ์ (Inter symbol Interference : ISI) ที่เกิดจากสัญญาณหลายเส้นทาง (multi-path) เราเรียกแถบป้องกันแคบๆ นี้ว่า การเสริมไซคลิก (Cyclic prefix)



รูปที่ 2.18 แสดงบล็อกไดอะแกรมตัวส่ง OFDM

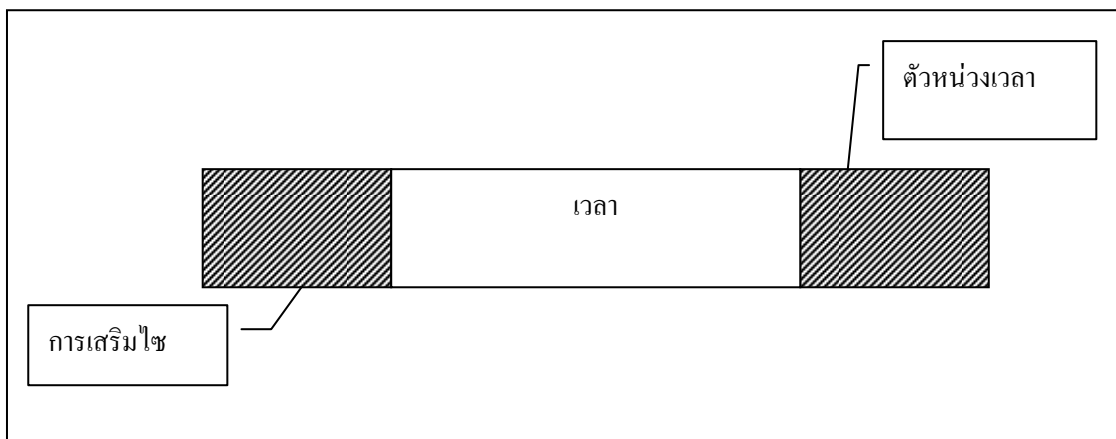
เครื่องรับจะดำเนินการกระบวนการตรงกันข้ามกับเครื่องส่ง ในเครื่องส่ง ในเครื่องรับจะใช้การแปลงฟาสต์ฟูเรียร์แปลงสัญญาณที่อยู่บนแกนเวลาไปเป็นแถบความถี่สมมูลย์ ดังรูปการแสดงบล็อกไดอะแกรม



รูปที่ 2.19 แสดงบล็อกไดอะแกรมรับ OFDM

1) การเสริมสร้างไซคลิก

เทคนิคนี้ใช้สำหรับแก้ไขสัญญาณที่คุณภาพลดลงในช่องสื่อสารอันเนื่องมาจากสัญญาณมีการเคลื่อนที่หลายเส้นทาง ทำได้โดยการใช้ลำดับของแถบป้องกันที่เราเรียกว่า การเสริมไซคลิก การเสริมไซคลิกเป็นการคัดลอกส่วนท้ายสุดของสัญลักษณ์ OFDM ให้มีความยาวเท่ากับหรือมากกว่าค่าหน่วยเวลาสูงสุดของการกระจายช่วงเวลาอันเนื่องมาจากสัญญาณหลายเส้นทาง การเสริมไซคลิกจะถูกคัดลอกลงในส่วนหน้าของสัญลักษณ์ OFDM เพื่อทำหน้าที่เป็นแถบป้องกันการทับซ้อนระหว่างสัญลักษณ์ที่จะเกิดขึ้นระหว่างการทำการส่งดังแสดงในรูป การแปลงจากข้อมูลแบบขนานไปเป็นแบบอนุกรมจะเกิดขึ้นหลังจากคัดลอกไซคลิกลงในตัวส่ง



รูปที่ 2.20 แสดงการเสริมสร้างไซคลิก

ที่เครื่องส่งส่วนที่เรียกว่าแถบป้องกันจะถูกจำกัดออกไป ซึ่งทราบได้ก็ตามที่ช่วงห่างของแถบป้องกันมีค่ามากที่สุดของการหน่วงเวลาในช่องสื่อสาร การทับซ้อนของสัญญาณที่รับมาจากหลายเส้นทางของสัญลักษณ์ก่อนหน้านี้จะถูกจำกัดออกไปด้วยซึ่งจะทำให้สัญญาณที่แท้จริงที่ต้องการรับยังคงมีความตั้งฉากซึ่งกันและกัน ซึ่งแน่นอนที่กระบวนการนี้ทำให้ช่วงเวลาที่ทำการเสริมไซคลิกไม่สามารถส่งข้อมูลข่าวสารได้ทำให้เกิดการสูญเสียช่วงเวลานี้ไป นอกจากนี้การเสริมไซคลิกยังทำให้เกิดข้อเสียในเรื่องกำลังงานและแถบความถี่ในการส่งแต่ก็เป็นข้อตกลงเพื่อปรับเข้าหากันระหว่างสมรรถภาพและประสิทธิภาพ

2) เวกเตอร์ใน OFDM

สายอากาศแบบไดเวอร์ซิตี (diversity) ถูกนำมาทดสอบสำหรับแก้ปัญหาอันเนื่องมาจากผลกระทบจากสัญญาณหลายเส้นทาง ซึ่งในระบบ OFDM จะใช้สเปซเชียล (spatial) ไดเวอร์ซิตีซึ่งเรียกว่า การมัลติเพล็กซ์โดยการแบ่งความถี่ที่สัญญาณพาหะตั้งฉากซึ่งกันและกันและกันทางเวกเตอร์ (Vector OFDM : VOFDM) VOFDM ใช้สายอากาศ สเปซเชียลไดเวอร์ซิตีเพื่อนำประโยชน์ของการรับสัญญาณวิทยุหลายทิศทางที่มีเฟสและขนาดต่างกันมาเป็นข้อดี โดยการรวมสัญญาณเหล่านี้เข้าด้วยกันเพื่อเพิ่มความน่าเชื่อถือในการรับสัญญาณแบบไม่เป็นเส้นตรง (non-line-of-sight) ที่เครื่องรับจะมีสายอากาศ 2 ชุด ทำหน้าที่รับสัญญาณจากเส้นทางที่ต่างกันและทำการรวมเข้าด้วยกันเพื่อให้ได้อัตราส่วนของสัญญาณต่อสัญญาณรบกวนสูงที่สุดเท่าที่จะเป็นไปได้

ระบบการทำงานของบริษัทซิสโก้ (Cisco) ได้ออกแบบให้ VOFDM ทำการทดสอบและรวมแต่ละสัญญาณที่รับเข้ามาจากสายอากาศเข้าด้วยกัน จากนั้นวิเคราะห์สัญญาณพาหะย่อยทีละสัญญาณและใช้ข้อมูลที่ได้มาทำการสร้างกลับเป็นแต่ละสัญญาณพาหะย่อยที่มีอัตราส่วนของสัญญาณรบกวนต่อสัญญาณสอดแทรกที่สูงขึ้น ผลที่ตามมาคืออัตราการผิดพลาดของข้อมูลลดลง

3) การปรับแต่งการมอดูเลตใน OFDM

เพื่อให้การทำงานง่ายขึ้นต่อระบบ MCM ส่วนใหญ่จะใช้เทคนิคการมอดูเลตแบบคงที่สำหรับทุกสัญญาณพาหะย่อย ในระบบที่ใช้เทคนิคนี้จะต้องเลือกประเภทของดิจิตอลมอดูเลต (digital modulation)

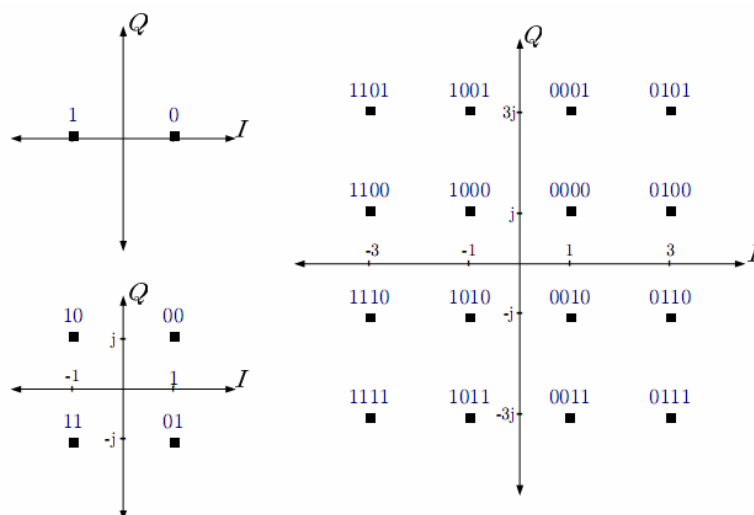
สำหรับสัญญาณพาหะย่อยที่ได้ยอมรับอัตราการผิดพลาดของข้อมูล (Bit error rate) ได้ภายใต้สถานะของช่องสื่อสารที่แย่ที่สุดโดยทั่วไปจะเลือกใช้ BPSK หรือ QPSK ที่ 1 หรือ 2 บิต/วินาที/เฮิรตซ์ตามลำดับ

ในการส่งข้อมูลจากแหล่งข้อมูลหนึ่งไปยังหลายๆ ปลายทางของระบบ OFDM อาจทำได้โดยการเปลี่ยนแปลงเทคนิคทางมอดูเลต ทั้งนี้ขึ้นอยู่กับคุณภาพของสัญญาณสามารถรับได้ที่ปลายทาง ประเภทของการมอดูเลตที่นิยมใช้ประกอบด้วย

1. Pulse Amplitude Modulation (PAM) การ Modulate ทางแอมพลิจูดของพัลส์จะอาศัยหลักการ Sampling ของสัญญาณที่เป็น Analog ตามช่วงเวลา ทำให้สัญญาณนั้นขาดจากกันเป็นพัลส์ ๆ โดยขนาด ของแต่ละพัลส์จะเท่ากับขนาดของสัญญาณเดิมในช่วงเวลา นั้น ๆ ซึ่งจาก

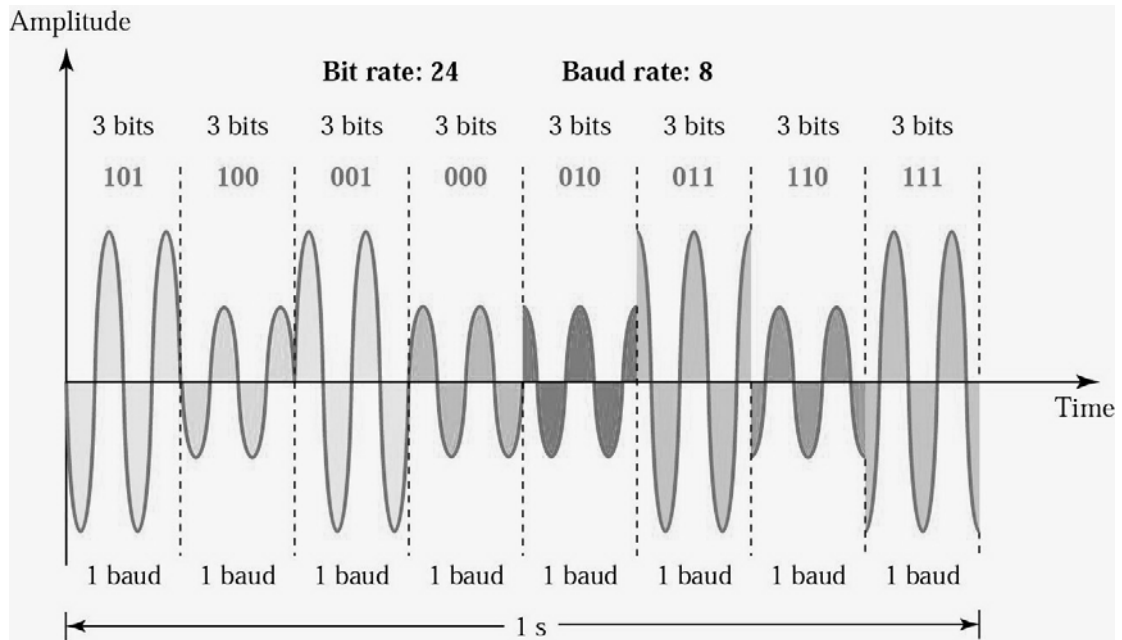
การ Sampling จะทำด้วยอัตรา สองเท่าของแบนด์วิดท์ของ Analog เป็นจำนวนครั้งต่อวินาที (Sampling = 2 BW MHz) หมายความว่า ถ้า Sampling ด้วยอัตราที่น้อยเท่าใด ก็จะได้สัญญาณพัลส์ที่ใกล้เคียงกับสัญญาณเดิม มากที่สุด แต่ถ้าน้อยเกินไปจะเป็น Analog เหมือนเดิม

2. QAM Quadrature Amplitude Modulation (QAM) สำหรับแต่ละวิธีการผสมสัญญาณต่างก็มีข้อเด่น-ด้อย แตกต่างกันไป อาทิเช่น ข้อด้อยของ PSK ได้แก่ การออกแบบอุปกรณ์ที่สามารถจำแนกความแตกต่างของเฟสจำนวนมาก ทำได้ลำบาก ดังนั้นจึงเป็นการเหมาะสมที่จะมีการเสนอให้ใช้ วิธีการ Modulation หลายวิธีรวมกัน เพื่อประสิทธิภาพสูงสุด อย่างไรก็ตาม เนื่องจากเงื่อนไขข้อจำกัดทาง Bandwidth จึงทำให้ FSK ไม่สามารถเข้าร่วมกับ Modulation วิธีอื่นได้ วิธีที่เหมาะสม คือ การใช้ ASK ที่มีความแตกต่าง x กรณี ร่วมกับ PSK ที่มีความแตกต่าง y กรณี ดังนั้นความแตกต่างทั้งหมดที่เป็นไปได้จึงเท่ากับ x คูณ y กรณี เรียกวิธีการ Modulation แบบนี้ว่า Quadrature Amplitude Modulation (QAM) ซึ่งทำให้ความแตกต่างระหว่าง Symbol มีมากที่สุด (ออกแบบอุปกรณ์แยกแยะได้ง่าย) ดังนั้นจำนวนบิตต่อสัญลักษณ์ที่เป็นไปได้เท่ากับ $\log_2(x \times y)$ บิต การรวม PSK กับ ASK ด้วยวิธี QAM ทำได้หลายรูปแบบ แต่วิธีที่นิยมใช้ ได้แก่ QAM ที่มีจำนวน Amplitude Shift น้อยกว่า Phase Shift เนื่องจาก Phase มีความต้านทานต่อสัญญาณรบกวนมากกว่า Amplitude ดังรูป



รูปที่ 2.21 Constellation Diagram ของ 2-QAM(ซ้าย-บน)4-QAM (ซ้าย-ล่าง) และ 16-QAM (ขวา)

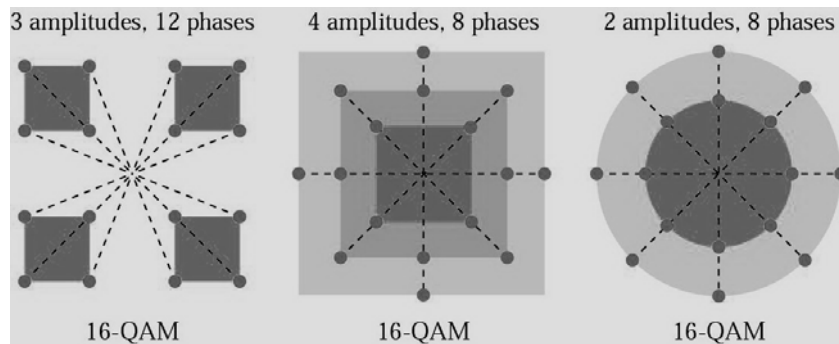
ความแตกต่างของ Phase จะมากกว่า Amplitude



รูปที่ 2.22 การผสมสัญญาณของข้อมูลจำนวน 24 บิต ด้วยวิธีการ QAM
ในกรณีนี้ Bit Rate มีค่าเป็น 3 เท่าของ Baud Rate

เนื่องจาก N - QAM กำหนดเพียงแค่จำนวนของการรวมกันของ Amplitude (A) และ Phase (P) ดังนั้นการปรับโครงสร้างของ A - P Shift เมื่อ N มีค่าเท่ากัน สามารถทำได้ตามความเหมาะสมกับการใช้งานแต่ละประเภท ดังกรณี 16-QAM สามารถจัดโครงสร้างของ A - P Shift ได้หลายรูปแบบดังนี้

- 3A/12P ทนต่อสัญญาณรบกวนมากที่สุด เพราะอัตราส่วน P/A สูง (ITU-T)
- 4A/ 8P จำแนกความแตกต่างระหว่างสัญญาณได้ดี การออกแบบอุปกรณ์จึงสามารถทำได้ง่าย เนื่องจากแต่ละค่า A มี P เพียง 4 รูปแบบ (OSI)
- 2A/ 8P QAM พื้นฐาน



รูปที่ 2.23 โครงสร้าง 16-QAM ที่แตกต่างกัน

แถบความถี่ของ QAM เท่ากับ PSK และเท่ากับ ASK (เมื่อส่งด้วยอัตรา Baud เท่ากัน) ดังสมการ

$$BW_{QAM} = N_{baud}$$

เพื่อให้ระบบสามารถยอมรับอัตราการผิดพลาดของข้อมูล เราสามารถจะเลือกเทคนิคการมอดูเลตโดยอัตโนมัติตามขนาดของแถบความถี่ที่ใช้การส่ง การปรับแต่ง (Adaptive) เทคนิคการมอดูเลตแบบนี้ทำให้คุณภาพการรับ-ส่งดีขึ้นไม่ว่าสถานะของช่องสื่อสารจะดีหรือไม่ดีก็ตาม สิ่งก็ตามมาคือการปรับแต่งการมอดูเลตทำให้เพิ่มประสิทธิภาพการใช้งานแถบความถี่ของระบบด้วย

4) การเลือนหายของความถี่ที่เลือก (Frequency selective fading)

ในระบบแถบกว้างเช่น OFDM ผลกระทบจากการเคลื่อนที่หลายเส้นทางของความถี่ที่ต่างกัน (หรือสัญญาณพาหะย่อยใน OFDM) จะแตกต่างกัน เพื่อให้เราเข้าใจลักษณะการเลือนหายของความถี่ที่เลือก เรามาลองพิจารณาสิ่งที่เกิดขึ้นกับความถี่ (สัญญาณพาหะย่อย) 2 ความถี่บนแถบความถี่ที่ใช้นำข่าวสารมาจากเครื่องส่งไปยังเครื่องรับ ถ้าความถี่ทั้งสองอยู่ใกล้ซึ่งกันและกัน ดังนั้นความแตกต่างของเส้นทางที่เคลื่อนที่จะมีผลต่อกับความถี่ทั้งสองใกล้เคียงกัน ทำให้การเปลี่ยนแปลงขนาดและเฟสใกล้เคียงกันมาก อย่างไรก็ตามถ้าความถี่ทั้งสองอยู่ห่างกันออกไปทำให้ความสัมพันธ์ระหว่างความถี่ทั้งสองแย่ง เพราะว่าการเลือนของเฟสตามแต่ละเส้นทางจะต่างกันมาก ซึ่งการลดลงของความสัมพันธ์ของความถี่จะขึ้นอยู่กับค่าหน่วงเวลาระหว่างสัญญาณที่รับเข้ามาจากเส้นทางที่ต่างกัน

ดังนั้นการที่หลายๆ ความถี่พาหะมาจับจองอยู่บนแถบความถี่เดียวกันมากจนทำให้ความเปลี่ยนแปลงแต่ละความถี่เกิดความแตกต่างกันมากจะทำให้สัญญาณผิดพลาดได้ ทั้งนี้เพราะว่าขนาดและเฟสของความถี่ที่จะประกอบกันเป็นสัญญาณข่าวสารที่เครื่องรับจะไม่เหมือนกันกับที่เป็นในเครื่องส่ง ปรากฏการณ์นี้เรียกว่า การเลือนหายของความถี่ที่เลือก (Frequency selective fading)

ในสภาวะแวดล้อมที่คลื่นวิทยุมีการเดินทางหลายเส้นทาง การเลือนหายของความถี่ที่เลือกสามารถส่งผลต่อการเปลี่ยนแปลงอย่างสูงต่อกำลังงานที่เครื่องรับรับได้ สำหรับช่องสื่อสารที่ไม่มีสัญญาณจากเส้นทางโดยตรงจากการเครื่องส่ง การเลือนหายของคลื่นวิทยุเหล่านี้อาจสูงถึง 30dB ผลที่ตามมาคืออัตราส่วนของสัญญาณต่อสัญญาณรบกวนลดลงและข้อมูลที่ได้รับมีความผิดพลาดมากขึ้น แต่เนื่องจากข้อมูลถูกส่งไปกับสัญญาณพาหะย่อยที่แยกจากกัน ดังนั้นการแก้ไขข้อผิดพลาดของข้อมูลในแต่ละสัญญาณพาหะย่อยอาจจะใช้วิธีการตรวจสอบและแก้ไขข้อผิดพลาดที่ปลายทาง (Forward error detection and correction)

บทที่ 3

การใช้งานโปรแกรม Simulink

3.1 ที่มาของโปรแกรม Simulink

MATLAB เป็นซอฟต์แวร์ (software) สำหรับการวิเคราะห์เชิงตัวเลข (numerical analysis) เขียนขึ้นโดย Dr. Cleve Moler ตั้งแต่ ค.ศ 1982 ในตอนแรกเป็นการรวบรวมโปรแกรมการคำนวณเกี่ยวกับชุดข้อมูลตาราง (matrices) ที่เขียนขึ้นโดยใช้ภาษาฟอร์แทรน (FORTRAN) แล้วเรียบเรียงเป็นหนังสือชื่อ Demonstration of Matrix Library โดย C.B Moler ซึ่งต่อมามีการพัฒนาจนกลายเป็นโปรแกรมขนาดใหญ่ ที่เขียนโดยใช้ภาษา ซี (C) และแอสเซมเบลอร์ (assembler)

รากฐานของ MATLAB นั้น มาจากโครงการ EISPACK (Eigen System Package) และ LINPACK (Linear System Package) ณ Argonne National Laboratory สหรัฐอเมริกา โดย EISPACK เป็น โปรแกรมย่อยภาษาฟอร์แทรน (FORTRAN SUBROUTINES) ซึ่งจะคำนวณ eigen values และ eigen vectors สำหรับ matrices ประเภทต่าง ๆ ซึ่งจะอธิบายโดยละเอียดในหนังสือ Matrix Eigen System Routines : EISPACK Guide โดย Smith, Boyle, Dongarra, Garbow, Ikebe, Klema และ Moler ปัจจุบันมีการใช้ MATLAB ในการเรียนการสอน ทางวิศวกรรมศาสตร์ในมหาวิทยาลัยทั่วไปในสหรัฐอเมริกา ซอฟต์แวร์เป็นลิขสิทธิ์ของ The MathWorks, Inc., 21 Eliot St., South Natick MA01760

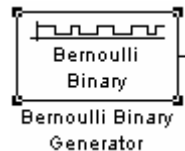
สำหรับด้านการจำลองการทำงานด้วยคอมพิวเตอร์ มีการใช้โปรแกรม Simulink ซึ่งสามารถที่จะเขียนโปรแกรมในลักษณะเป็นหน่วย (unit) ซึ่งแทนด้วยรูปภาพของก้อนวัตถุ (block diagram) แล้วนำข้อมูลจากตัวแปรต่าง ๆ ของ MATLAB, built-in functions และ M-files มาใช้ใน ด้านคำนวณได้หรือแสดงผลจากการออกแบบได้มีความใกล้เคียงกับการทำงานของอุปกรณ์ต่าง ๆ

โปรแกรมประกอบไปด้วย 5 ส่วน คือ

- 3.1 ที่มาของโปรแกรม Simulink
- 3.2 แหล่งจ่าย (Source)
- 3.3 ภาคการเข้ารหัส (Encode)
- 3.4 ภาคการถอดรหัส (Decode)
- 3.5 ภาคส่ง (Transmitter)
- 3.6 ภาครับ (Receiver)
- 3.7 ภาคการปรับตัว (Adaptive)
- 3.8 ภาคการแสดงผล

3.2)แหล่งจ่าย(Source)

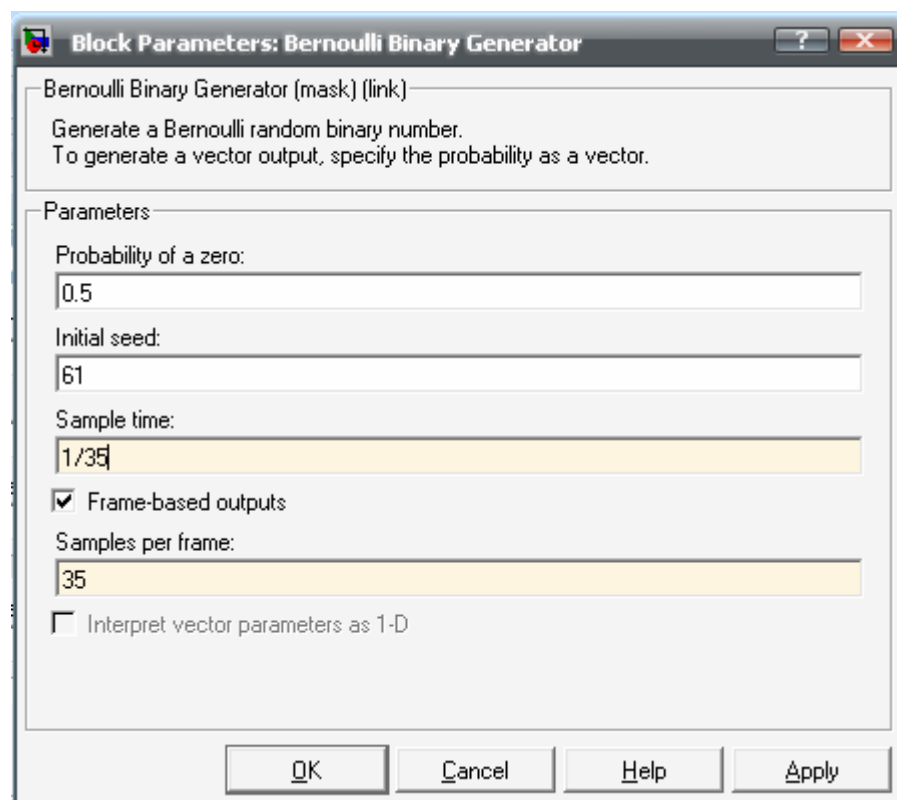
เป็นการจำลองการรับค่าจากผู้ใช้งานจริงโดยเราจะทำการ โดยจะมีการรับข้อมูลเป็นเฟรม แต่ละเฟรมจะรับข้อมูลครั้งละ 35 บิต เหตุที่เราจำเป็นต้องกำหนดข้อมูลให้ใช้ 35 บิตนี้ เพื่อที่เราจะได้ทำการเปรียบเทียบผลในการเข้ารหัสว่ามีค่า bit error rate สูงกว่ากัน โดยบล็อกที่ใช้ในโปรแกรม simulink คือ



รูปที่ 3.1 บล็อกไดอะแกรม Bernoulli Binary Generator

บล็อก Bernoulli Binary Generator บล็อกเครื่องกำเนิด ก่อให้เกิดเลขบิตโดยการสุ่มจำนวนกำลังใช้ Bernoulli การแบ่งสรร. Bernoulli การแบ่งสรรกับปัจจัยกำหนด p ผลิตศูนย์ด้วยความน่าจะเป็น p และหนึ่งด้วยความน่าจะเป็น $1-p$

ความน่าจะเป็นของปัจจัยกำหนดศูนย์ระดับ p ,และสามารถจำนวนจริงอื่นระหว่างศูนย์และหนึ่ง.



รูปที่ 3.2 การตั้งค่าบล็อกไดอะแกรม Bernoulli Binary Generator

Probability of a zero:0.5

ความน่าจะเป็นของโอกาสที่จะเกิดค่าศูนย์ขึ้นในการสุ่ม

Initial seed:61

แจกจ่ายค่าสำหรับจำนวน โดยการสุ่มเครื่องกำเนิด ในความยาวของ
เวกเตอร์ซึ่งเหมือนกัน Probability of a zero

Sample time:1/35

ระยะเวลาของแต่ละการชักตัวอย่างจากข้อมูล(Sample-based) เวกเตอร์
หรือแต่ละแถวของเฟรมข้อมูล(Frame-based) แม่แบบ

Frame-based outputs:35

ตัดสินใจไม่ว่าที่สิ่งทีออกมาเป็นframe-based หรือsample-based . หาก
กระตุ้นการใช้งานจะทำการแปลปัจจัยกำหนดเวกเตอร์อย่างที1-D ไม่ได้
รับการควบคุม

Samples per frame

กำหนดจำนวนตัวอย่างในแต่ละแถว เกี่ยวกับframe-based สัญญาณสิ่งที่
ออกมา เมื่อทำการกระตุ้นบล็อก Frame-based outputs

Interpret vector parameters as 1-D

ถ้าเลือกบล็อกนี้ให้ทำงาน,จะทำให้ข้อมูลที่ออกมาเป็นone-dimensional
สัญญาณ. มิฉะนั้น,ข้อมูลที่ออกมาเป็นtwo-dimensional สัญญาณ. เมื่อทำ
การกระตุ้นบล็อก Frame-based outputs

3.3)การเข้ารหัส(Encode)

อย่างที่แสดงในรูปกระบวนการการเข้ารหัสประกอบด้วยการเชื่อมเข้าด้วยกัน
เกี่ยวกับข้างนอกReed-Solomon (RS)รหัสและภายในconvolutional รหัส(CC)อย่างทีFEC
แบบแผน. ข้อมูลผ่านในรูปแบบบล็อกผ่าน RS encodeจากนั้นแล้ว,มันไปข้าม
convolutional encode โดยกระบวนการสร้างรหัสแบบยัดหยุ่นนั้นต้องอาศัย puncturing
โดยจะทำการตัดแต่งสัญญาณเพื่อให้ได้อัตราเร็วของการสร้างรหัส สุดท้ายขึ้นส่วนของตัว
เข้ารหัสเป็นกระบวนการของการสอดแทรก(interleaving) เพื่อหลีกเลี่ยง การเกิดความ
ผิดพลาด

ในขั้นตอนการทำ FEC (Forward Error Coding) การเข้ารหัสที่ถูกต้องและการเปลี่ยนแปลงคุณภาพให้ขนาดบล็อกและอัตรารหัสใช้สำหรับการปรับซึ่งแตกต่างดังตารางที่ 3.1

AMC	Modulation	RS code	CC
1	2-PAM	(12,12,0)	1/2
2	4-QAM	(32,24,4)	2/3
3	4-QAM	(40,36,2)	5/6
4	16-QAM	(64,48,4)	2/3
5	16-QAM	(80,72,4)	5/6
6	64-QAM	(108,96,6)	3/4
7	64-QAM	(120,108,6)	5/6

ตารางที่ 3.1 แสดงการเข้ารหัสในแบบต่างๆ

ช่องทางซึ่งที่ติดต่อกับสถานีฐาน (BS) และอัตราการเข้ารหัส, ยึดหลักรายงานนี้, การถอดรหัสซึ่งจะจงถูกเลือกสำหรับ downlink การส่งมาข้อมูล. ดังนั้นเงื่อนไขช่องทาง, SNR ต่ำ, ที่เวลาใช้ในการเข้ารหัสที่ต่ำ จะมีผลกับการแก้ไขความผิดที่ดีกว่า



รูปที่ 3.3 แสดงแผนภาพไดอะแกรมการเข้ารหัส (Encode)

-Reed Solomon encode

โดยการเข้ารหัสนี้จะได้ผลการเข้ารหัสไม่เป็นไบนารี (nonbinary Cyclic Codes) ในส่วนของสัญญาณสัญลักษณ์ (Symbol) ทำขึ้นโดย m-bit โดยที่ m เป็นจำนวนเต็มบวกมีค่ามากกว่า 2 รหัส R-S(n, k, t) จะได้ดังนี้

n: คือ จำนวนรวมของสัญลักษณ์รหัสในบล็อกการเข้ารหัส

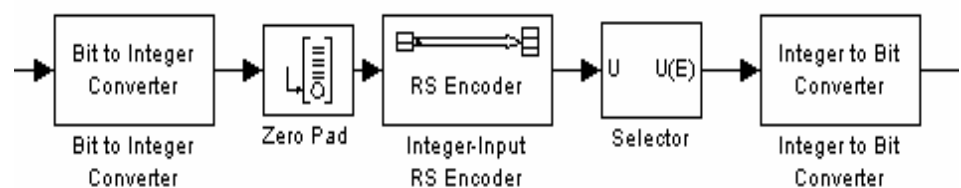
k: คือจำนวนสัญลักษณ์ของข้อมูลเริ่มต้นในการเข้ารหัส

t: คือสัญลักษณ์ความผิดพลาดที่สามารถแก้ไขได้ของรหัส

โดยที่ $n - k = 2t$ (จำนวนสัญลักษณ์ของรหัสแก้ไข(parity)) จำนวนที่ใช้ในการป้องกันการเกิดความผิดพลาดในreed-solomonจะคำนวณได้จาก

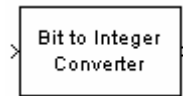
$$N_{RS} = \frac{S_{\text{packet}}}{8k}$$

ในส่วนการทำงานของโปรแกรม simulink มีหลักการทำงานคือเตรียมข้อมูลอินพุทของRSตัวเข้ารหัสในรูปแม่แบบ โดย”Bit to integer Converter-block” ตรงกันกับการคำนวณจากความยาวของแวกก่อนการเข้ารหัส (k) ,และจะได้ค่า $n-k = 2t$ ($2t = 8$) ต่อไปที่บล็อก “Zero pad-block” ทำการเติมค่าคงที่ใดๆลงไปด้านหน้าของแวกข้อมูลที่ทำให้ใดความยาวของ 239 ไบต์สำหรับแต่ละ เมื่อข้อมูลมาถึง จัดสรรเวกเตอร์ใหม่โดยข้อมูลที่ออกมาของตัวเข้ารหัสใน”Reed -solomon-block” จะมีขนาดขนาด $255 \times N_{RS}$ ขึ้นต่อบล็อก “Selectrows-block” ไปเลือกช่วงข้อมูลถูกต้องของไบต์ภายหลังกระบวนการการเข้ารหัส ดังนั้นช่วงข้อมูลที่เลือกนั้นเราจะเลือกเอาแค่ข้อมูลจริงกับค่า $2t = 8$ ไบต์ของไบต์โดยไม่สนใจในส่วนของการเติมศูนย์ ขึ้นตอนสุดท้าย” Integer to bit Converter-block ”บล็อกนี้จะทำการเรียงเรียงข้อมูลกับเข้าไปใหม่โดยจะใช้ค่า $2t$ และพร้อมสำหรับการส่งต่อเพื่อเข้ารหัสconvolutional ในลำดับต่อไป

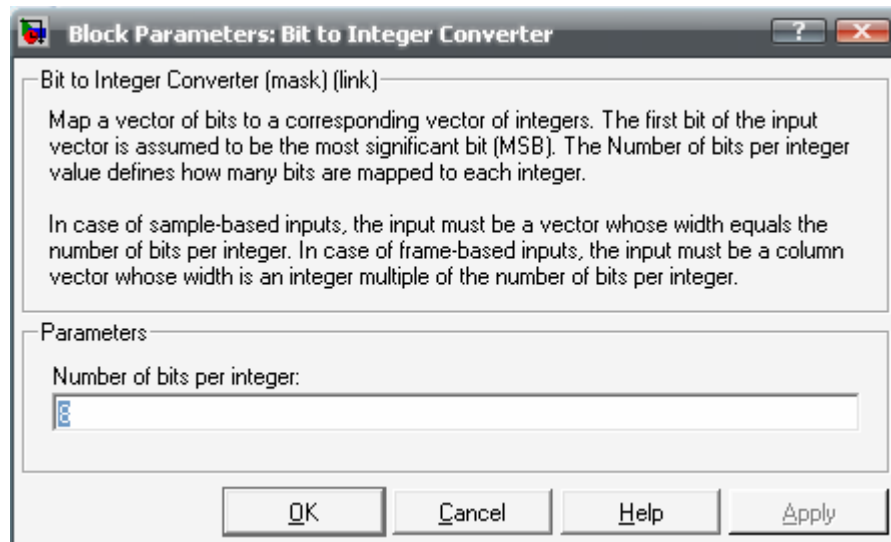


รูปที่3.4แสดงแผนภาพไดอะแกรมการเข้ารหัสReed-Solomon

3.3.1) Bit to integer Converter-block



รูปที่ 3.5 บล็อกไดอะแกรม Bit to integer Converter



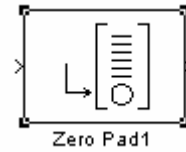
รูปที่ 3.6 การตั้งค่าบล็อกไดอะแกรม Bit to integer Converter-block

The Bit to Integer Converter block จะทำหน้าที่เปลี่ยนแปลงจากจำนวนบิต (Bit) เปลี่ยนแปลงเป็นจำนวน Integer โดยจะใช้ค่าในช่วงจำนวนเต็ม (M) ระหว่าง 0 และ $2^M - 1$ อย่างที่ผลลัพธ์, ความยาวเวกเตอร์สิ่งทีออกมาเป็นความยาวเวกเตอร์ $1/M$ จากการคำนวณตามสมการเมื่อ $n = 255$, $k = 239$ จะได้ว่า $255 - 239 = 8$ bits ดังนั้นเราจึงแทนค่า $M = 8$

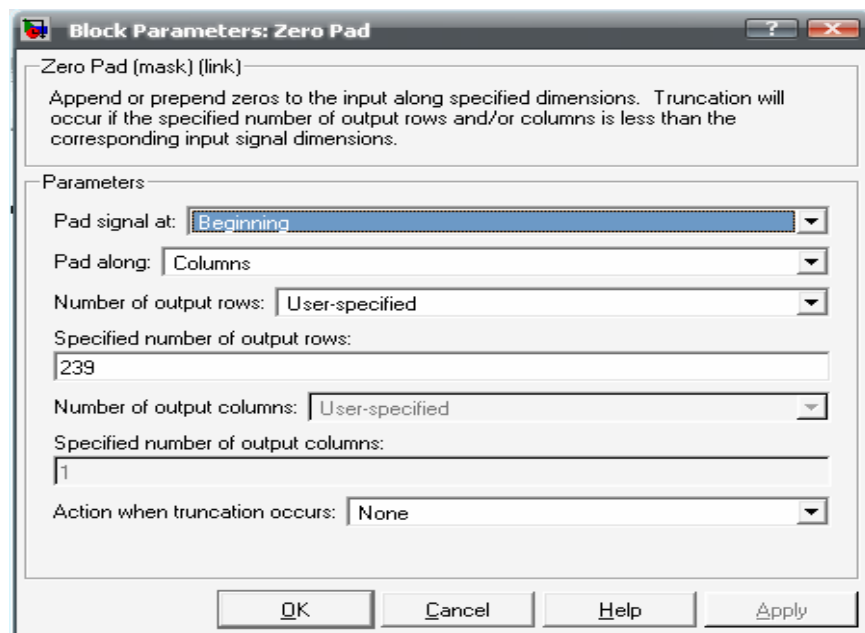
Number of bits per integer: 8

จำนวนเต็มที่ได้ในช่องนี้ได้จะอยู่ในช่วง 1-31

3.3.2) Zero Pad block



รูปที่ 3.7 บล็อกไดอะแกรม Zero Pad



รูปที่ 3.8 การตั้งค่าบล็อกไดอะแกรม

การทำงานของบล็อกนี้ก็เป็นเพียงการเติมศูนย์ลงในคอลัมน์ของข้อมูลโดยจะสามารถกำหนดเติมทางด้านหน้า และต้องทราบค่าข้อมูลทางด้านขวาออกว่ามีจำนวนเท่าใด โดยในข้างต้นเราทราบแล้วว่าข้อมูลทางด้านขวาออกของเราคือ 239 นั่นเอง Pad signal at: Beginning

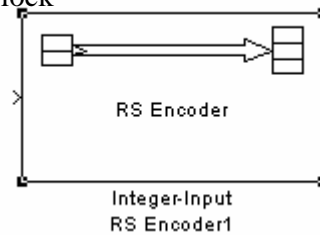
Pad along: Columns

Number of output rows: User-specified

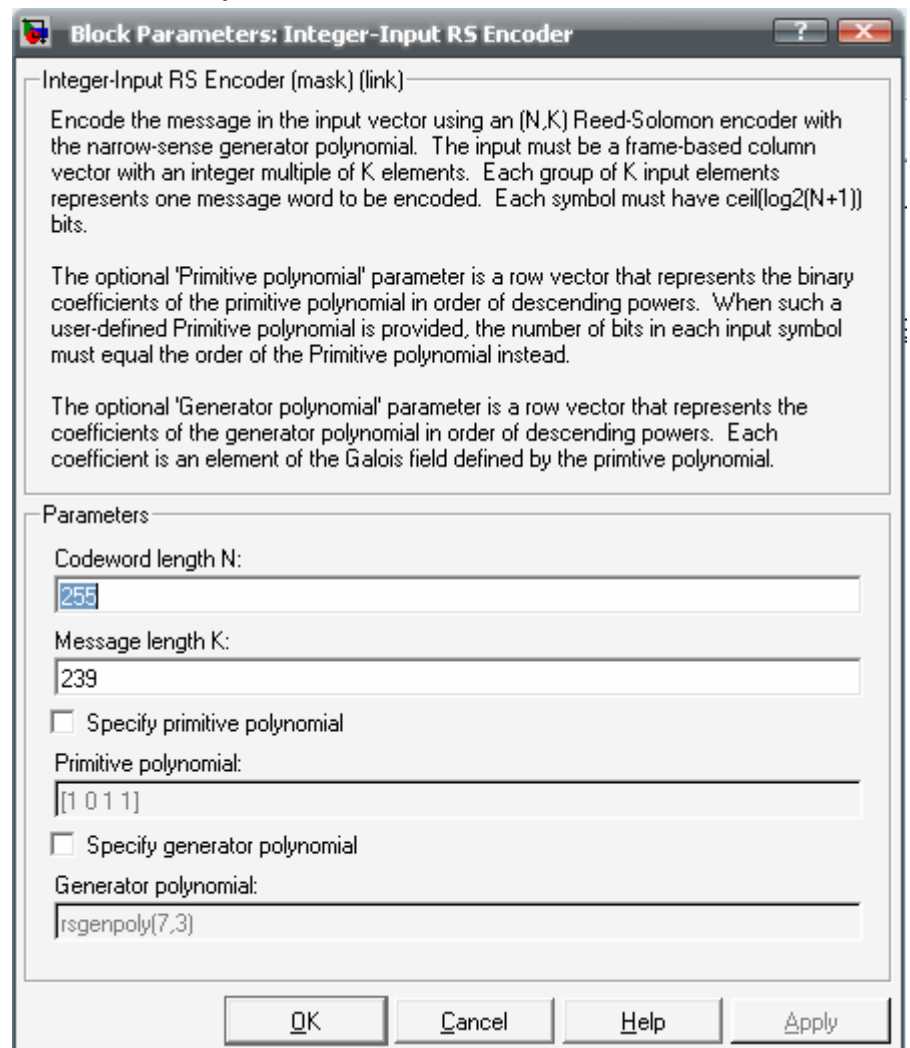
Specified number of output rows =239

Action when truncation occurs: None

3.3.3) Reed –Solomon encode-block



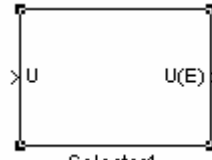
รูปที่3.9 บล็อกไดอะแกรม Reed –Solomon encode



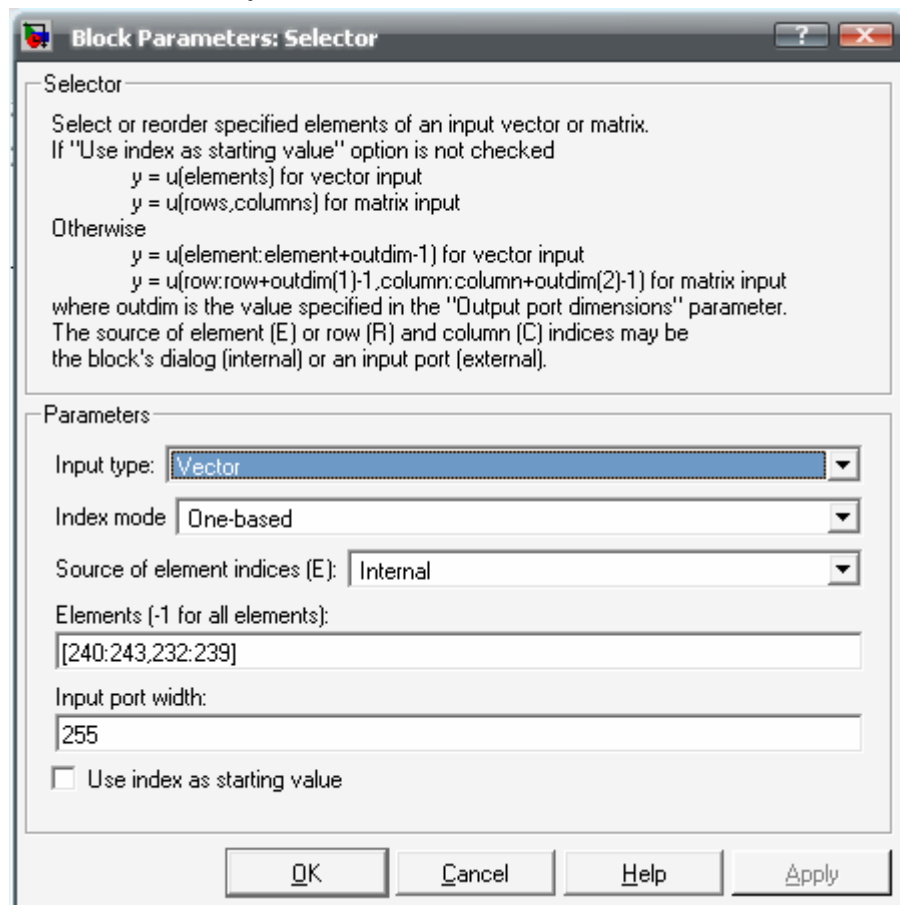
รูปที่3.10การตั้งค่าบล็อกไดอะแกรม Reed –Solomon encode

โดยการทำงานของInteger-Input บล็อกRSตัวเข้ารหัสสร้างReed-Solomon รหัสกับความยาวข่าวสาร $K=239$ และcodeword ความยาว $N=255$ คุณระบุทั้งคู่ N และ K เข้าที่บล็อกโดยตรง. สัญลักษณ์สำหรับรหัสเป็นจำนวนเต็มระหว่าง 0 และ 2^M-1 , โดยจะอาศัยค่าที่ใช้ดังนี้

3.3.4) Selectrows-block



รูปที่3.11 บล็อกไดอะแกรม Selectrows



รูปที่3.12 การตั้งค่าบล็อกไดอะแกรม Selectrows

ทำหน้าที่เลือกช่วงของสิ่งที่เราต้องการนำค่ามาใช้คือช่วงของข้อมูลที่เรานำเข้าไป ดังตารางที่3.12 โดยไม่สนใจการเติมศูนย์ และจะรับค่าข้อมูลในรูปเวกเตอร์ เมื่อข้อมูลทั้งหมดในด้านขาเข้าข้อมูลนี้คือ 255 จะได้ดังนี้

Input Type: vector

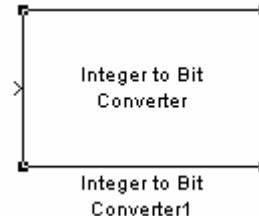
Index mode: One-based

Source of element indices: internal

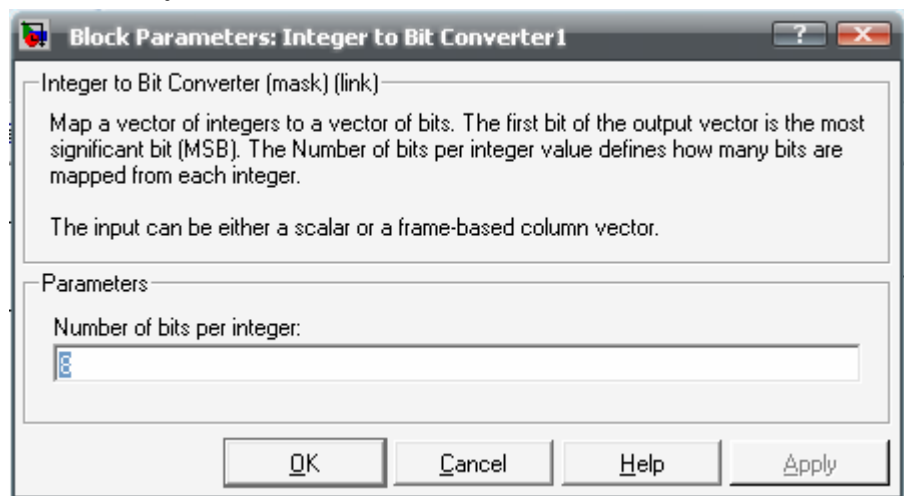
Elements: (ช่วงของข้อมูลที่เรต้องการ)

Input port width: 255

3.3.5) Integer to bit Converter-block



รูปที่3.13 บล็อกไดอะแกรม Integer to bit Converter



รูปที่3.14การตั้งค่าบล็อกไดอะแกรม

Integer to bit Converter-block จะทำหน้าที่เปลี่ยนแปลงจากจำนวน Integer เปลี่ยนแปลงเป็นจำนวนบิต(Bit)โดยจะใช้ค่าในช่วงจำนวนเต็ม(M)ระหว่าง 0 และ 2^M-1 อย่างที่ผลลัพธ์,ความยาวเวกเตอร์สิ่งทีออกมาเป็นความยาวเวกเตอร์ $1/M$ จาก การคำนวณตามสมการเมื่อ $n = 255$, $k = 239$ จะได้ว่า $255-239=8$ bits ดังนั้นเราจึง แทนค่า $M = 8$

Number of bits per integer: 8

จำนวนเต็มทีใส่ในช่องนี้ได้จะอยู่ในช่วง 1-31

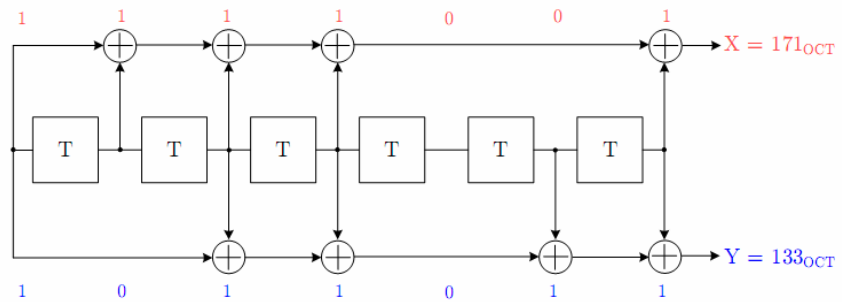
3.3.6) Convolution encode

หลังจากหลังจากการเข้ารหัสด้วยกระบวนการR-S ข้อมูลที่เป็นbits จะถูกเข้ารหัสด้วย Convolution encode โดยมีอัตราเร็วของการเข้ารหัสธรรมเท่ากับ $1/2$ และความ

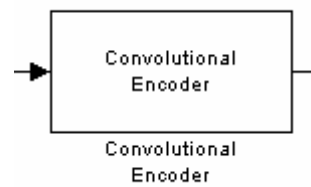
ยาวเท่ากับ 7 generator polynomials ถูกใช้เพื่อผลทำให้มีเอาพุตต่อออกทั้งแกน X และ แกน Y ถูกระบุในการแสดงออกซึ่งตามมานี้

$$G_1 = 171_{\text{oct}} \text{ for } X$$

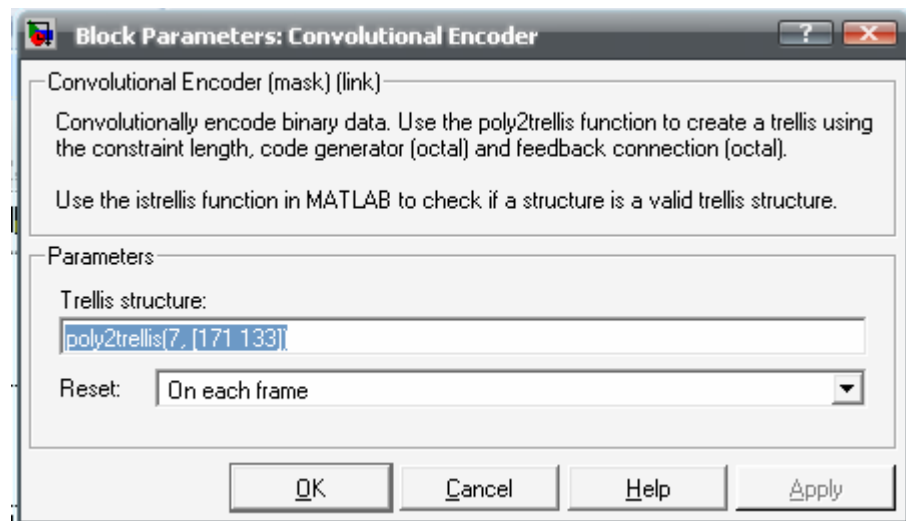
$$G_2 = 133_{\text{oct}} \text{ for } Y$$



รูปที่ 3.15 การทำงานของ Convolution encode



รูปที่ 3.16 บล็อกไดอะแกรม Convolution encode



รูปที่ 3.17 การตั้งค่าบล็อกไดอะแกรม Convolution encode

Convolutional Block ตัวเข้ารหัสเข้ารหัสผลที่ตามกันมาของเวกเตอร์อินพุตเลขคู่ เพื่อผลิตผลที่ตามกันมาของเวกเตอร์สิ่งที้ออกมาเลขคู่ จะเป็นดังนี้

Trellis structure: poly2trellis(7, [171 133])

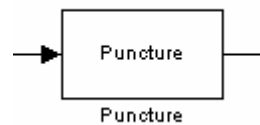
Reset: On each frame

3.3.7)Puncturing process

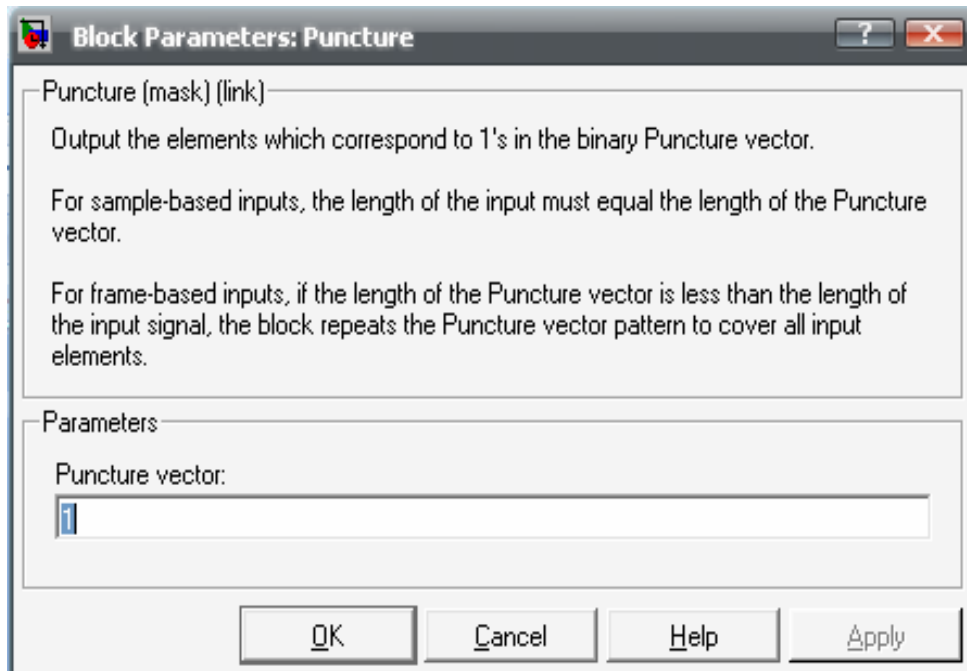
กระบวนการกำหนดอัตราเร็วในการเข้ารหัส โดยจะลดจำนวนรวมของข้อมูล เพื่อการส่งผ่านข้อมูลได้ดีขึ้นจะได้ดังนี้

Rate Puncture vector

1/2	[1]
2/3	[1 1 1 0]
3/4	[1 1 0 1 1 0]
5/6	[1 1 0 1 1 0 0 1 1 0]



รูปที่3.18 บล็อกไดอะแกรมPuncture



รูปที่3.19 การตั้งค่าบล็อกไดอะแกรมPuncture

3.3.8)Interleaver

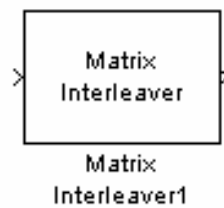
interleaver จะมีวิธีการ 2 ขั้นตอนคือ สิ่งแรกนำข้อมูลผ่านเมทริกซ์ ทำการป้องกันรหัสโดยการกำหนดแถวเมทริกซ์ 192 จากนั้นก็ทำการป้องกันรหัสคอลัมน์(N_c)โดยทั้งคอลัมน์จะเป็นไปตามข้อมูลในด้านอินพุต

$$N_{\text{input}} = 192 \times N_c$$

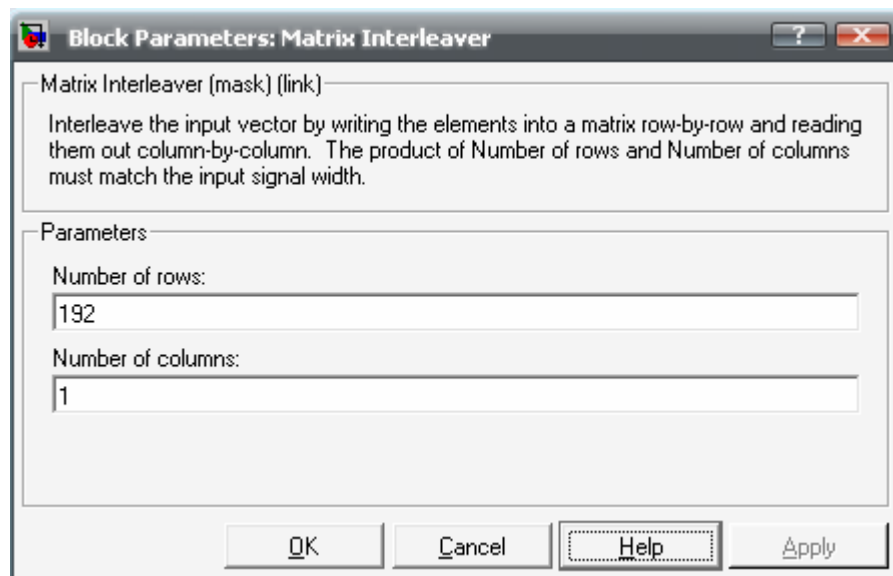
จะทำให้ได้ค่าดังตารางที่3.2

AMC	row	column
1	192	1
2	192	1
3	192	2
4	192	4
5	192	4
6	192	6
7	192	6

ตารางที่ 3.2 แสดงแถวและคอลัมน์ที่ใช้ Interleaver



รูปที่ 3.20 บล็อกไดอะแกรม Matrix Interleaver



รูปที่ 3.21 การตั้งค่าบล็อกไดอะแกรม Matrix Interleaver

3.4)การถอดรหัส(Decode)

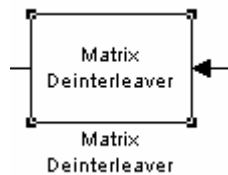
ตัวถอดรหัสนี้จะรับผลที่ตามกันมาของบิตหรือLLRsจากdemapper และ,ตามวิธีการการเข้ารหัสซึ่งถูกใช้ความพยายามเพื่อจำลองข่าวสารตั้งแต่เดิม ก่อให้เกิดโดยต้นเหตุเหมือนในบล็อกตัวเข้ารหัส,ตัวถอดรหัสถูกเรียงเรียงของสี่ขั้นตอน โดยแต่ละอันจะดำเนินการการกระทำซึ่งสวนทางกันกับกระบวนการเข้ารหัส



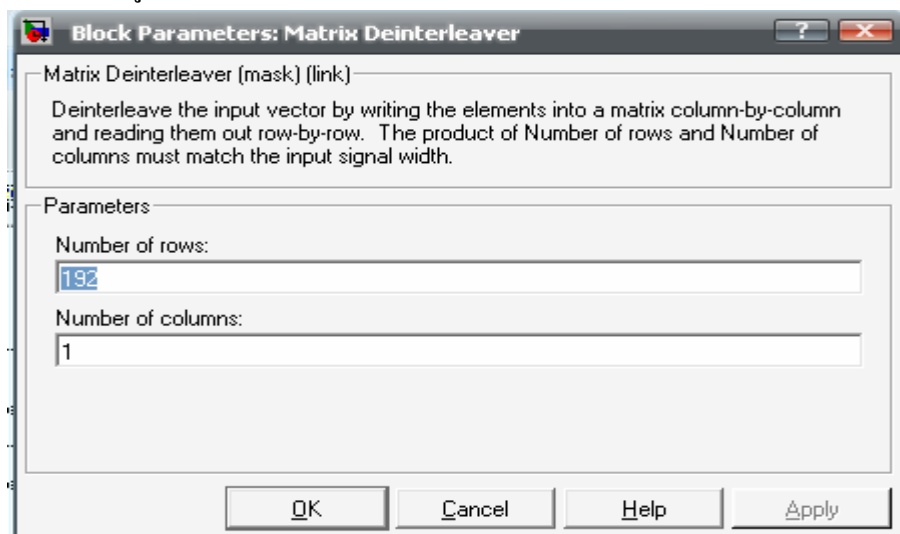
รูปที่ 3.22 แสดงขั้นตอนการถอดรหัส

3.4.1)Deinterleaving

ขั้นตอนนี้จะเหมือนการทำ interleaver แต่จะทำการย้อนกลับโดยกำหนดค่าเดียวกันในการตั้งค่าพารามิเตอร์ก็ยังคงใช้ค่าจากตารางที่ 3.2



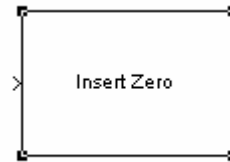
รูปที่3.23 บล็อกไดอะแกรม Deinterleaving



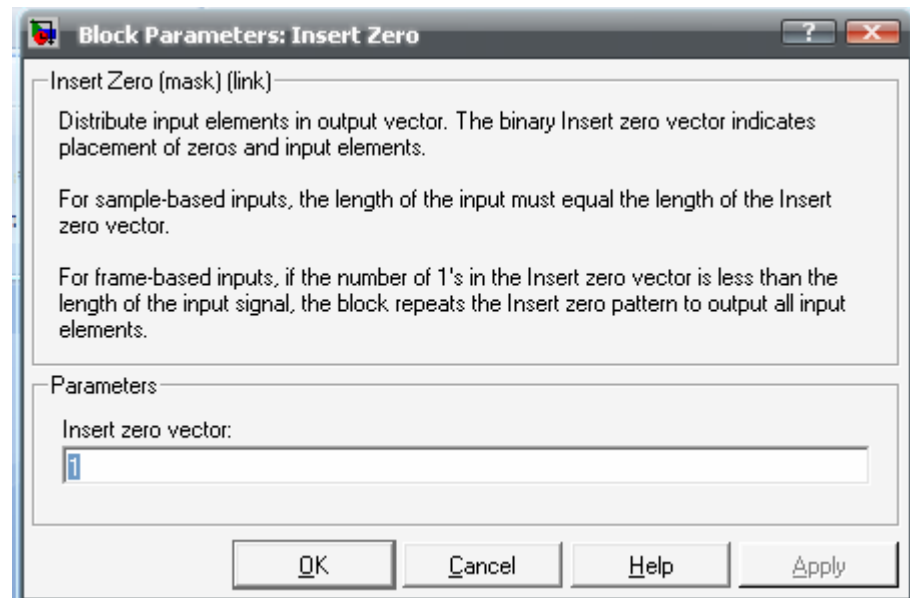
รูปที่3.24การตั้งค่าบล็อกไดอะแกรม Deinterleaving

3.4.2) Inserting zeros

โดยบล็อก Inserting zero นี้จะมีขั้นตอนการทำงานการย้อนกลับกันกับบล็อก Puncturing โดยใช้ค่า Rate Puncture vector เหมือนกันในแต่ละ AMC



รูปที่ 3.25 บล็อก ไดอะแกรม Inserting zero

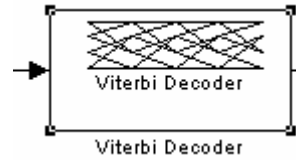


รูปที่ 3.26 การตั้งค่าบล็อก ไดอะแกรม Inserting zero

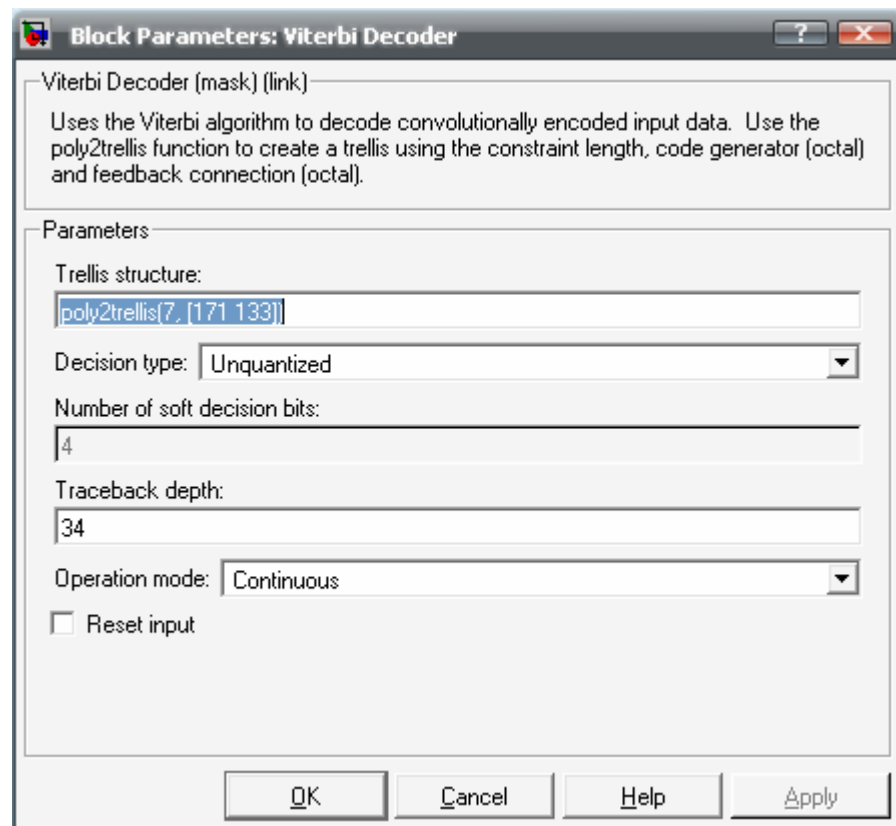
3.4.3) Viterbi decoder

โดยบล็อก Viterbi decoder นี้จะมีขั้นตอนการทำงานการย้อนกลับกันกับบล็อก Convolution encode

การถอดรหัส Convolution นั้นมีขั้นตอนที่ซับซ้อนกว่าวิธีการเข้ารหัสมาก ซึ่งการถอดรหัสมีหลายวิธีแต่การถอดรหัสด้วย Viterbi Algorithm นั้นเป็นวิธีที่ให้ประสิทธิภาพสูง ซึ่งลักษณะการทำงานของ Viterbi Algorithm เป็นแบบ Maximum Likelihood Decoding โดยผลลัพธ์ที่ได้จากการถอดรหัสจะเป็นเส้นทางเพียงเส้นทางเดียวที่มีความน่าจะเป็นสูงสุดจากเส้นทางทั้งหมดใน Trellis Diagram ซึ่งมีลักษณะเหมือนกับข้อมูลที่ถูกลบมากที่สุด



รูปที่ 3.27 บล็อกไดอะแกรม Viterbi decoder



รูปที่ 3.28 การตั้งค่าบล็อกไดอะแกรม Viterbi decoder

Trellis structure: poly2trellis(7, [171 133])

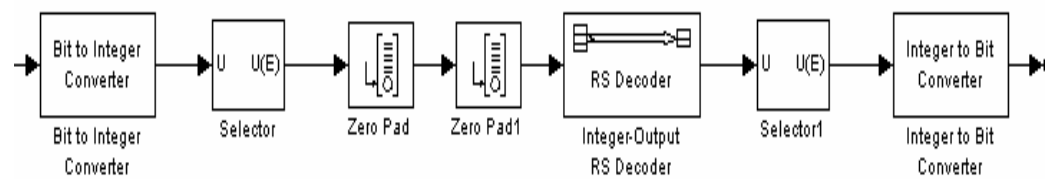
Decision type: Unquantized

Traceback depth: 34

Operation mode: Continuous

3.4.4) Reed-Solomon decoder

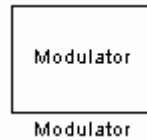
โดยบล็อก Reed-Solomon decoder นี้จะมีขั้นตอนการทำงาน เพื่อที่จะถอดรหัสข้อมูลออกจาก Reed-Solomon encoder โดยจะใช้ บล็อกโคอะแกรมต่างเหมือนกับ Reed-Solomon encoder จึงไม่ขออธิบาย ในส่วนของบล็อกโคอะแกรมแต่จะขอกล่าวถึงในขั้นตอนการทำงาน



รูปที่ 3.29 แสดงขั้นตอนการถอดรหัสของ Reed-Solomon decoder

ของตัวถอดรหัสในเมทริกซ์ โดย "Bit to integer Converter-block" ตรงกันกับการคำนวณจากความยาวของแถวก่อนการเข้ารหัส (k) ,และจะ ได้ค่า $n-k = 2t$ ($2t = 8$) ต่อไป "Select rows-block" ไปเลือกช่วงข้อมูล ถูกต้องของไบต์ที่ได้รับมาจากกระบวนการ Viterbi decoder ที่บล็อก "Zero pad-block" ทำการเติมค่าคงที่ใดๆลงไปด้านหน้าของแถวข้อมูล ที่ ให้ได้ความยาวของ 239 ไบต์ และจะทำการเติมค่าคงที่ใดๆลงไปอีกครั้งที่ แถวข้อมูลที่ทำให้ได้ความยาวของ 255 ไบต์แต่ครั้งนี้เป็นการเติมในด้านหลัง สำหรับแต่ละ เมื่อข้อมูลมาถึง จดสรรเวกเตอร์ใหม่โดยข้อมูลที่ออกมาของ ตัวถอดรหัสใน "Reed -solomon-block" จะมีขนาดขนาด $255 \times N_{RS}$ ขึ้นต่อ ที่บล็อก "Select rows-block" ไปเลือกช่วงข้อมูลถูกต้องของไบต์ภายหลัง กระบวนการการถอดรหัสด้วย ดังนั้นช่วงข้อมูลที่เลือกนั้นเราจะเลือกเอา แค่ข้อมูลจริงของไบต์โดยไม่สนใจในส่วนของการเติมศูนย์ ขั้นตอน สุดท้าย" Integer to bit Converter-block "บล็อกนี้จะทำการเรียงเรียง ข้อมูลกับเข้าไปใหม่โดยจะใช้ค่า $2t$

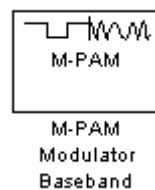
3.5 ภาคส่ง (Transmitter)



รูปที่ 3.30 บล็อกไดอะแกรม Modulator

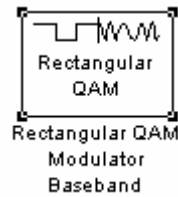
3.5.1 Modulator

ภายใน Modulator ประกอบไปด้วยการมอดูเลตทั้งหมด 4 แบบดังนี้



รูปที่ 3.31 บล็อกไดอะแกรม PAM-Modulator

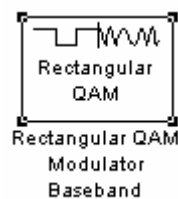
- BPSK (Binary phase-shift keying) มีการใช้บล็อกไดอะแกรมของ PAM เพื่อสะดวกในการกำหนดค่า Parameters ดังนี้
 - M-ary number : 2
 - Input type : Bit
 - Constellation Ordering : Gray
 - Normalization method : Average Power
 - Average power (watts) : 2
 - Sample per symbol : 1



รูปที่ 3.32 บล็อกไดอะแกรม QAM-Modulator(QPSK)

- QPSK (Quadrature phase-shift keying) มีการใช้บล็อกไดอะแกรมของ QAM เพื่อสะดวกในการกำหนดค่า Parameters ดังนี้

- M-ary number : 4
- Input type : Bit
- Constellation Ordering : Gray
- Normalization method : Average Power
- Average power (watts) : 1
- Phase offset (rad) : 0
- Sample per symbol : 2

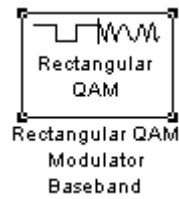


รูปที่ 3.33 บล็อกไดอะแกรม QAM-Modulator

- 16-QAM (16-Quadrature amplitude modulation) มีการใช้ บล็อกไดอะแกรมของ QAM ในการกำหนดค่า Parameters ดังนี้

- M-ary number : 16
- Input type : Bit
- Constellation Ordering : Gray
- Normalization method : Average Power

- Average power (watts) : 1
- Phase offset (rad) : 0
- Sample per symbol : 4

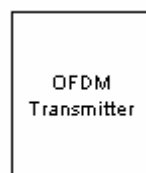


รูปที่ 3.34 บล็อกไดอะแกรม PAM-Modulator

- 64-QAM (64- Quadrature amplitude modulation) มีการใช้
บล็อกไดอะแกรมของ QAM ในการกำหนดค่า Parameters ดังนี้

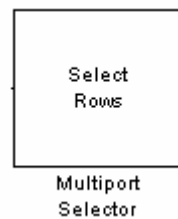
- M-ary number : 64
- Input type : Bit
- Constellation Ordering : Gray
- Normalization method : Average Power
- Average power (watts) : 1
- Phase offset (rad) : 0
- Sample per symbol : 6

3.5.2 OFDM Transmitter



รูปที่ 3.35 บล็อกไดอะแกรม OFDM Transmitter

OFDM (Orthogonal Frequency Division Multiplex) เป็นเทคโนโลยีที่แบ่งช่องสัญญาณออกเป็นหลายๆช่องย่อยตามความถี่เช่นเดียวกับที่เราแบ่งช่องสัญญาณในเทคโนโลยี FDMA แต่คุณสมบัติที่เหนือกว่า คือ คลื่นพาห้จะ Orthogonal กับคลื่นพาห้ของช่องสัญญาณอื่นๆที่เหลือทั้งหมด ดังนั้นสัญญาณที่ได้จะเป็นอิสระต่อกัน ไม่มีการรบกวนกันระหว่างคลื่นพาห้จะทำให้ใช้สเปกตรัมได้สูงสุด ประกอบไปด้วยบล็อกไดอะแกรมดังต่อไปนี้



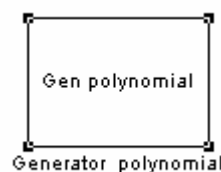
รูปที่ 3.36 บล็อกไดอะแกรม Multiport selector

- **Multiport selector**

ทำการแปลงข้อมูลจากแนวนอนให้เป็นแนวตั้งโดยแบ่งเป็นแถวทั้งหมด 10 แถว โดยกำหนด Parameters ดังนี้

- Select : Row
- Indices to output:

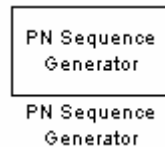
{1:21,22:40,41:59,60:78,79:97,98:116,117:135,136:154,155:173,174:192}



รูปที่ 3.37 บล็อกไดอะแกรม Generator polynomial

- Generator polynomial

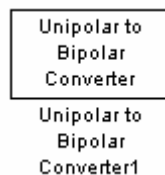
ทำการสร้างค่าที่เป็น pilot เพื่อคั่นระหว่างข้อมูล มีบล็อกไดอะแกรมดังนี้



รูปที่ 3.38 บล็อกไดอะแกรม PN sequence generator

- PN sequence generator (Pseudonoise) ทำการสร้างข้อมูล polynomial 8 หลัก เพื่อใช้เป็นค่า pilot โดยกำหนด Parameters ดังนี้

- Generator polynomial : [1 0 0 1 0 0 0 1]
- Initial states : [1 1 1 1 1 1 1]
- Shift (or mask) : 0
- Sample time : inf



รูปที่ 3.39 บล็อกไดอะแกรม Unipolar to Bipolar converter

- Unipolar to Bipolar converter ทำการแปลงเลขที่รับจาก PN sequence generator ให้เป็น -1 และ 1 เพื่อป้องกันการที่เลข 0 ติดกันยาวๆและส่งไปทั้งหมด 8 เส้น โดยเส้นที่ 8 มีการกำหนด Gain = -1 โดยกำหนด Parameter ดังนี้

- M-ary number : 2
- Polarity : Negative



รูปที่ 3.40 บล็อกไดอะแกรม Dc carrier

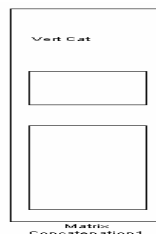
- **Dc carrier** ทำการส่งค่า real และ complex เพื่อให้ Matrix concatenation ทำงานได้ โดยกำหนด Parameters ดังนี้

- Main

- Constant value : 0+0i
- Sample mode : Discrete
- Output : Sample-based
- Sample time : inf

- Data types

- Output data type : Inherit from 'Constant value'

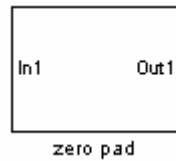


รูปที่ 3.41 บล็อกไดอะแกรม Matrix concatenation

- **Matrix concatenation**

ทำการรวมสัญญาณที่ได้จาก Multiport selector, Generator polynomial, Dc carrier เพื่อส่งข้อมูลเพียงอันเดียวออกไป โดยกำหนด Parameters ดังนี้

- Number of input : 19
- Concatenation method : Vertical



รูปที่ 3.42 บล็อกไดอะแกรม Zero pad(1)

- **Zero pad**

ทำการเพิ่มศูนย์ก่อนทำการ IFFT มีบล็อกไดอะแกรมดังนี้



รูปที่ 3.43 บล็อกไดอะแกรม Zero pad (2)

- Zero pad ทำการเพิ่ม 0 ที่ตัวและเพิ่มส่วนไหนของข้อมูลโดยกำหนด Parameters ดังนี้

- Pad signal at : End
- Pad along : Columns
- Number of output rows : User - Specified
- Specified number of output rows : 256



รูปที่ 3.44 บล็อกไดอะแกรม Selector

- Selector ทำการเลือกส่วนที่จะเพิ่มไว้ที่หัวหรือท้ายข้อมูลและถ้าเลือกท้ายข้อมูลก็จะมี การสลับที่ของข้อมูลเช่นจากข้อมูลช่องที่ 1-100 นำไปไว้ท้ายข้อมูลและข้อมูลช่องที่ 101-256 สลับมาไว้ด้านหน้าโดยกำหนด Parameters ดังนี้

- Input type : Vector
- Index mode : One-based

- Source of element indices (E) : Internal
- Elements (-1 for all elements) : [101:256 1:100]
- Input port width : 256

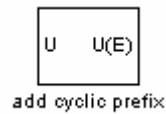


รูปที่ 3.45 บล็อกไดอะแกรม IFFT

- **IFFT** (Inverse Fast Fourier Transform)

ทำการแปลงสัญญาณที่อยู่ในแกนความถี่ให้เป็นแกนเวลาโดย
กำหนด Parameters ดังนี้

- Main
 - Twiddle factor computation : Table lookup
 - Optimize table for : Speed
- Fixed-point
 - Fixed-point operational parameters
 - Rounding mode : Floor
 - Overflow mode : Wrap
- Fixed-point data types
 - Sine table : Specify word length
 - Word length : 16
 - Product output : Same as input
 - Accumulator : Same as input
 - Output : Same as input

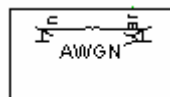


รูปที่ 3.46 บล็อก Add cyclic prefix

- Add cyclic prefix

ทำการป้องกันการรบกวนกันระหว่างช่องสัญญาณหลายเส้นทาง จึงเพิ่ม cyclic โดยมีอัตราการเพิ่มตาม sub carrier ดังนี้ 1/4 , 1/8 , 1/16 , 1/32 ของ sub carrier โดยกำหนด Parameters ดังนี้

- Input type : Vector
- Index mode : One-based
- Source of element indices (E) : Internal
- Elements (-1 for all elements) : [193:256 1:256]
- Input port width : 256



รูปที่ 3.47 บล็อกไดอะแกรม AWGN

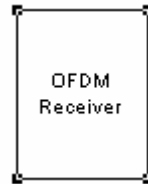
3.5.3 AWGN (Add White Gaussian Noise)

ทำการกำหนดอัตราส่วนระหว่างกับสัญญาณรบกวน (SNR) ของโปรแกรมและมีการกำหนดให้สัญญาณรบกวนเปลี่ยนแปลงได้โดย กำหนด Parameters ดังนี้

- Initial seed : 1
- Mode : Variance from port

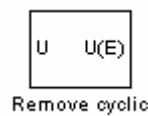
3.6 ภาครับ (Receiver)

3.6.1 OFDM Receiver



รูปที่ 3.48 บล็อกไดอะแกรม OFDM Receiver

ทำการแปลงค่ากลับเพื่อนำข้อมูลมาประมวลผลที่ได้จากการส่งสัญญาณประกอบไปด้วยบล็อกไดอะแกรมดังต่อไปนี้



รูปที่ 3.49 บล็อกไดอะแกรม Remove cyclic

- Remove cyclic

ทำการถอดตัวป้องกันการรบกวนกันระหว่างช่องสัญญาณหลายเส้นออกจากข้อมูล โดยกำหนด Parameters ดังนี้

- Input type : Vector
- Index mode : One-based
- Source of element indices (E) : Internal
- Elements (-1 for all elements) : [65:320]
- Input port width : 320



รูปที่ 3.50 บล็อกไดอะแกรม FFT

- FFT

ทำการแปลงค่ากลับเพื่อให้ได้ข้อมูลดั้งเดิมก่อนทำการ

IFFT โดยกำหนด Parameters ดังนี้

- Main

- Twiddle factor computation : Table lookup

- Optimize table for : Speed

- Fixed-point

- Fixed-point operational parameters

- Rounding mode : Floor

- Overflow mode : Wrap

- Fixed-point data types

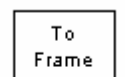
- Sine table : Specify word length

- Word length : 16

- Product output : Same as input

- Accumulator : Same as input

- Output : Same as input



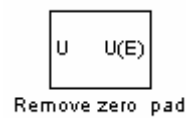
Frame Status
Conversion

รูปที่ 3.51 บล็อกไดอะแกรม Frame Status Conversion

- Frame Status Conversion

ทำการเปลี่ยนค่าที่ได้ให้เป็นเฟรมเหมือนก่อนทำการ IFFT ก่อนที่จะถอดข้อมูลออกจาก Pilot โดยกำหนด Parameters ดังนี้

- Output signal : Frame-based

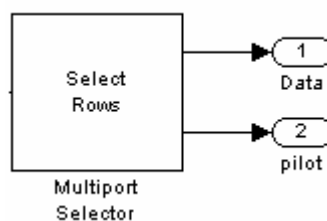


รูปที่ 3.52 บล็อกไดอะแกรม Remove Zero pad

- Remove zero pad

เป็นการนำเลข 0 ที่ต่อท้ายออกโดยอาศัยบล็อก selector โดยกำหนด Parameters ดังนี้

- Input type : Vector
- Index mode : One-based
- Source of element indices (E) : Internal
- Elements (-1 for all elements) : [157:256 1:101]
- Input port width : 256



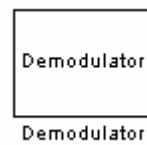
รูปที่ 3.53 บล็อกไดอะแกรม Multiport selector

- Multiport selector

เป็นการนำข้อมูลทั้งหมดแยกออกจาก Pilot โดยมีการกำหนด Parameter ดังนี้

- Select : Rows
- Indices to output:

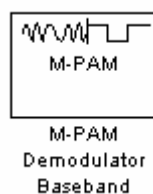
[1:21,23:41,43:61,63:81,83:101,103:121,123:141,143:161,163:181,183:201], [22 42 62 82 122 142 162 182]



รูปที่ 3.54 บล็อกไดอะแกรม Demodulator

3.6.2 Demodulator

ภายใน Demodulator ประกอบไปด้วยการดีมอดูเลตทั้งหมด 4 แบบดังนี้



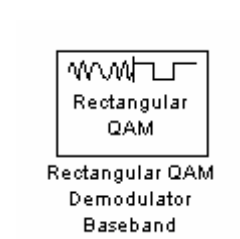
รูปที่ 3.55 บล็อกไดอะแกรม PAM-Demodulator

- BPSK (Binary phase-shift keying)

มีการใช้บล็อกไดอะแกรมของ PAM เพื่อสะดวกในการกำหนดค่า Parameters ดังนี้

- M-ary number : 2

- Input type : Bit
- Constellation Ordering : Gray
- Normalization method : Average Power
- Average power (watts) : 2
- Sample per symbol : 1

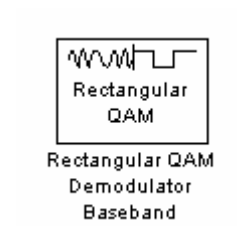


รูปที่ 3.56 บล็อกไดอะแกรม QAM-Demodulator

- QPSK (Quadrature phase-shift keying)

มีการใช้บล็อกไดอะแกรม ของ QAM เพื่อสะดวกในการ
กำหนดค่า Parameters ดังนี้

- M-ary number : 4
- Input type : Bit
- Constellation Ordering : Gray
- Normalization method : Average Power
- Average power (watts) : 1
- Phase offset (rad) : 0
- Sample per symbol : 2



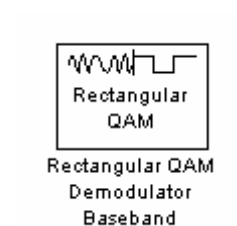
รูปที่ 3.57 บล็อกไดอะแกรม QAM-Demodulator(1)

- 16-QAM (16-Quadrature amplitude modulation)

มีการใช้บล็อกไดอะแกรม ของ QAM ในการกำหนดค่า

Parameters ดังนี้

- M-ary number : 16
- Input type : Bit
- Constellation Ordering : Gray
- Normalization method : Average Power
- Average power (watts) : 1
- Phase offset (rad) : 0
- Sample per symbol : 4



รูปที่ 3.58 บล็อกไดอะแกรม QAM-Demodulator(2)

- 64-QAM (64- Quadrature amplitude modulation) มีการใช้บล็อกไดอะแกรม ของ QAM ในการกำหนดค่า Parameters ดังนี้

- M-ary number : 64

- Input type : Bit
- Normalization method : Average Power
- Average power (watts) : 1
- Phase offset (rad) : 0
- Sample per symbol : 6

3.7 ภาคการปรับตัว (Adaptive)

ทำการควบคุมการเพิ่มหรือลดอัตราส่วนระหว่างสัญญาณต่อสัญญาณรบกวน (SNR) โดยเมื่อค่า SNR เพิ่มขึ้นหรือลดลงก็จะมี การปรับเปลี่ยนการมอดูเลตตามค่าเป้าหมายที่กำหนดไว้ ประกอบไปด้วยบล็อกไดอะแกรมดังต่อไปนี้



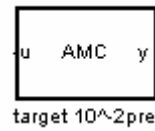
รูปที่ 3.59 บล็อกไดอะแกรม Constant

- Constant

ทำการกำหนดค่า snr ว่าจะให้เพิ่มขึ้นหรือลดลงโดยกำหนด Parameters ดังนี้

- Main
 - Constants value : [จำนวนจริง]
 - Sample time : inf
- Signal data types
 - Output data type mode : Inherit from 'Constants

value'



รูปที่ 3.60 บล็อกไดอะแกรม AMC

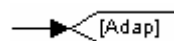
- AMC

ทำการตัดสินใจว่าเมื่อ snr เพิ่มขึ้นควรใช้การมอดูเลตแบบใดและเมื่อ snr ลดลงควรใช้การมอดูเลตแบบใดโดยส่งคำสั่งไปที่บล็อกไดอะแกรม Modulator Demodulator โดยการเขียนโปรแกรมควบคุม เช่น

$1 < \text{snr} < 5$	ใช้การมอดูเลตแบบ BPSK
$5 < \text{snr} < 12$	ใช้การมอดูเลตแบบ QPSK
$12 < \text{snr} < 16$	ใช้การมอดูเลตแบบ 16-QAM
$16 < \text{snr} < 25$	ใช้การมอดูเลตแบบ 64-QAM

- Adapt

เป็นตัวส่งคำสั่งไปที่บล็อก Modulator Demodulator โดยที่ตัว Modulator Demodulator ก็จะมีตัวรับคำสั่งเช่นกัน



รูปที่ 3.61 บล็อกไดอะแกรม Adapt

- ที่ตัวส่งคำสั่งกำหนด Parameters ดังนี้

- Tag : Adap
- Tag visibility : local
- Icon display : Tag

- ที่ตัวรับคำสั่งกำหนด Parameters ดังนี้

- Goto Tag : Adap
- Icon display : Tag



รูปที่ 3.62 บล็อกไดอะแกรม Inverse dB

- Inverse dB

ทำการแปลงค่าที่รับมาจากบล็อก constant เพื่อใช้เป็นค่าสัญญาณรบกวน ที่ส่งไปใน AWGN โดยกำหนด Parameters ดังนี้



รูปที่ 3.63 บล็อกไดอะแกรม Gain

- Gain

เป็นการส่งค่าอัตราขยายเพื่อใช้ในการแปลงค่ามี Parameters ดังนี้

- Main

- Gain : -0.1

- Multiplication : Element-wise($K*u$)

- Sample time (-1 for inherited) : -1

- Signal data types

- Output data type mode : Same as input

- Round integer calculations toward : Floor

- Tick on Saturate on integer overflow

- Parameter data types

- Parameter data mode : Same as input

$$10^u$$

รูปที่ 3.64 บล็อกไดอะแกรม Math

- Math

เป็นการนำค่าที่ได้มากลับส่วนตามสมการในParameters ดังนี้

- Function : 10^u
- Output signal type : auto
- Sample time (-1 for inherited) : -1

โปรแกรม AMC ที่ใช้ในการตัดสินใจเลือกการมอดูเลต

โดยกำหนดช่วงการมอดูเลต ดังนี้

- | | |
|--------------------------|-------------------------|
| - $1 < \text{snr} < 5$ | ใช้การมอดูเลตแบบ BPSK |
| - $5 < \text{snr} < 12$ | ใช้การมอดูเลตแบบ QPSK |
| - $12 < \text{snr} < 16$ | ใช้การมอดูเลตแบบ 16-QAM |
| - $16 < \text{snr} < 50$ | ใช้การมอดูเลตแบบ 64-QAM |

และกำหนดให้ BPSK = 1 , QPSK = 2 , 16-QAM = 3 , 64-QAM = 4

U = input (snr) Y = output

if (u >= 1) && (u < 5)

u=1;

elseif (u >= 5) && (u < 12)

u=2;

elseif (u >= 12) && (u < 16)

u=3;

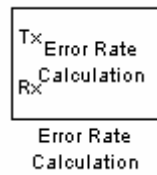
elseif (u >= 16) && (u < 50)

u=4;

else u=1;

end

3.8 ภาคการแสดงผล

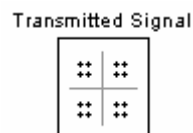


รูปที่ 3.65 บล็อกไดอะแกรม Error rate calculation

- การวัด bit error rate

เป็นการเปรียบเทียบความผิดพลาดในระบบ โดยเป็นอัตราส่วนระหว่างจำนวนบิตผิดพลาดส่วนด้วยบิตทั้งหมดได้โดยใช้บล็อกไดอะแกรม error rate calculation และมีบล็อก display แสดงผลโดยกำหนด Parameters ดังนี้

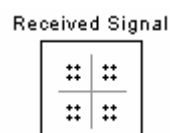
- Receive delay : 0
- Computation delay : 0
- Computation mode : Entire frame
- Output data : Port



รูปที่ 3.66 บล็อกไดอะแกรม Discrete-Time Scatter Plot Scope

- การวัดสัญญาณที่ภาคส่ง

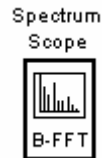
เป็นการแสดงรูป constellation ของการมอดูเลตที่ภาคส่งโดยใช้บล็อกไดอะแกรม Discrete-Time Scatter Plot Scope



รูปที่ 3.67 บล็อกไดอะแกรม Discrete-Time Scatter Plot Scope

- การวัดสัญญาณที่ภาครับ

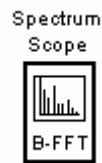
เป็นการแสดงรูป constellation ของการมอดูเลตที่ภาครับโดยใช้
บล็อกไดอะแกรม Discrete-Time Scatter Plot Scope



รูปที่ 3.68 บล็อกไดอะแกรม Spectrum Scope(1)

- การวัดสเปกตรัม

เป็นการวัดสเปกตรัมที่ผ่านช่องสัญญาณมาถ้ารูปร่างคล้ายพัลส์แสดงว่า
สัญญาณที่ส่งมามีคุณภาพสูงแต่ถ้าสัญญาณที่ส่งมามีคุณภาพแย่อาจจะดูไม่ออกเลย
ว่าเป็นการวัดสเปกตรัม โดยใช้บล็อกไดอะแกรม Spectrum Scope



รูปที่ 3.69 บล็อกไดอะแกรม Spectrum Scope(2)

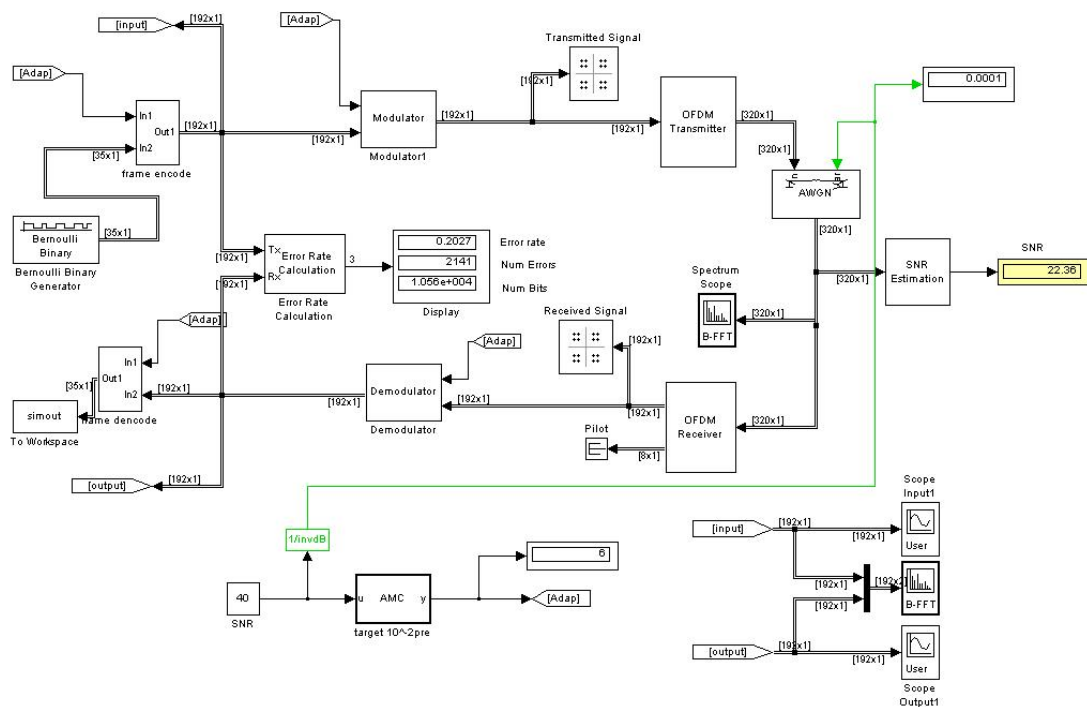
- การเปรียบเทียบกำลังส่งที่ภาคส่งและภาครับ

เป็นการวัดกำลังงานว่าที่ภาคส่งและภาครับมีความแตกต่างกันเพียงใด
โดยใช้บล็อกไดอะแกรม Spectrum Scope

บทที่ 4

ผลการทดลอง

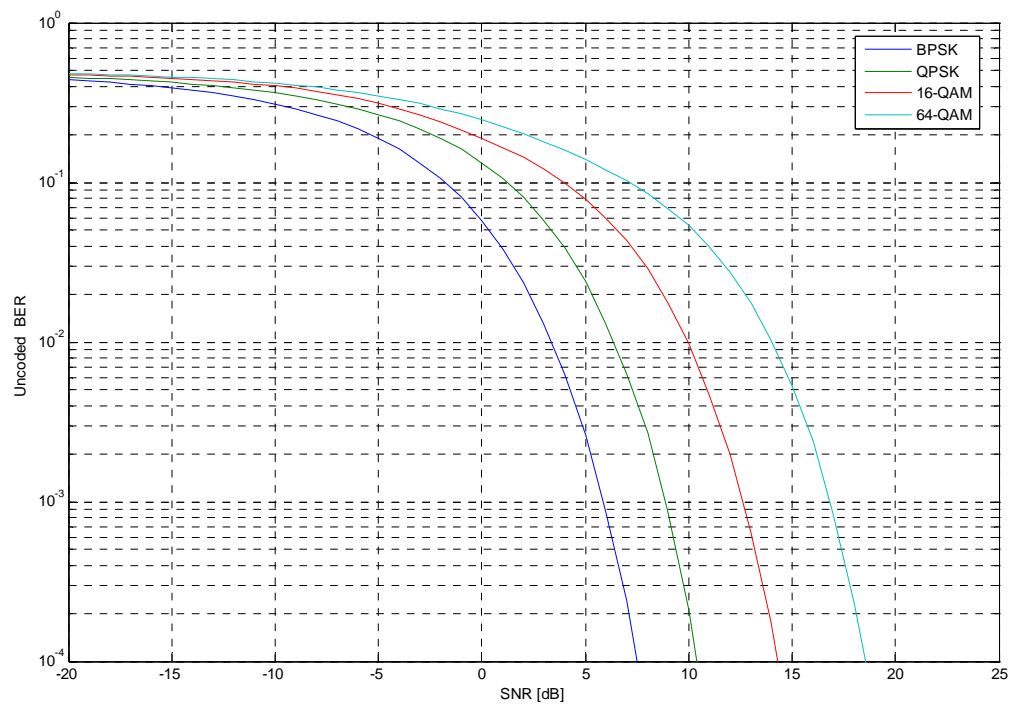
จากการออกแบบจำลอง WiMAX จากการใช้โปรแกรม Simulink จะมีหลักการการทำงานแบ่งออกเป็น 3 ส่วนคือ อันดับแรกรับข้อมูลเข้า(input data)มาทำการเข้ารหัส(Encoder)และส่งต่อไปมอดูเลต(Modulation) โดยใช้เทคนิคการมอดูเลต Orthogonal Frequency Division Multiplex (OFDM)ก็จะสิ้นสุดในภาคส่ง(Transmitter) โดยจะทำการส่งผ่านทางช่องส่งสัญญาณ(Channel AWGN) เมื่อมาถึงในส่วนของภาครับ (Receiver) จะทำการดีมอดูเลต(Demodulation)โดยใช้เทคนิคการมอดูเลตแบบOFDM จากนั้นจึงทำการในส่วนการถอดรหัสสัญญาณ(Decoder)และได้ข้อมูลขาออก(output) โดยในส่วนของภาคปรับตัวจะทำงานเชื่อมโยงกับส่วนการมอดูเลต โดยจะรับค่า SNR(Signal to noise ratio) มาจากช่องส่งสัญญาณ (Channel AWGN) นำมาประมวลผลและสั่งให้ภาคมอดูเลต(Modulation)ที่ภาคส่ง(Transmitter) และดีมอดูเลต(Demodulation)ในภาครับ(Receiver) ดังรูปที่ 4.1



รูปที่ 4.1 แบบจำลอง WiMAX จากการใช้โปรแกรม Simulink

ผลการทดลองทั้งหมดเป็นอัตราการวัดบิตที่ผิดพลาดและการวัดอัตราส่วนสัญญาณต่อสัญญาณรบกวน โดย การวัดอัตราของบิตที่ผิดพลาด (BER) สามารถคำนวณได้จากจำนวนบิตที่ผิดพลาดส่วนด้วยจำนวนบิตทั้งหมด และ อัตราส่วนสัญญาณต่อสัญญาณรบกวน (SNR) สามารถคำนวณได้จากกำลังส่งสัญญาณส่วนด้วยสัญญาณรบกวน และมีการหาค่าข้อมูลที่ส่งออก (Throughput) ได้โดยนำจำนวนบิตข้อมูลที่ส่งทั้งหมดลบด้วยจำนวนบิตที่ผิดพลาดก็จะได้ข้อมูลที่ส่งออกซึ่งค่าทั้งหมดที่ได้มาจากแบบจำลองที่แสดงผลโดยบล็อกไดอะแกรม Error rate calculation

4.1 ผลการทดลองการวัด Bit Error Rate กับ SNR

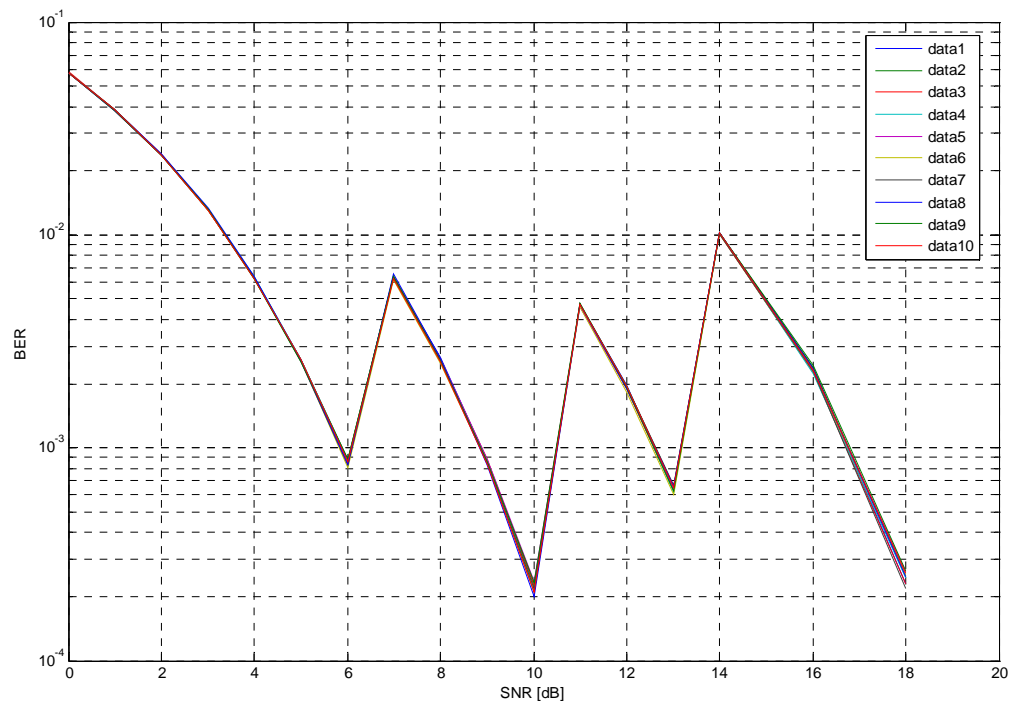


รูปที่ 4.2 กราฟแสดงความสัมพันธ์ระหว่าง SNR กับ Bit Error Rate

จากการทดลองจะเห็นว่ากราฟมีแกน X เป็นค่าของ SNR และแกน Y เป็นค่าของ Bit Error Rate และเส้นกราฟที่ได้มีลักษณะการลดลงของค่า Bit Error Rate ตามค่าของ SNR ที่เพิ่มขึ้น โดยค่าของ BPSK มีค่า SNR ต่ำที่สุด นั่นคือ BPSK ให้ประสิทธิภาพในการทนต่อสัญญาณรบกวนได้สูงสุดแต่ให้ความเร็วในการส่งข้อมูลได้ต่ำสุด ส่วน 64-QAM มีค่า SNR มากที่สุด นั่นคือ ให้ความเร็วในการส่งข้อมูลได้สูงสุดแต่ประสิทธิภาพในการทนต่อสัญญาณรบกวนได้ต่ำสุด

4.2 ผลการทดลองการกำหนดเป้าหมายของจำนวนบิตที่ผิดพลาด

4.2.1 ผลการทดลองการกำหนดเป้าหมายที่ 10^{-2}

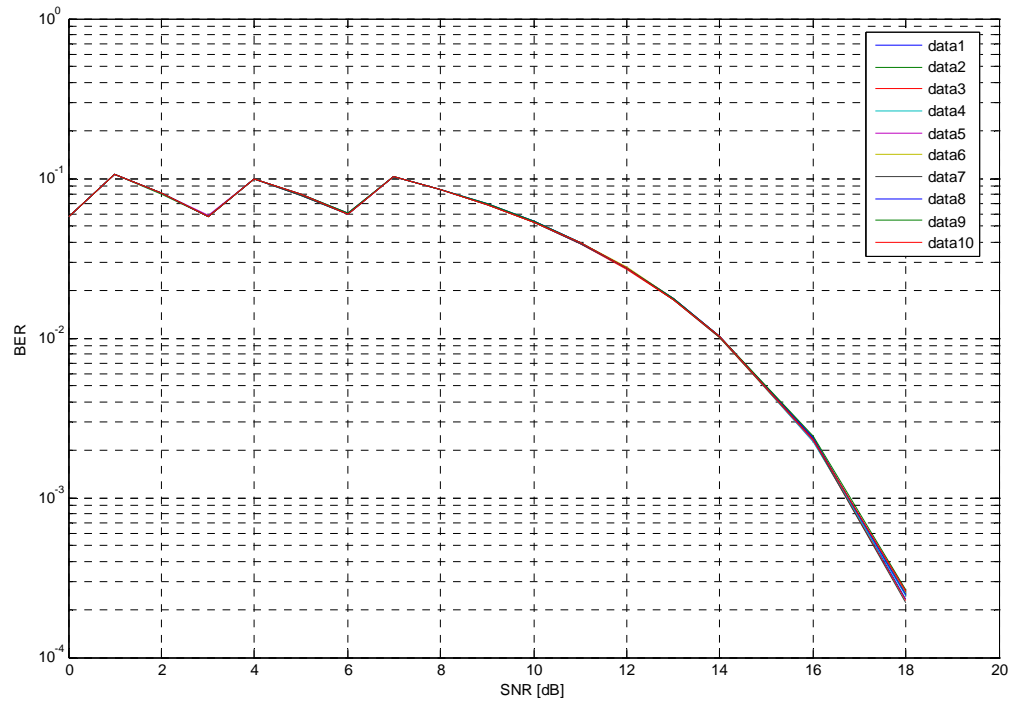


รูปที่ 4.3 กราฟแสดงความสัมพันธ์ระหว่าง SNR กับ BER ของ
การกำหนดเป้าหมายที่ 10^{-2}

จากการทดลองจะเห็นว่ากราฟที่ได้มีลักษณะเพิ่มขึ้นลดลงไม่คงที่เท่าใดนักโดยแกน X เป็นค่าของ SNR และแกน Y เป็นค่าของ Bit Error Rate โดยกราฟที่เพิ่มขึ้นหรือลดลงนั้นเป็นผลมากจากการที่กำหนดช่วงการมอดูเลตของ SNR และจะเห็นว่าค่าของ Bit Error Rate ส่วนใหญ่จะมีค่าไม่เกิน 10^{-2} ที่เป็นเป้าหมายที่กำหนดไว้โดยช่วงที่กำหนดได้มาจากกราฟแสดงความสัมพันธ์ระหว่าง SNR กับ BER เช่น

- การมอดูเลตแบบ BPSK จะอยู่ในช่วงของ SNR ระหว่าง -5 ถึง 4 dB
- การมอดูเลตแบบ QPSK จะอยู่ในช่วงของ SNR ระหว่าง 4 ถึง 7 dB
- การมอดูเลตแบบ 16-QAM จะอยู่ในช่วงของ SNR ระหว่าง 7 ถึง 13 dB
- การมอดูเลตแบบ 64-QAM อยู่ระหว่าง $13 < \text{SNR dB}$

4.2.2 ผลการทดลองการกำหนดเป้าหมายที่ 10^{-1}



รูปที่ 4.4 กราฟแสดงความสัมพันธ์ระหว่าง SNR กับ BER

ของการกำหนดเป้าหมายที่ 10^{-1}

จากการทดลองจะเห็นว่ากราฟที่ได้มีลักษณะเพิ่มขึ้นลดลงในระยะหนึ่งและหลังจากนั้น ก็จะมีการลดลงเรื่อยๆไม่มีการเพิ่มขึ้นเลย โดยแกน X เป็นค่าของ SNR และแกน Y เป็นค่าของ Bit Error Rate โดยกราฟที่เพิ่มขึ้นหรือลดลงนั้นเป็นผลมาจากการที่กำหนดช่วงการมอดูเลตของ SNR และจะเห็นว่าค่าของ Bit Error Rate ส่วนใหญ่จะมีค่าไม่เกิน 10^{-1} ที่เป็นเป้าหมายที่กำหนดไว้โดยช่วงที่กำหนดได้มาจากกราฟแสดงความสัมพันธ์ระหว่าง SNR กับ BER เช่น

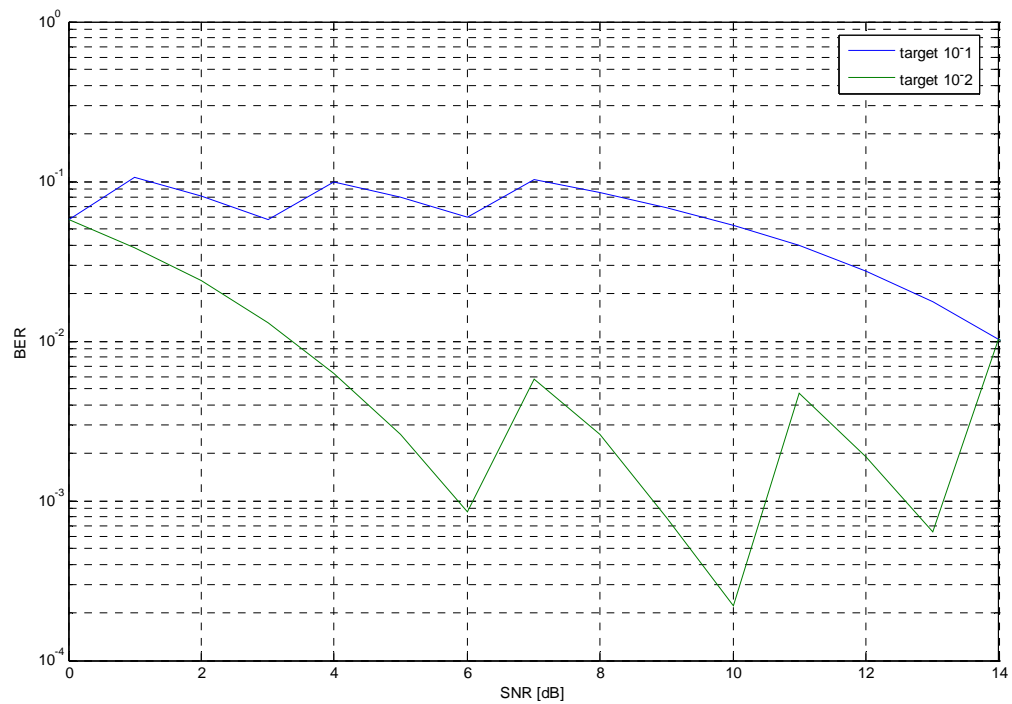
การมอดูเลตแบบ BPSK จะอยู่ในช่วงของ SNR ระหว่าง -5 ถึง 2 dB

การมอดูเลตแบบ QPSK จะอยู่ในช่วงของ SNR ระหว่าง 2 ถึง 4 dB

การมอดูเลตแบบ 16-QAM จะอยู่ในช่วงของ SNR ระหว่าง 4 ถึง 7 dB

การมอดูเลตแบบ 64-QAM อยู่ระหว่าง $7 < \text{SNR dB}$

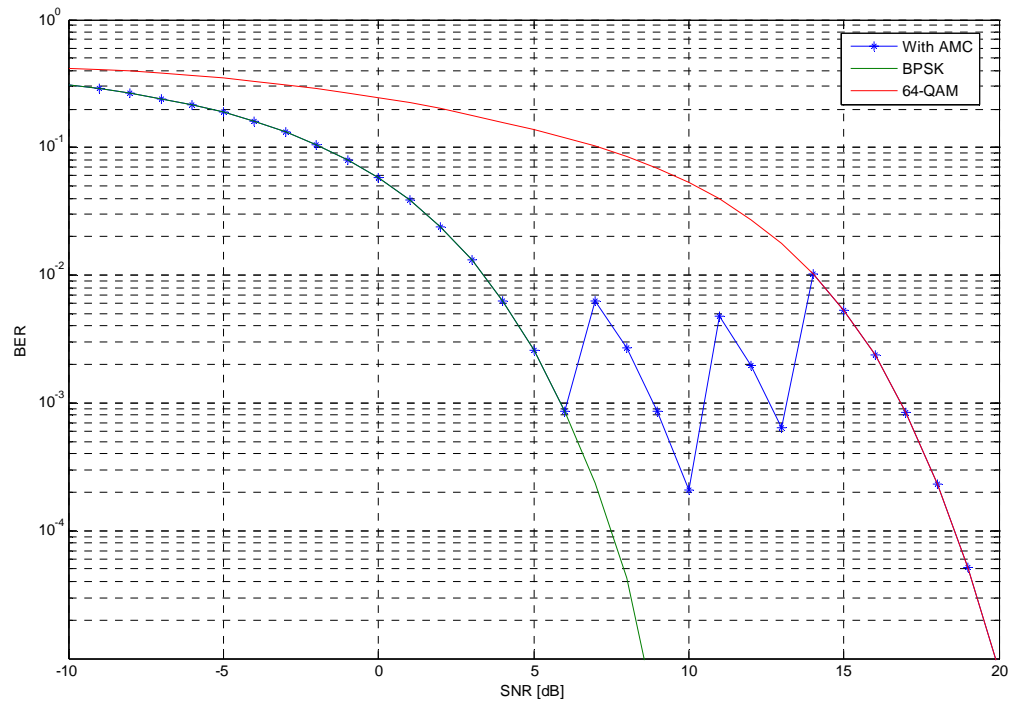
4.2.3 ผลการทดลองการกำหนดเป้าหมายเฉลี่ยระหว่าง 10^{-1} และ 10^{-2}



รูปที่ 4.5 กราฟแสดงความสัมพันธ์ระหว่าง SNR กับ BER ของ
การกำหนดเป้าหมายเฉลี่ยระหว่าง 10^{-1} และ 10^{-2}

จากการทดลองที่มีการกำหนดเป้าหมายที่ 10^{-1} และ 10^{-2} จะเห็นว่า เมื่อมีการเปลี่ยนแปลงการมอดูเลตค่า Bit Error Rate ก็จะมีการเปลี่ยนแปลงทำให้ได้กราฟเพิ่มขึ้นหรือลดลงในช่วง SNR ที่กำหนด โดยแกน X เป็นค่าของ SNR และแกน Y เป็นค่าของ Bit Error Rate โดยกราฟของ 10^{-1} จะมีการเปลี่ยนแปลงในช่วง 0 – 7 dB จากนั้นก็จะลดลงอย่างคงที่ และ กราฟของ 10^{-2} จะมีการเปลี่ยนแปลงในช่วง 6 – 14 dB จากนั้นก็จะลดลงอย่างคงที่ โดยทั้งสองกราฟก็จะไม่มีค่า Bit Error Rate เกินกว่าที่กำหนดไว้ในช่วงของการเปลี่ยนแปลงการมอดูเลตแบบต่างๆ

4.3 ผลการทดลองการวัด Bit Error Rate กับ SNR ของแบบจำลองที่มีการปรับตัวที่เป้าหมายของจำนวนบิตที่ผิดพลาดที่ 10^{-2} และแบบจำลองที่ไม่มีการปรับตัว

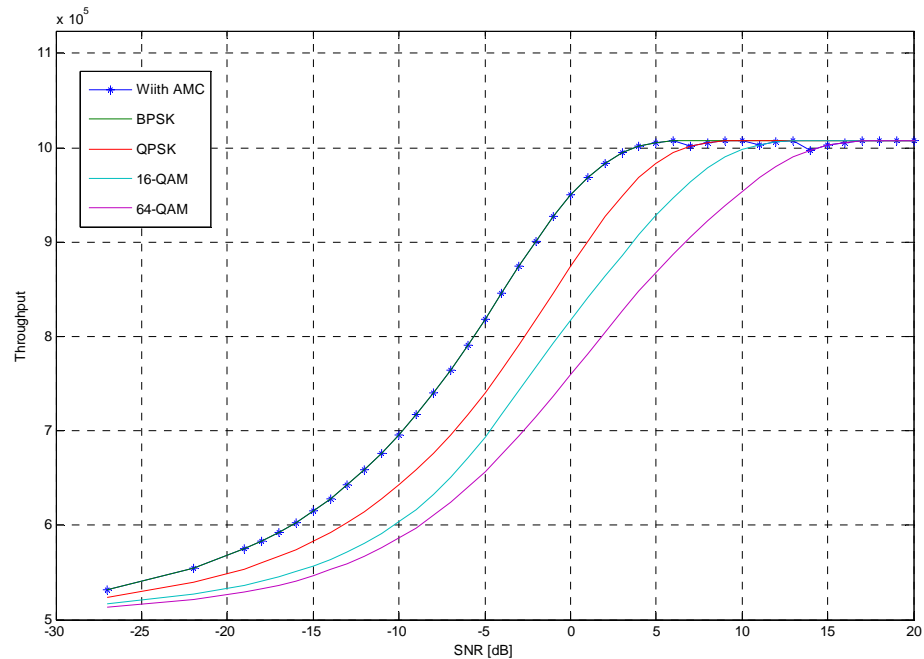


รูปที่ 4.6 กราฟแสดงความสัมพันธ์ระหว่าง SNR กับ Bit Error Rate ของแบบจำลองที่มีการปรับตัวที่เป้าหมายของจำนวนบิตที่ผิดพลาดที่ 10^{-2} และแบบจำลองที่ไม่มีการปรับตัว

จากการทดลองจะเห็นว่ากราฟสองเส้นมีการลดลงอย่างคงที่ตามการเพิ่มขึ้นของค่า SNR ส่วนอีกเส้นหนึ่งมีการเพิ่มขึ้นและลดลงในช่วงๆหนึ่งจากนั้นก็ลดลงอย่างคงที่ โดยแกน X เป็นค่าของ SNR และแกน Y เป็นค่าของ Bit Error Rate โดย กราฟของแบบจำลองที่มีการปรับตัว (AMC) นั้นจะเริ่มเปลี่ยนแปลงตอนที่มีการเปลี่ยนการมอดูเลตตามช่วง SNR ที่กำหนดไว้โดย การปรับตัวนั้นจะเริ่มที่ SNR ตั้งแต่ 6 dB ถึง 14 dB โดยก่อน 6 dB นั้นก็จะมีค่า BER เท่ากับกราฟที่ไม่มีการปรับตัว และหลังจาก 14 dB มีค่า BER เท่ากับกราฟที่ไม่มีการปรับตัว

4.4 ผลการทดลองการวัดจำนวนข้อมูลที่ส่งออกของแบบจำลองที่มีการปรับตัวและไม่มีการปรับตัว

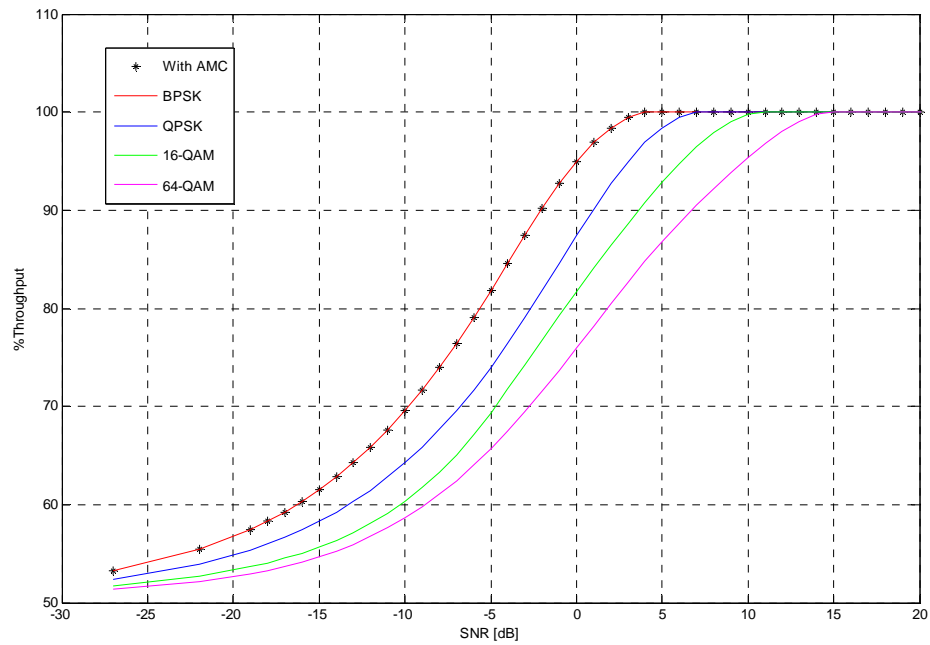
4.4.1 เมื่อจำนวนบิตเท่ากัน



รูปที่ 4.7 กราฟแสดงความสัมพันธ์ระหว่าง SNR กับ throughput เมื่อจำนวนบิตเท่ากัน

จากการทดลองเมื่อมีการกำหนดบิตที่กันจะเห็นว่ากราฟที่มีการส่งข้อมูลออกได้มากที่สุดคือ BPSK และ AMC เพราะว่ามีค่า SNR ต่ำที่สุด โดย 64-QAM ส่งข้อมูลได้น้อยที่สุด เพราะว่ามีค่า SNR มีค่ามากที่สุด โดยแกน X เป็นค่าของ SNR และแกน Y เป็นค่าของ Throughput โดยจากกราฟได้กำหนดจำนวนบิตไว้ที่ 1 ล้านบิต จะเห็นได้ว่าเส้นแรกที่ถึง 1 ล้านบิตก่อน คือ BPSK ต่อมาเป็น QPSK และ 16-QAM และ 64-QAM ตามลำดับ ส่วน AMC นั้นก็จะปรับไปเรื่อยๆ ตามค่า SNR ที่กำหนดไว้ว่ามีการมอดูเลตแบบใดก็จะได้ค่า Throughput ตามค่านั้น

4.4.2 เมื่อคิดเป็นเปอร์เซ็นต์ต่อเวลาที่เท่ากัน



รูปที่ 4.8 กราฟแสดงความสัมพันธ์ระหว่าง SNR กับ throughput เมื่อคิดเป็นเปอร์เซ็นต์

จากการทดลองจะเห็นว่าเมื่อเทียบเป็นเปอร์เซ็นต์จะเห็นว่ากราฟที่ได้จะมีการส่งข้อมูลเพิ่มขึ้นเมื่อค่า SNR มีการเพิ่มขึ้นและจะมีการส่งข้อมูลลดลงเมื่อค่า SNR ลดลงและจากกราฟจะเห็นว่าค่าเปอร์เซ็นต์การส่งข้อมูลของระบบที่มีการปรับตัวจะมีค่าสูงอยู่ตลอดเวลาไม่ว่าค่า SNR ลดลงเพียงใด

บทที่ 5

สรุปผลการทดลอง

จากการทดลองผลที่ได้ทำให้ทราบว่าระบบ WiMAX ใช้การมอดูเลตที่ปรับตัวได้เพื่อการส่งข้อมูลที่ดีขึ้นและจำนวนข้อมูลที่มากขึ้นตามค่าการเพิ่มหรือลดลงของ SNR โดยการกำหนด SNR ก็จะมีเป้าหมายที่ต้องการว่าจะกำหนดว่าจำนวนบิตที่ผิดพลาดมีค่าเท่าไรถึงยอมรับได้ จากนั้นก็จะเป็นการกำหนดว่าค่า SNR เท่าไรควรมีการมอดูเลตแบบใด เช่น

ถ้ากำหนดเป้าหมายที่ 10^{-2}

การมอดูเลตแบบ BPSK อยู่ระหว่าง $-5 < \text{SNR} < 4$

การมอดูเลตแบบ QPSK อยู่ระหว่าง $4 < \text{SNR} < 7$

การมอดูเลตแบบ 16-QAM อยู่ระหว่าง $7 < \text{SNR} < 13$

การมอดูเลตแบบ 64-QAM อยู่ระหว่าง $13 < \text{SNR}$

ถ้ากำหนดเป้าหมายที่ 10^{-1}

การมอดูเลตแบบ BPSK อยู่ระหว่าง $-5 < \text{SNR} < 2$

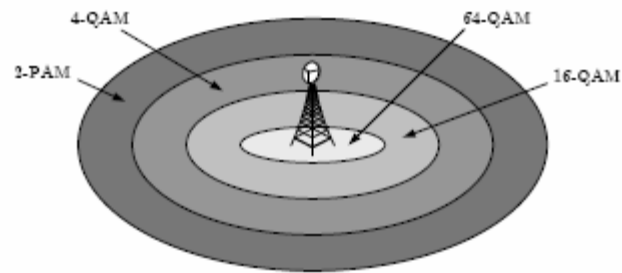
การมอดูเลตแบบ QPSK อยู่ระหว่าง $2 < \text{SNR} < 4$

การมอดูเลตแบบ 16-QAM อยู่ระหว่าง $4 < \text{SNR} < 7$

การมอดูเลตแบบ 64-QAM อยู่ระหว่าง $7 < \text{SNR}$

การกำหนดช่วงของการมอดูเลตนี้เพื่อไว้ทำการตัดสินใจว่าค่า SNR มีค่าเท่าใดจึงเปลี่ยนการมอดูเลตเพื่อการส่งข้อมูลที่มีประสิทธิภาพมากที่สุดเพื่อไม่สิ้นเปลืองพลังงาน

จากผลที่ได้ทำให้ทราบว่าเมื่อมี AMC ทำให้ Throughput ที่ได้มีค่ามากขึ้นซึ่งมากกว่าระบบที่ไม่มี AMC เมื่อเปรียบเทียบในช่วงของ SNR เดียวกัน โดยการมอดูเลตนั้นก็จะเป็นไปตามค่า SNR ที่กำหนดไว้ดังที่แสดงในรูป 5.1



รูปที่ 5.1 แสดงการใช้การมอดูเลตแบบต่างๆเมื่อเทียบกับระยะห่างจากสถานีฐาน

บรรณานุกรม

- 1) ทฤษฎีเทคโนโลยี WiMAX (Worldwide Interoperability of Microwave Access)
- 2) ปราโมทย์ อนันต์วราพงษ์ (2545). OFDM เทคโนโลยีการมัลติเพล็กซ์ในระบบสื่อสารยุคใหม่) กรุงเทพฯ:ซีเอ็ดยูเคชั่น.
- 3) BCH and Reed-Solomon Codes สืบค้นเมื่อวันที่ 5 สิงหาคม 2551 จาก
- 4) www.kmitl.ac.th/dslabs/linkinmainpage/Channel_coding_T/Download/BCH%20and%20ORS%20Codes%201.pdf
- 5) Convolution Code สืบค้นเมื่อวันที่ 7 สิงหาคม 2551 จาก
<http://images.ejeepss.multiply.com/attachment/0/RfTrAoKCqEAABMQetY1/Convolution%20Code.pdf?nmid=21911335>
- 6) A. Ganz, Z. Ganz, and K. Wongthavarawat, “Multimedia Wireless Networks: Technologies, Standards, and QoS”, Prentice Hall PTR, Upper Saddle River, NJ, September 2003.
- 7) D. Sweeney, “WiMAX Operator’s Manual”, Apress, 2004
- 8) K. Wongthavarawat, and A. Ganz, “Packet Scheduling for QoS Support in IEEE802.16 Broadband Wireless Access Systems”, International Journal of Communication Systems, Vol. 16, p81-96, 2003
- 9) IEEE 802.16 Working Group, “IEEE Std 802.16-2004 - Air Interface for Fixed Broadband Wireless Access Systems”
- 10) IEEE 802.16 Task Group e, “P802.16e/Draft 10 – Air Interface for Fixed and Mobile Broadband Wireless Access Systems”
- 11) US FCC, “GN Docket No. 04-163: Wireless Broadband Access Task Force Report”
- 12) US FCC, “Code of Federal Regulation Title 47”, <http://wireless.fcc.gov/rules.html>
- 13) A. Fellah, “WiMAX Spectrum Picture”
- 14) . Intel, “White Paper – Deploying License-Exempt WiMAX Solutions”
- 15) WiMAX Forum, “WiMAX Forum Regulatory Working Group Initial Profiles and Goals”
- 16) WiMAX Forum, “WiMAX Regulatory Take Force: Objectives, Status and Key Issues”
- 17) รู้จัก WiMAX เชิงลึก ตอนแอร์อินเตอร์เฟซ สืบค้นเมื่อวันที่ 7 สิงหาคม 2551 จาก
<http://www.thaiinternetnetwork.com/content/detail.php?id=0514>

ประวัติผู้เขียน



นายกิตติกร ศุภเสถียรกุล เกิดเมื่อวันที่ 5 พฤษภาคม 2527 ภูมิลำเนาเดิมอยู่บ้านเลขที่ 58/2 หมู่ที่ 1 ตำบลบ่อกรู อำเภอดงนาหวาง จังหวัดสุพรรณบุรี จบระดับมัธยมศึกษาตอนปลายจากโรงเรียนนครธรรมศาสตร์ศึกษา จังหวัดสุพรรณบุรี ปีการศึกษา 2545 ปัจจุบันกำลังศึกษาอยู่ชั้นปีที่ 4 สาขาวิชาวิศวกรรมโทรคมนาคม สำนักวิชาวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีสุรนารี จังหวัดนครราชสีมา



นายศุภกิจ ลักขรุช เกิดเมื่อวันที่ 31 พฤษภาคม พ.ศ.2528 ภูมิลำเนาเดิมอยู่บ้านเลขที่ 271 หมู่ที่ 1 ตำบลอาโพน อำเภอบัวเชด จังหวัดสุรินทร์ จบระดับมัธยมศึกษาตอนปลายจากโรงเรียนสุรวิทยาคาร จังหวัดสุรินทร์ ปีการศึกษา 2546 ปัจจุบันกำลังศึกษาอยู่ชั้นปีที่ 4 สาขาวิชาวิศวกรรมโทรคมนาคม สำนักวิชาวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีสุรนารี จังหวัดนครราชสีมา