



รายงานผลการวิจัย

เรื่อง

โปรแกรมช่วยศึกษาและวิจัยสำหรับระบบวิศวกรรมการสื่อสาร 4 : ทงโรจน์ไวยังชีพ

Study and Research Aiding Program for Communication Engineering System

โดย

ภาควิชาวิศวกรรมโทรคมนาคม

มหาวิทยาลัยธุรกิจบัณฑิตย์

รายงานการวิจัยนี้ได้รับทุนอุดหนุนจากมหาวิทยาลัยธุรกิจบัณฑิตย์

2548

ISBN 974-671-325-6

ชื่อเรื่อง : โปรแกรมช่วยศึกษาและวิจัยสำหรับระบบวิศวกรรมการสื่อสาร

ผู้วิจัย : ภาควิชาวิศวกรรมโทรคมนาคม สถาบัน : มหาวิทยาลัยธุรกิจบัณฑิต

ปีที่พิมพ์ : พ.ศ. 2548

สถานที่พิมพ์ : โรงพิมพ์มหาวิทยาลัยธุรกิจบัณฑิต

แหล่งที่เก็บรายงานการวิจัยฉบับสมบูรณ์ จำนวนหน้างานวิจัย : 156 หน้า

: ศูนย์วิจัย มหาวิทยาลัยธุรกิจบัณฑิต

คำสำคัญ

ลิขสิทธิ์ : มหาวิทยาลัยธุรกิจบัณฑิต

: ระบบวิศวกรรมการสื่อสาร, เทคโนโลยีการสื่อสาร, สัญญาณแอนะล็อก, สัญญาณดิจิทัล, สัญญาณสเปกตรัม

บทคัดย่อ

โปรแกรมช่วยศึกษาและวิจัยสำหรับระบบวิศวกรรมการสื่อสาร (SRAPCES) โดยมีวัตถุประสงค์เพื่อทำให้เกิดความสะดวกในด้านการเรียน และการวิจัยสำหรับผู้ศึกษาที่มีความสนใจในระบบการสื่อสาร โดยสามารถที่จะศึกษาผ่านเครื่องคอมพิวเตอร์ (CBT) โดยการทำงานของโปรแกรมนี้ ต้องอาศัยซอฟต์แวร์ของโปรแกรม MATLAB ในการแสดงผล โปรแกรม SRAPCES นี้จะทำให้เกิดสะดวกต่อผู้เรียนและผู้บรรยาย และโปรแกรมนี้ทำการออกแบบเป็นโปรแกรมช่วยศึกษาและวิจัยสำหรับนักศึกษาหรือผู้ที่สนใจในระบบการสื่อสารแบบแอนะล็อก ดิจิทัล และสเปกตรัม โดยโปรแกรมแสดงผลลัพธ์นี้ จะสามารถเห็นรูปลักษณะจริงของกราฟที่เกิดขึ้นเหมือนเรียนภาควิชาปฏิบัติได้ โปรแกรม SRAPCES นี้มีข้อดีคือ ข้อที่ 1 โปรแกรม SRAPCES สามารถแสดงผลเป็นแบบกราฟิกในระบบการสื่อสาร เพื่อให้ผู้เรียนเห็นภาพกราฟิกแทนคำสั่งข้อความพื้นฐานนั้น แม้ผู้เรียนอาจจะไม่คุ้นเคยกับการใช้โปรแกรม MATLAB นี้มาก่อนแต่ผู้เรียนก็สามารถใช้ฟังก์ชันต่าง ๆ ของโปรแกรม MATLAB ได้ ข้อที่ 2 โปรแกรม SRAPCES เป็นการเชื่อมโยงระหว่างหน้าต่างกราฟิกต่าง ๆ และ เป็นการเชื่อมโยงกับ Microsoft FrontPage เพราะ ฉะนั้นการปรับปรุงและการจัดการเนื้อหาให้ตรงตามคำบรรยายสามารถทำได้ง่าย ข้อที่ 3 ผู้เรียนสามารถทำการเปลี่ยนแปลงค่าตัวแปรจากหน้าต่างแสดงผลของกราฟิกนั้นให้เป็นไปในลักษณะเดียวกับตัวแปรทางกายภาพที่เปลี่ยนแปลงให้เป็นระบบตามความเป็นจริงได้ นอกจากนี้โปรแกรม SRAPCES ยังสามารถช่วยทำให้ผู้เรียนมีความเข้าใจทฤษฎีในระบบการสื่อสารมากขึ้น เช่นเดียวกับการปฏิบัติการทดลอง

Title : Study and Research Aiding Program for Communication Engineering System

Researcher : Telecommunication Engineering

Institution : Dhurakijpundit University

Year of publication : B.D.2547

Publisher : Dhurakijpundit University press

Sources

No. of page : 156 pages

: Research Centre Dhurakijpundit University

Keyword

Copy right : Dhurakijpundit University

: Communication Engineering System, Telecommunication Technology, Analog Signal, Digital Signal, Spectrum Signal

Abstract

The aim of Study and Research Aiding Program on Communication Engineering System (SRAPCES) project is to facilitate learning and researching for learners who are interested in Telecommunication through Computer Based Training (CBT). To accomplish this task, specially designed software based on MATLAB is proposed. SRAPCES offers efficiency and conveniently for these learners and lecturers. This program is designed as the SRAPCES for students or people interesting in the analog, digital, and spectrum. The program shows the result in the same as practical laboratory. First, SRAPCES utilized GUI to communicate with the learners instead of text command based. This allows learners who might not be familiar with MATLAB to be able to utilize various functions previously available only to the users who are familiar with MATLAB. Second, SRAPCES allows links between each GUI window and also links to Microsoft FrontPage. Consequently, updating and managing content by lectures can be accomplished without difficulty. Third, learners can adjust various relevant parameters from the offered GUI in the same manner as changing physical parameters in an actual system. In addition, SRAPCES provides graphical result of each adjusting parameter. Thus facilitates learners toward better understanding of the theoretical concepts in Telecommunication just as working in a practical laboratory.

ไม่มีเอกสารจากต้นฉบับ
หน้า กิตติกรรมประกาศ

สารบัญ

	หน้า
บทคัดย่อภาษาไทย	ก
บทคัดย่อภาษาอังกฤษ	ข
สารบัญ	(1)
สารบัญตาราง	(4)
สารบัญรูปภาพ	(5)
บทที่ 1 บทนำ	1
1.1 ความเป็นมาและความสำคัญของปัญหา	1
1.2 วัตถุประสงค์ของการวิจัย	2
1.3 ขอบเขตของการวิจัย	3
1.4 วิธีการดำเนินงานวิจัย	3
1.5 ประโยชน์ที่คาดว่าจะได้รับ	3
1.6 แผนผังงานวิจัย	4
1.7 ผังตารางเวลาการดำเนินงานวิจัย	5
บทที่ 2 หลักการและทฤษฎีเบื้องต้นที่เกี่ยวข้อง	6
2.1 กล่าวนำ	6
2.2 ทฤษฎีเบื้องต้นของระบบการสื่อสารแบบแอนะล็อก	6
2.2.1 อนุกรมฟูรีเยร์	6
2.2.2 การแปลงฟูรีเยร์	9
2.2.3 วงจรสมมูลผ่านต่ำของสัญญาณผ่านแถบ	12
2.2.4 ความหมายของกระบวนการเร็นดัม	14
2.2.5 ฟังก์ชันความหนาแน่นความน่าจะเป็น (Probability Density Function, PDF)	15
2.2.6 กระบวนการเกาส์และเกาส์มาร์คอฟ	16
2.2.7 การแก้ไขปัญหาจากสัญญาณรบกวนเกาส์เซียน	20
2.2.8 การส่งสัญญาณเร็นดัมผ่านฟิลเตอร์	26
2.2.9 การมอดูเลตสัญญาณแอนะล็อก	27

สารบัญ (ต่อ)

	หน้า
2.2.10 การวิเคราะห์สเปกตรัมของสัญญาณเอเอ็ม	30
2.2.11 หลักการในการเปลี่ยนสัญญาณแอนะล็อกไปเป็นสัญญาณดิจิทัล	35
2.3 ทฤษฎีเบื้องต้นของระบบสื่อสารแบบดิจิทัล	41
2.3.1 การมอดูเลตและการดีมอดูเลตสัญญาณ	41
2.3.2 การเข้ารหัสและการถอดรหัส	41
2.3.3 การผิดเพี้ยนของสัญญาณจากการส่งผ่านระบบเชิงเส้น	41
2.3.4 อีควอไลเซอร์	43
2.3.5 การมอดูเลตแบบ AM (Amplitude modulation)	44
2.3.6 ผลกระทบที่เกิดขึ้นในกรณีที่อัตราสุ่มตัวอย่างสูงไม่เพียงพอ	45
2.3.7 แพทเทิร์นรูปดวงตา (eye pattern)	48
2.3.8 การแก้ปัญหาการผิดเพี้ยนของสัญญาณและการรบกวนข้ามสัญญาณ	49
2.3.9 แมทซ์ฟิลเตอร์ (matched filter)	52
2.3.10 การมอดูเลตโดยการเปลี่ยนขนาดของสัญญาณคลื่นพาห้	53
2.3.11 การมอดูเลตเชิงขนาดที่ใช้สัญญาณคลื่นพาห้อินเฟส และคลื่นพาห้ควอดเรเจอร์เฟส	55
2.3.12 การมอดูเลตแบบเปลี่ยนเฟสของสัญญาณคลื่นพาห้	57
2.3.13 การมอดูเลตโดยการเปลี่ยนความถี่ของคลื่นพาห้	58
2.3.14 การดีมอดูเลตสัญญาณแบบเอ็นเวลโลปดีเทคชัน	60
2.3.15 การดีเทคสัญญาณไบนารีโดยวิธีเอ็นเวลโลปดีเทคชัน	61
2.3.16 การดีเทคสัญญาณ M-QAM และสัญญาณ M-PSK	64
2.3.17 พรอบบะบิลิตีของการผิดพลาดของสัญญาณ ในระบบ M-QAM และ ระบบ M-PSK	66
2.3.18 ความจุของช่องสัญญาณแบบต่อเนื่อง	69
2.4 ทฤษฎีเบื้องต้นของระบบการสื่อสารแบบสเปกตรัม	69
2.4.1 ความจุช่องสัญญาณ (Channel Capacity, C)	69
2.4.2 การเข้ารหัสช่องสัญญาณ (Channel Coding)	72

สารบัญ (ต่อ)

	หน้า
2.4.3 ระบบสื่อสารสเปกตรัมแผ่ (Spread Spectrum Communication Systems)	76
บทที่ 3 การสร้างและออกแบบจำลองของโปรแกรม GUI (Graphical User Interface)	84
3.1 กล่าวนำ	84
3.2 การออกแบบและสร้างโปรแกรมแสดงผลลัพท์ GUI โดยใช้ซอฟต์แวร์ MATLAB	84
3.3 โปรแกรมที่ใช้ในการสร้างหน้าแสดงผล GUI	96
3.4 ข้อมูลต่างๆ ที่ใช้ในการสร้างโปรแกรมแสดงผล GUI	96
บทที่ 4 ผลของการจำลองแบบ และคู่มือการใช้งาน	100
4.1 กล่าวนำ	100
4.2 วิธีการใช้งานแบบจำลองโปรแกรม GUI	100
บทที่ 5 สรุปผลการจำลองแบบและข้อเสนอแนะ	115
5.1 กล่าวนำ	115
5.2 ข้อเสนอแนะ	116
บรรณานุกรม	118
ภาคผนวก	120

สารบัญตาราง

ตารางที่		หน้า
1.7	ผังตารางเวลาการดำเนินงานวิจัย	5
ตารางที่ 1	แสดงตัวอย่างผลการคำนวณสมการที่ (40) ที่มีค่า k ต่าง ๆ กัน	20
ตารางที่ ก.1	แสดงข้อเสนอองานวิจัย	119



สารบัญรูปภาพ

ภาพที่	หน้า
รูปที่ 1.1 แผงผังแสดงการวิจัย	4
รูปที่ 2.1 สัญญาณพัลส์สี่เหลี่ยมรายคาบ	7
รูปที่ 2.2 กราฟของฟังก์ชัน sinc (x)	8
รูปที่ 2.3 สเปกตรัมของสัญญาณพัลส์สี่เหลี่ยมที่มี $\tau = T_0/4$	8
รูปที่ 2.4 สัญญาณพัลส์สี่เหลี่ยมลูกเดียว	9
รูปที่ 2.5 สเปกตรัมเชิงขนาดของสัญญาณพัลส์สี่เหลี่ยมลูกเดียว	11
รูปที่ 2.6 รูปสัญญาณอิมพัลส์เมื่อ $\varepsilon \rightarrow 0$	11
รูปที่ 2.7 ฟิเตอร์แบบความถี่ต่ำ	12
รูปที่ 2.8 ฟังก์ชันถ่ายโอนและผลตอบสนองอิมพัลส์ของวงจรฟิเตอร์	13
รูปที่ 2.9 ฟังก์ชันถ่ายโอนที่สร้างได้จริง	14
รูปที่ 2.10 สัญญาณรบกวนแบบความร้อน	15
รูปที่ 2.11 ตัวอย่างของ PDF พื้นที่ใต้เส้นโค้งตรงบริเวณ $a < X \leq b$	16
รูปที่ 2.12 ความน่าจะเป็นฟังก์ชันที่มีการกระจายแบบเกาส์เซียนหรือแบบนอร์มอล	17
รูปที่ 2.13 บล็อกไดอะแกรมที่ใช้วิเคราะห์ผลกระทบของสัญญาณรบกวนเกาส์เซียน	21
รูปที่ 2.14 บล็อกไดอะแกรมของการหา ทรานสเฟอร์ฟังก์ชันของเมทซ์ฟิเตอร์	21
รูปที่ 2.15 ลักษณะการเกิดจุดสูงสุดเมื่อใช้เมทซ์ฟิเตอร์รับสัญญาณ	22
รูปที่ 2.16 รูปร่างของสัญญาณพัลส์ผลตอบสนองอิมพัลส์ของเมทซ์ฟิเตอร์	25
รูปที่ 2.17 วงจรฟิเตอร์แบบอินทิเกรต	
และคัมพ์รูปร่างของสัญญาณขาเข้าและสัญญาณขาออก	26
รูปที่ 2.18 การป้อนสัญญาณแรนดัมสู่วงจรฟิเตอร์	27
รูปที่ 2.19 ตัวอย่างสเปกตรัมของสัญญาณแอนะล็อกที่มีความถี่สูงสุดต่ำกว่า W	28
รูปที่ 2.20 รูปร่างของสัญญาณเบสแบนด์และสัญญาณ AM	28
รูปที่ 2.21 สัญญาณ AM ที่มีมอดูเลชันอินเด็กซ์ต่ำกว่าและสูงกว่า 1	29
รูปที่ 2.22 สเปกตรัมของสัญญาณ AM	30
รูปที่ 2.23 ระบบแปลงความถี่ของสัญญาณที่ใช้วงจร	32
รูปที่ 2.24 ตัวอย่างวงจรเอ็นเวลโลปดีเทกชัน	33

สารบัญรูปภาพ (ต่อ)

ภาพที่	หน้า
รูปที่ 2.25 การรับสัญญาณ AM	
(ก) แบบเอ็นเวลโลปดีเทคชัน (ข) แบบซูเปอร์เฮเตอร์โรไดน์	34
รูปที่ 2.26 บล็อกไดอะแกรมของซิงโครนัสดีเทคชัน	35
รูปที่ 2.27 ขั้นตอนการสร้างสัญญาณ PCM	36
รูปที่ 2.28 ตัวอย่างการแควนไตซ์สัญญาณแบบเชิงเส้น	36
รูปที่ 2.29 การเกิดแควนไตเซชันนอยส์กรณีของการแควนไตซ์แบบเชิงเส้น	38
รูปที่ 2.30 SQR ของสัญญาณ PCM ที่มีขึ้นของการแควนไตซ์สม่ำเสมอ	40
รูปที่ 2.31 ตัวอย่างการผิดเพี้ยนเชิงขนาด	42
รูปที่ 2.32 การใช้ไอควอลไลเซอร์แก้ปัญหาการผิดเพี้ยนของสัญญาณ	43
รูปที่ 2.33 ความหนาแน่นสเปกตรัมพลังงานของพัลส์สี่เหลี่ยม	44
รูปที่ 2.34 ตัวอย่างสเปกตรัมของสัญญาณอนาล็อกที่มีความถี่สูงสุดต่ำกว่า W	44
รูปที่ 2.35 รูปร่างของสัญญาณเบสแบนด์และสัญญาณ AM	45
รูปที่ 2.36 บล็อกไดอะแกรมแสดงการสุ่มตัวอย่าง	
และการสร้างสัญญาณเดิมขึ้นมาแบบอุดมคติ	46
รูปที่ 2.37 สเปกตรัมของสัญญาณเบสแบนด์	
และสเปกตรัมของสัญญาณสุ่มตัวอย่างที่มีอัตราสุ่มต่าง ๆ กัน	47
รูปที่ 2.38 การซ้อนกันของสเปกตรัมเมื่อ $f_s < 2W$	47
รูปที่ 2.39 การเกิดสเปกตรัมปลอมจาก 5.5 kHz เป็น 2.5 kHz	
เมื่ออัตราการสุ่มเป็น 8 kHz	48
รูปที่ 2.40 ตัวอย่างของแพทเทินรูปดวงตาในกรณีของไลน์โค้ดดิ้งแบบโพลาร์ NRZ	49
รูปที่ 2.41 แพทเทินรูปดวงตากับความหมายของส่วนต่าง ๆ	49
รูปที่ 2.42 สัญญาณพัลส์ขาออกที่ไม่เกิดการรบกวนระหว่างสัญลักษณ์	50
รูปที่ 2.43 ฟูเรียร์สเปกตรัมของสัญญาณพัลส์ในรูปที่ 2.42	50
รูปที่ 2.44 ส่วนประกอบต่าง ๆ	
ในช่องสัญญาณที่มีผลต่อการตอบสนองความถี่ของช่องสัญญาณ	52
รูปที่ 2.45 บล็อกไดอะแกรมของการหาทรานสเฟอร์ฟังก์ชันของเมทซ์ฟิลเตอร์	52
รูปที่ 2.46 ลักษณะการเกิดจุดสูงสุดเมื่อใช้เมทซ์ฟิลเตอร์รับสัญญาณ	53

สารบัญรูปภาพ (ต่อ)

ภาพที่	หน้า
รูปที่ 2.47 รูปร่างของสัญญาณอนอ็อฟคีย์อิง (OOK)	54
รูปที่ 2.48 ความหนาแน่นสเปกตรัมของสัญญาณยูนิโพลาร์ NRZ และความหนาแน่นสเปกตรัมของสัญญาณ OOK	54
รูปที่ 2.49 บล็อกไดอะแกรมของการมอดูเลตแบบ QAM และการจัดตำแหน่งของสัญญาณ	55
รูปที่ 2.50 การจัดตำแหน่งของสัญญาณ 16 QAM และ 64 QAM	57
รูปที่ 2.51 รูปคลื่นของกรณีสัญญาณ PSK แบบ 2 ระดับ	57
รูปที่ 2.52 การจัดตำแหน่งของสัญญาณ 4-PSK และ 16-PSK	58
รูปที่ 2.53 การมอดูเลตสัญญาณดิจิทัลโดยการเปลี่ยนความถี่ของคลื่นพาห้	59
รูปที่ 2.54 ความหนาแน่นสเปกตรัมกำลังของสัญญาณ CPFSK	60
รูปที่ 2.55 PDF ของสัญญาณคลื่นพาห้ที่มีสัญญาณรบกวนผสมอยู่	61
รูปที่ 2.56 บล็อกไดอะแกรมของการทำเอ็นเวลโลปดีเทคชัน	62
รูปที่ 2.57 PDF ของ $p_y(y H_0)$ และ $p_y(y H_1)$ ในกรณีของเอ็นเวลโลปดีเทคชัน	62
รูปที่ 2.58 บล็อกไดอะแกรมของการดีเทคสัญญาณ FSK แบบเอ็นเวลโลปดีเทคชัน	63
รูปที่ 2.59 PDF ของสัญญาณ y_0 และ y_1	63
รูปที่ 2.60 พรอบบะบิลิตีของการผิดพลาดของสัญญาณไบนารีที่มีการดีเทคแบบต่าง ๆ	64
รูปที่ 2.61 เวกเตอร์ของสัญญาณและเวกเตอร์ของสัญญาณรบกวน	65
รูปที่ 2.62 การแบ่งอาณาเขตของสัญญาณในระบบ 16-QAM	65
รูปที่ 2.63 การแบ่งอาณาเขตของสัญญาณในระบบ 16-PSK	66
รูปที่ 2.64 พรอบบะบิลิตีของการผิดพลาดของสัญญาณ M-QAM	67
รูปที่ 2.65 พรอบบะบิลิตีของการผิดพลาดของสัญญาณ M-PSK	67
รูปที่ 2.1-1 แบบจำลองของช่องสัญญาณ	69
รูปที่ 2.1-2 ตัวอย่างช่องสัญญาณที่ส่งสัญญาณดิจิทัล "1" กับ "0"	70
รูปที่ 2.3-1 สัญญาณไซน์ซอซอด์ความถี่เดียว	75
รูปที่ 2.3-2 สเปกตรัมของสัญญาณไซน์ซอซอด์ความถี่เดียว (ก) ขนาด (ข) เฟส	76
รูปที่ 2.3-3 เฟสเซอร์ของสัญญาณไซน์ซอซอด์ความถี่เดียว	76
รูปที่ 2.3-4 เฟสเซอร์แบบความถี่สองด้านของสัญญาณไซน์ซอซอด์ความถี่เดียว	77

สารบัญรูปภาพ (ต่อ)

ภาพที่	หน้า
รูปที่ 2.3-5 สเปกตรัมแบบความถี่สองด้านของสัญญาณไซน์ซอซอต์ความถี่เดียว	77
รูปที่ 2.3-6 แบบจำลองระบบสื่อสารสเปกตรัมแผ่	78
รูปที่ 2.3-7 การคิมอดูเลตของสัญญาณสเปกตรัมแผ่แบบ DS	79
รูปที่ 2.3-8 รูประบบสเปกตรัมแผ่แบบ FH	81
รูปที่ 3.1 แผนผังสำหรับแบบจำลอง SRAPCES โดยใช้โปรแกรม GUI	85
รูปที่ 3.1 แบบจำลองหน้าต่างแสดงผล MENU	86
รูปที่ 3.2 แบบจำลองหน้าต่างแสดงผล DIGITAL TRANSMISSION	87
รูปที่ 3.3 แบบจำลองหน้าต่างแสดงผล CHAPTER 6	88
รูปที่ 3.4 แบบจำลองหน้าต่างแสดงผล Problem 6.03	89
รูปที่ 3.5 การสร้างแบบจำลองหน้าต่างแสดงผล Confirm Close Operation	90
รูปที่ 3.6 แบบจำลองหน้าต่างแสดงผล THEORY	91
รูปที่ 3.7 แบบจำลองหน้าต่างแสดงผล EXAMPLE	92
รูปที่ 3.8 รูปหน้าต่างแสดงผล คำอธิบายความหมายสัญลักษณ์	93
รูปที่ 3.9 รูปหน้าต่างแสดงผลการประยุกต์ใช้ M-file ใน matlab editor	94
รูปที่ 3.10 แบบจำลองหน้าต่างแสดงผล RUN PROGRAME	95
รูปที่ 4.1 การตั้งค่าหน้า Display Properties	101
รูปที่ 4.2 การเปิดโปรแกรม MATLAB 7.0	102
รูปที่ 4.3 การเปิดไดรว์ข้อมูลที่อยู่ใน Current Directory และ การเปิดหน้าต่างต่าง menu	103
รูปที่ 4.4 การเชื่อมโยงของหน้าต่างแสดงผล MENU	104
รูปที่ 4.5 การเชื่อมโยงของหน้าต่างแสดงผล DIGITAL TRANSMISSION	104
รูปที่ 4.6 การเชื่อมโยงของหน้าต่างแสดงผล CHAPTER 6	105
รูปที่ 4.7 การเชื่อมโยงของหน้าต่างแสดงผล Problem 6.03	106
รูปที่ 4.8 การเชื่อมโยงของหน้าต่างแสดงผล THEORY	107
รูปที่ 4.9 การเชื่อมโยงของหน้าต่างแสดงผล EXAMPLE	108
รูปที่ 4.10 การเชื่อมโยงของหน้าต่างแสดงผล คำอธิบายความหมายสัญลักษณ์	109
รูปที่ 4.11 การเชื่อมโยงของหน้าต่างแสดงผล การประยุกต์ใช้ M-file ใน matlab editor	110
รูปที่ 4.12 การเชื่อมโยงของหน้าต่างแสดงผล RUN PROGRAM	111

สารบัญรูปภาพ (ต่อ)

ภาพที่		หน้า
รูปที่ 4.13	การเชื่อมโยงของหน้าแสดงผล CLOSE	112
รูปที่ 4.14	การเชื่อมโยงของหน้าแสดงผล HELP	113



บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ในโลกยุคโลกาภิวัตน์ที่กำลังดำเนินอยู่มีการสื่อสารโทรคมนาคมนับว่ามีส่วนสำคัญอย่างยิ่งทั้งในแง่ที่เป็นตัวกระตุ้นให้เกิดกระแสโลกาภิวัตน์ และเป็นปัจจัยหลักที่จะทำให้เกิดการพัฒนาอย่างต่อเนื่องต่อไป ซึ่งถ้าเราลองพิจารณารูปแบบของข่าวสารที่ส่งกันอยู่จะเห็นได้ว่ามีหลัก ๆ อยู่ 4 รูปแบบด้วยกันคือ ตัวอักษร เสียง ภาพนิ่ง และภาพเคลื่อนไหว การพัฒนาการของเทคโนโลยีในด้านการสื่อสารในช่วง 150 ปีเศษที่ผ่านมาถือเป็นการพัฒนาระบบการสื่อสารให้มีความสามารถสูงนั่นก็คือระบบสื่อสารที่สามารถส่งผ่านข่าวสารได้อย่างรวดเร็ว และต้องมีความถูกต้องสูง ซึ่งในบางครั้งการติดต่อสื่อสารที่ต้องการให้เกิดประสิทธิภาพสูงนั้น อาจเกิดความผิดพลาดเกิดขึ้นจากการสื่อสารได้ ดังนั้นจึงมีการคิดค้น ตรวจสอบ และทำการแก้ไขความผิดพลาดที่เกิดขึ้น โดยอาจใช้ความน่าจะเป็นช่วยในการตรวจสอบได้

เมื่อทำการสรุปวิวัฒนาการของระบบไฟฟ้าสื่อสารที่ผ่านมา ตั้งแต่อดีตจนถึงปัจจุบันจะพบว่าเทคโนโลยีทางด้านไฟฟ้าสื่อสารมีการพัฒนาในลักษณะหลัก ๆ ดังนี้

1. มีการใช้ความถี่คลื่นพาห์ที่สูงขึ้นตามลำดับความถี่คลื่นพาห์สูงสุดที่ใช้อยู่ในปัจจุบัน คือความถี่ที่มีความยาวคลื่นประมาณ $1 \mu\text{m}$ ที่ใช้ในระบบสื่อสารเส้นใยแสง ผลที่เกิดขึ้นมากก็คือมีความถี่ที่ใช้ได้มากขึ้น และสามารถส่งข่าวสารได้มากขึ้นตามความถี่ที่สูงสุด
2. มีการเปลี่ยนแปลงจากเทคโนโลยีแอนะล็อกเป็นเทคโนโลยีทางด้านดิจิทัล การเปลี่ยนแปลงนี้เกิดขึ้นทั้งในระบบส่งสัญญาณ และในระบบชุมสายโทรศัพท์การเปลี่ยนแปลงนี้ได้ให้เทคโนโลยีทางด้านโทรคมนาคมและเทคโนโลยีทางด้านคอมพิวเตอร์ มีส่วนเกี่ยวข้องและพึ่งพาอาศัยกันมากยิ่งขึ้น และส่งผลให้การพัฒนาการระบบการสื่อสารโทรคมนาคมเป็นไปได้อย่างรวดเร็ว ยิ่งขึ้นการพัฒนาทั้งหลายที่เกิดขึ้นนั้นจะสอดคล้องกับความต้องการที่กล่าวในข้อ 2. คือทำให้รวดเร็วขึ้น มีความถูกต้องสูงและมีความเชื่อถือได้ของระบบสูงและพึงสังเกตหลักความจริงที่ระบบใหม่เกิดขึ้นนั้นจะต้องมีความสามารถเมื่อเทียบกับต้นทุน (Cost Performance) ดีขึ้นเมื่อเทียบกับระบบเดิมที่มีอยู่เสมอ

สำหรับการเรียนการศึกษาในภาควิชาวิศวกรรมไฟฟ้าแขนงวิชาวิศวกรรมโทรคมนาคมนั้นบางรายวิชามีความจำเป็นที่จะต้องเห็นรูปลักษณะจริงเพื่อเข้าใจในเนื้อหาบทเรียนนั้น เช่น สูตรที่ใช้ใน

สำหรับการเรียนการศึกษาในภาควิชาวิศวกรรมไฟฟ้า แขนงวิชาวิศวกรรมโทรคมนาคมนั้นบางรายวิชามีความจำเป็นที่จะต้องเห็นรูปลักษณะจริงเพื่อเข้าใจในเนื้อหาบทเรียนนั้น เช่น สูตรที่ใช้ในการคำนวณและเมื่อคำนวณออกมาแล้วนำไปแสดงในรูปของกราฟมีลักษณะอย่างไร ซึ่งบางครั้งเมื่อนักศึกษาได้เรียนรู้แต่ทางด้านทฤษฎีก็ไม่สามารถจินตนาการผลที่ได้ จึงทำให้ไม่เข้าใจเนื้อหาในบทเรียนเท่าที่ควร ด้วยเหตุนี้จึงได้มีการออกแบบและเขียนโปรแกรม SRAPCES โดยใช้โปรแกรม GUI นี้ขึ้นมาเพื่อให้ให้นักศึกษาที่เรียนอยู่ในภาควิชาวิศวกรรมไฟฟ้าแขนงวิชาวิศวกรรมโทรคมนาคมมีความรู้ความเข้าใจในเนื้อหาบทเรียนมากขึ้น

1.2 วัตถุประสงค์ของการวิจัย

เป็นการออกแบบจำลองของโปรแกรม GUI (Graphical User Interface) ของระบบโปรแกรมคอมพิวเตอร์เบื้องต้น โดยใช้โปรแกรม MATLAB เพื่อใช้ในงานระบบการวิจัยด้านวิศวกรรมการสื่อสารให้แสดงผลตามทฤษฎีดังต่อไปนี้

- 1) เพื่อเข้าใจหลักการทำงานขั้นพื้นฐานของระบบการสื่อสารที่ใช้คลื่นวิทยุหรือสัญญาณที่ส่งทางวิทยุ
- 2) เพื่อศึกษารูปแบบของสัญญาณต่าง ๆ ที่ใช้ในการส่งสัญญาณแอนะล็อก
- 3) เพื่อศึกษาการมอดูเลตสัญญาณแบบแอนะล็อก
- 4) เพื่อศึกษาเปลี่ยนสัญญาณแอนะล็อกเป็นสัญญาณดิจิทัล
- 5) เทคนิคการมอดูเลตและดีมอดูเลตสัญญาณดิจิทัลแบบเบสแบนด์ ที่ใช้ในการส่งข่าวสารแบบดิจิทัลผ่านช่องสัญญาณ AWGN
- 6) เพื่อศึกษาเทคนิคมอดูเลตไบนารีและดีมอดูเลตที่ได้ผลดีที่สุดของสัญญาณ
- 7) เพื่อศึกษาลักษณะของช่องสัญญาณความถี่จำกัด และความผิดพลาดในการดีมอดูเลต
- 8) เพื่อศึกษาการปรับแต่งช่องสัญญาณ และการมอดูเลตสัญญาณต่าง ๆ
- 9) เพื่อศึกษาให้เกิดความเข้าใจในเรื่อง ความจุช่องสัญญาณ, การเข้ารหัส และระบบสื่อสารสเปกตรัมแผ่
- 10) เพื่อศึกษาการทำงานของโปรแกรม MATLAB
- 11) เพื่อศึกษาสเปกตรัมของสัญญาณในลักษณะต่าง ๆ
- 12) เพื่อพัฒนาและประยุกต์โปรแกรมไปช่วยเป็นสื่อการเรียนการสอนและการวิจัย

1.3 ขอบเขตของการวิจัย

ทำการศึกษาโปรแกรม MATLAB เกี่ยวกับเรื่องระบบการสื่อสารแอนะล็อก ดิจิทัล และสเปกตรัม โดยอ้างอิงจากหนังสือ CONTEMPORARY COMMUNICATION SYSTEMS USING MATLAB เป็นหลัก ซึ่งในหนังสือมีตัวอย่างโปรแกรม MATLAB อยู่แล้ว และทำการออกแบบจำลองของโปรแกรม GUI เพื่อนำมาสร้างเป็นโปรแกรมแสดงผลในทางกราฟิก

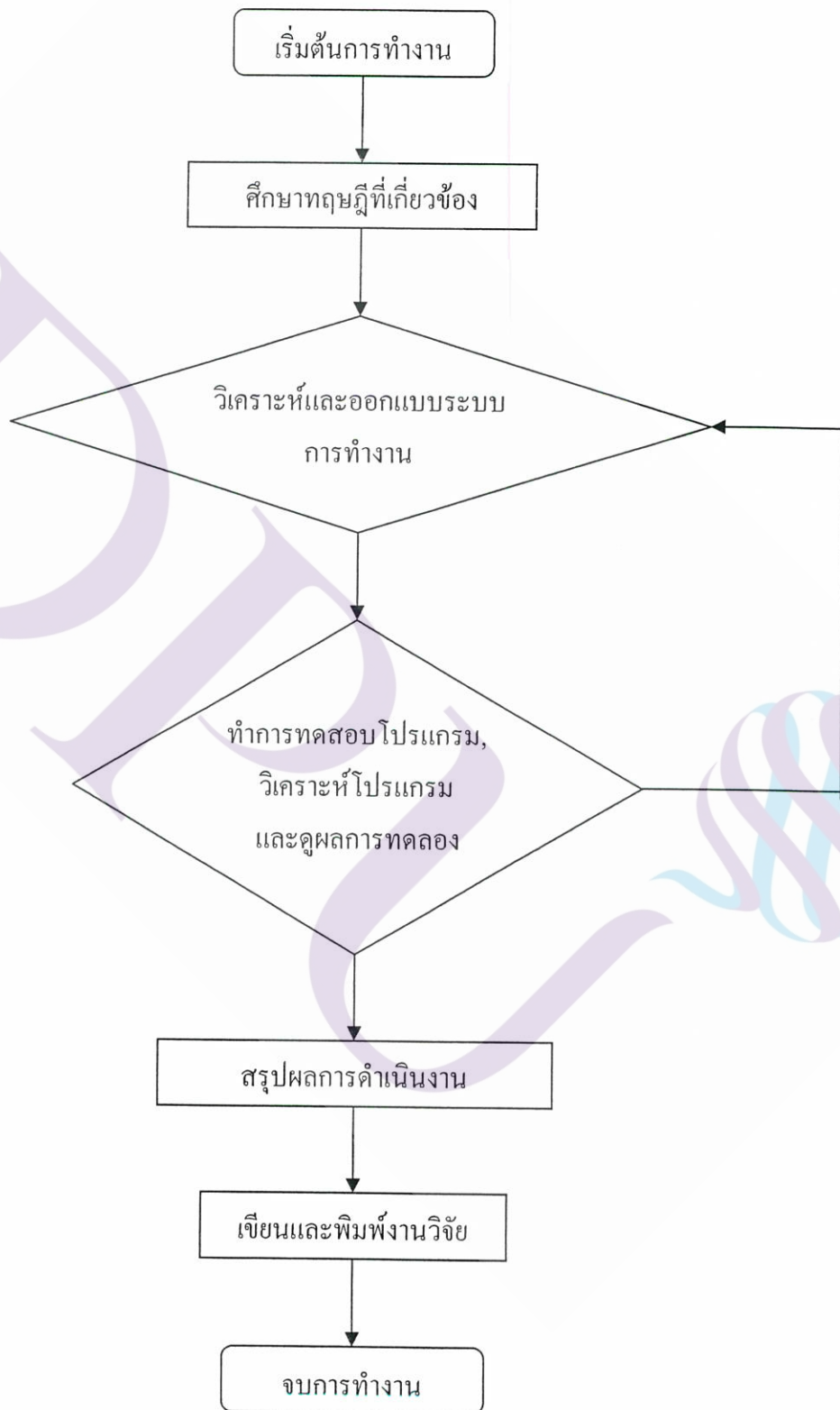
1.4 วิธีการดำเนินงานวิจัย

1. ศึกษาตัวอย่างโปรแกรม MATLAB เกี่ยวกับเรื่องระบบการสื่อสารแอนะล็อก, ดิจิทัล และสเปกตรัม โดยอ้างอิงหนังสือ CONTEMPORARY COMMUNICATION SYSTEM USING MATLAB
2. ศึกษาและเรียนรู้เกี่ยวกับเรื่องระบบการสื่อสารแอนะล็อก, ดิจิทัล และสเปกตรัม
3. วิเคราะห์คุณสมบัติของสัญญาณโดยใช้โปรแกรม MATLAB ช่วยในการวิเคราะห์แก้ปัญหา
4. ออกแบบจำลองระบบสัญญาณให้เป็นโปรแกรมแสดงผล ใช้โปรแกรม MATLAB ในการทำ GUI (Graphic User Interface)
5. ทำการทดสอบโปรแกรมแสดงผลที่ทำการออกแบบขึ้นมา
6. ทำการประเมินและสรุปผลการสร้างโปรแกรมแสดงผล
7. สรุปผลที่ได้จากผลการทดสอบ
8. เขียนและพิมพ์งานวิจัย

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1. ได้รับการเรียนรู้เกี่ยวกับระบบสื่อสารแอนะล็อก ดิจิทัล และสเปกตรัม
2. ได้รับทักษะในการเขียน โปรแกรม
3. ได้รับความรู้และเพิ่มทักษะในการใช้โปรแกรม MATLAB
4. โปรแกรมนี้ยังสามารถช่วยในการเรียนการสอน หรือเป็นแหล่งข้อมูลในการเรียนรู้ และค้นคว้าได้เป็นอย่างดี เพราะเป็นโปรแกรมที่ง่ายต่อการเรียนรู้
5. สามารถนำโปรแกรมที่ได้จากการวิจัยนี้ ไปใช้เป็นการเรียนการสอนตลอดจนการวิจัยด้านวิศวกรรมสื่อสารได้
6. เป็นพื้นฐานในการพัฒนาระบบการสื่อสารข้อมูล และการวิจัยในขั้นสูงต่อไป

1.6 แผนผังงานวิจัย



รูปที่ 1.1 แผนผังแสดงการวิจัย

1.7 ผังตารางเวลาการดำเนินงานวิจัย

ตารางที่ 1.1 ผังตารางเวลาการดำเนินโครงการวิจัย

ขั้นตอนการดำเนินการ	ม.ค. – ก.พ.	มี.ค. – เม.ย.	พ.ค. – มิ.ย.	ก.ค. - ส.ค.	ก.ย. – ต.ค.	พ.ย. – ธ.ค.
1. จัดรูปแบบโปรแกรมและทำการแก้ไขส่วนต่างๆ ที่ผิดพลาดเบื้องต้น	• •					
2. ศึกษาและเรียบเรียงเนื้อหาใหม่ในส่วนแอนะล็อก		• •	•			
3. ศึกษาและเรียบเรียงเนื้อหาใหม่ในส่วนดิจิทัล และสเปกตรัม			• •			
4. ตรวจสอบแก้ไขโปรแกรมและจัดทำรูปเล่มงานวิจัยฯ				• •		
5. รวบรวมการตรวจสอบงานวิจัย และแก้ไขข้อผิดพลาด					• •	• •

บทที่ 2

หลักการและทฤษฎีเบื้องต้นที่เกี่ยวข้อง

2.1 กล่าวนำ

การศึกษาในโครงการวิจัยนี้ ต้องอาศัยความรู้และความเข้าใจทางคณิตศาสตร์เป็นพื้นฐานอยู่บ้างเพื่อให้เข้าใจในเนื้อหาทฤษฎีของโครงการวิจัยนี้มากขึ้น ยกตัวอย่าง ความรู้ความเข้าใจเรื่องเมตริกซ์แบบต่าง ๆ การคำนวณหาค่าความน่าจะเป็น การใช้ลอการิทึม และควรมีความรู้ความเข้าใจในเรื่องสัญญาณดิจิทัล หลักการสื่อสารตลอดจนคุณลักษณะสัญญาณต่าง ๆ เช่น สัญญาณแบบ FSK (Frequency Shift Keying) เป็นต้น นอกจากนี้ยังควรมีพื้นฐานในการหาแบนด์วิดท์ (Bandwidth) หรือแถบความถี่ของสัญญาณความรู้เหล่านี้ถือว่าเป็นพื้นฐาน ในการทำโครงงานนี้อย่างมีประสิทธิภาพมากขึ้น

2.2 ทฤษฎีเบื้องต้นของระบบการสื่อสารแบบแอนะล็อก

2.2.1 อนุกรมฟูรีเยร์

ฟูรีเยร์ (Fourier) ซึ่งเป็นนักคณิตศาสตร์ชาวฝรั่งเศสได้สร้างรากฐานทางคณิตศาสตร์ที่สำคัญมากไว้อันหนึ่ง คือการกระจายรายคาบให้อยู่ในรูปของอนุกรมฟูรีเยร์ (Fourier Series Expansion) สำหรับฟังก์ชันที่ใช้เป็นฐานในการกระจายนั้นคือฟังก์ชันรูปไซน์ หรือเฟสของสัญญาณรูปไซน์ในที่นี้จะใช้ฐานในรูปฟังก์ชันเอกซ์โพเนนเชียลและเป็นค่าเชิงซ้อน เพราะเป็นรูปแบบในการคำนวณมากกว่า

ถ้าให้สัญญาณ $v(t)$ เป็นสัญญาณรายคาบที่มีเวลาเป็น T_0 หรือ $1/f$ สัญญาณ $v(t)$ นี้จะกระจายในรูปของอนุกรมฟูรีเยร์ดังนี้

$$v(t) = \sum_{n=-\infty}^{\infty} c_n e^{-j2\pi n f_0 t} \quad (1)$$

โดยที่ n เป็นเลขจำนวนเต็มลงตัว และ c_n เป็นค่าสัมประสิทธิ์ของเฟสเซอร์ หรือว่าเป็นสเปกตรัมลำดับ n ค่า c_n นี้จะหาได้โดยการคูณสมการ (1) ด้วย $e^{j2\pi n f_0 t}$ แล้วอินทิเกรตผลที่ได้ในในช่วงคาบเวลา T_0 ดังนี้

$$\begin{aligned} \int_{T_0} v(t) e^{-j2\pi n f_0 t} dt &= \sum_{m=-\infty}^{\infty} \int_{T_0} C_m e^{j2\pi m f_0 t} e^{-j2\pi n f_0 t} dt \\ &= T_0 c_n \end{aligned} \quad (2)$$

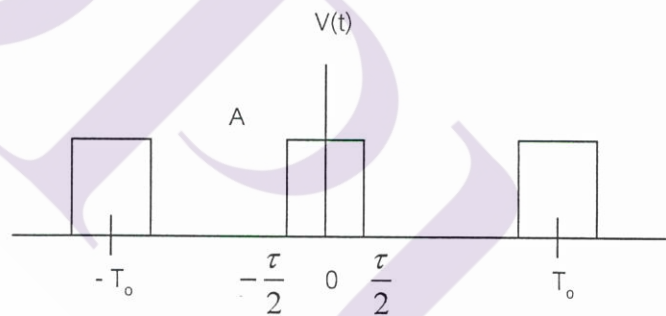
หรือ
$$c_n = \frac{1}{T_0} \int_{t_0} v(t) e^{-j\pi n f_0 t} dt \quad (3)$$

ค่าสัมประสิทธิ์ c_n นี้โดยทั่วไปจะเป็นค่าเชิงซ้อนซึ่งเขียนได้เป็น $c_n = |c_n| e^{j \arg c_n}$ ตัวอย่างการกระจายสัญญาณพัลส์สี่เหลี่ยมรายนานที่มีลักษณะดังแสดงในรูปที่ 2 กล่าวคือมีคาบเวลาเป็น T_0 มีความสูงพัลส์เป็น A และมีความกว้างของพัลส์เป็น τ จากสมการที่ (3) จะหาค่า c_n ได้ดังนี้

$$\begin{aligned} c_n &= \frac{1}{T_0} \int_{-T_0/2}^{T_0/2} v(t) e^{-j\pi n f_0 t} dt = \frac{1}{T_0} \int_{-\tau/2}^{\tau/2} A e^{-j\pi n f_0 t} dt \\ &= \frac{A}{-j2\pi n f_0 T_0} (e^{-j\pi n f_0 \tau} - e^{j\pi n f_0 \tau}) \\ &= \frac{A\tau}{T_0} \frac{\sin(\pi n f_0 \tau)}{\pi n f_0 \tau} \end{aligned} \quad (4)$$

และเนื่องจากรูปฟังก์ชัน $\sin(x)/x$ ที่ปรากฏในสมการที่ (4) จะพบบ่อยมากในเรื่องการสื่อสารจึงมีการนิยามฟังก์ชันดังนี้คือ

$$\text{sinc}(x) = \frac{\sin x}{x} \quad (5)$$



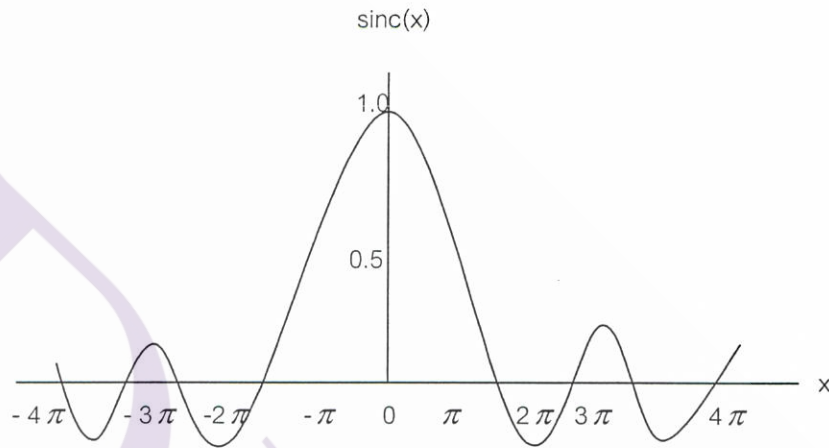
รูปที่ 2.1 สัญญาณพัลส์สี่เหลี่ยมรายนาน

เพื่อให้เขียนภาพได้ชัดเจนจะให้ $\tau = T_0 / 4$ ซึ่งเขียนรูปของสเปกตรัมได้ดังที่แสดงไว้ในรูปที่ 2.1 เส้นประที่เขียนไว้ในรูปที่ 2.1 ก็คือ $|\text{sinc}(\pi n \tau)|$ คูณด้วย $A\tau/T_0$ นั่นเอง

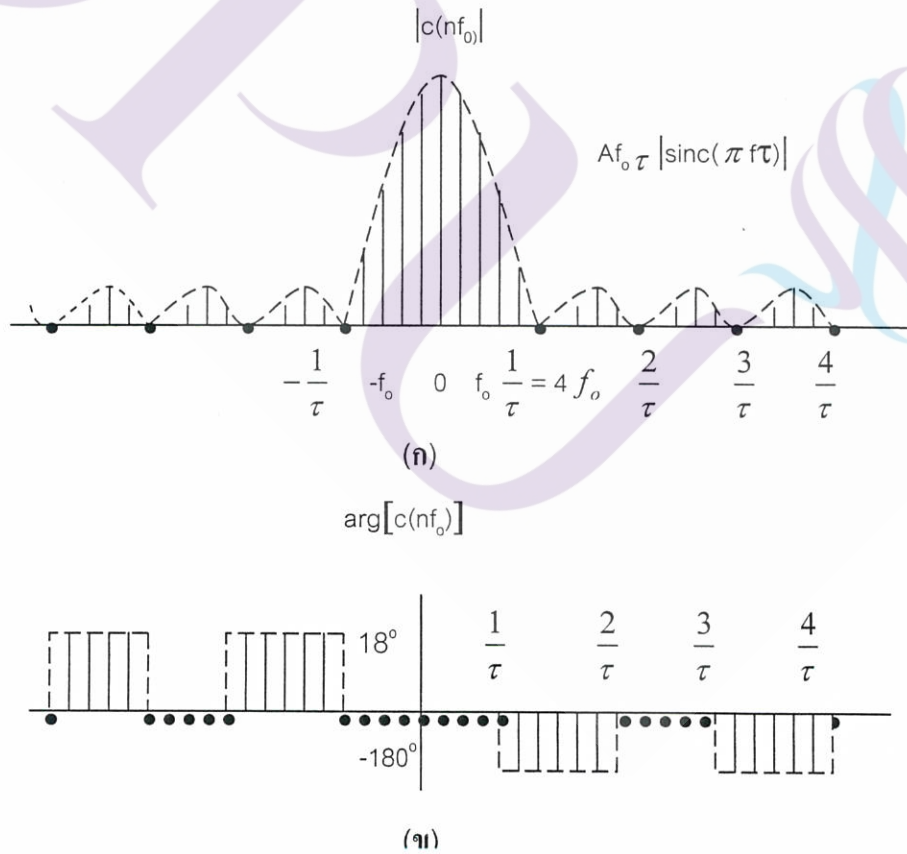
การกระจายอนุกรมฟูริเยร์ของสัญญาณรายนานตามคำอธิบายข้างต้น จะมีข้อสังเกตหลักๆ ดังนี้คือ

1. สเปกตรัมของสัญญาณที่กระจายออกมานี้ จะประกอบด้วยความถี่ f_0 เป็นความถี่พื้นฐาน (Fundamental Frequency) และมีความถี่ที่เป็นจำนวนเท่าลงตัวของ f_0 ซึ่งเรียกว่าความถี่ฮาร์โมนิกส์ของ f_0 ประกอบกันอยู่

2. c_0 คือค่าเฉลี่ยของ $v(t)$ เพราะ $c_0 = \frac{1}{T_0} \int_{t_0} v(t) dt$ ค่า c_0 ก็คือส่วนประกอบสเปกตรัมที่มีความถี่เป็น 0 หรือเป็นส่วนประกอบของสัญญาณกระแสตรงนั่นเอง
3. จากสมการที่ (3) จะเห็นได้ว่าถ้า $v(t)$ เป็นค่าจริง $c_{-n} = c_n^*$



รูปที่ 2.2 กราฟของฟังก์ชัน sinc (x)



รูปที่ 2.3 สเปกตรัมของสัญญาณพัลส์สี่เหลี่ยมที่มี $\tau = T_0/4$

2.2.2 การแปลงฟูรีเยร์

ดังที่ได้กล่าวไว้แล้ว สัญญาณที่ใช้ในการสื่อสารทั่วไปมักจะมีการเปลี่ยนแปลงตามเวลาที่ไม่เป็นรายคาบ ทั้งนี้เพราะสัญญาณที่เป็นรายคาบนั้นมีข่าวสารน้อย ดังนั้นการวิเคราะห์สัญญาณจึงจัดให้เป็นพื้นฐานของการวิเคราะห์ของการสื่อสารเริ่มต้น ซึ่งมีรายละเอียดดังนี้

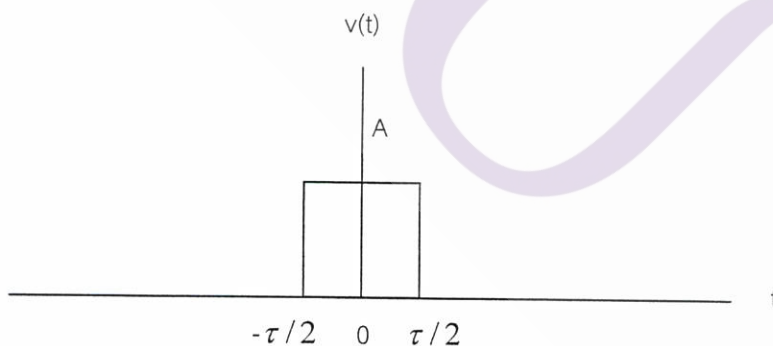
การแปลงฟูรีเยร์หรือฟูรีเยร์อินทิกรัลนั้นมีนิยามดังนี้

$$V(f) = F[v(f)] = \int_{-\infty}^{\infty} v(t)e^{-j2\pi ft} dt \quad (6)$$

เมื่อเปรียบเทียบสมการข้างบนกับสมการก่อนหน้านี จะเห็นได้ว่ามีส่วนคล้ายคลึงกันมาก จะต่างกันตรงที่ $V(t)$ เป็นการอินทิเกรตด้วยเวลา $-\infty$ ถึง ∞ และไม่มีการเฉลี่ยด้วยเวลา และ $V(f)$ แสดงส่วนประกอบสเปกตรัมของสัญญาณที่ความถี่ f ที่มีความถี่ต่อเนื่อง ในขณะที่ c_n เป็นส่วนประกอบสเปกตรัมของความถี่ nf_0 ที่มีความถี่ไม่ต่อเนื่อง เราอาจจะอธิบายความคล้ายคลึงว่า เปรียบเสมือนค่าคาบเวลา T_0 กว้างมากจนเข้าหาค่าอนันต์และขณะที่ T_0 เข้าหาค่าอนันต์นี้ความถี่พื้นฐาน f_0 ซึ่งเท่ากับ $1/T_0$ ก็จะมีค่าเล็กลงเข้าหาศูนย์ การแปลงฟูรีเยร์จะทำการแปลงสัญญาณจากอาณาจักรของเวลา (Time Domain) ไปอยู่ในรูปอาณาจักรของความถี่ (Frequency Domain) สำหรับการหาค่าของ $V(t)$ เดิมจาก $V(f)$ นั่นก็จะทำได้โดยการทำการแปลงฟูรีเยร์กลับ (Reverse Fourier Transform) ในรูป

$$V(f) = F[v(f)]^{-1} = \int_{-\infty}^{\infty} v(t)e^{-j2\pi ft} dt \quad (7)$$

ตัวอย่างการทำการแปลงฟูรีเยร์ของสัญญาณพัลส์สี่เหลี่ยมในรูปต่อไปนี้



รูปที่ 2.4 สัญญาณพัลส์สี่เหลี่ยมลูกเต๋า

$$\begin{aligned} V(f) &= \int_{-\infty}^{\infty} v(t)e^{-j2\pi ft} dt \\ &= \int_{-\tau/2}^{\tau/2} Ae^{-j2\pi ft} dt \end{aligned}$$

$$\begin{aligned}
&= \frac{A}{-j2\pi f} [e^{-j\pi f\tau} - e^{j\pi f\tau}] \\
&= A\tau \frac{\sin(\pi f\tau)}{\pi f\tau} \\
&= A\tau \operatorname{sinc}(\pi f\tau) \tag{8}
\end{aligned}$$

เมื่อนำสมการที่ (8) ไปเขียนกราฟก็จะได้ตามรูปที่ 2.4 ซึ่งจะเห็นได้ว่าสเปกตรัมจะมีค่าต่อตามค่า f และข้อสังเกตสำคัญประการหนึ่งคือ ถ้า τ มีขนาดเล็กกลง $v(f)$ จะกว้างออก ซึ่งสามารถอธิบายได้ทำนองเดียวกันกับที่กล่าวไว้แล้วคือการที่ τ มีขนาดเล็กกลงซึ่งแสดงว่าสัญญาณในอาณาจักรมีการเปลี่ยนแปลงอย่างรวดเร็ว จะทำให้สเปกตรัมในอาณาจักรของความถี่มีส่วนประกอบของความถี่สูงมากขึ้นในกรณีสุดโต่งที่ τ แคบมากจนเป็นอิมพัลส์ หรือเคลด้าฟังก์ชันเป็นกรณีที่สำคัญมาก จะขอล่าวเป็นรายละเอียดต่อไปนี้ ก่อนอื่นเคลด้าฟังก์ชัน $\sigma(t)$ ซึ่งเป็นฟังก์ชันที่นิยามขึ้นเพื่อความสะดวกในการวิเคราะห์ปัญหาทางคณิตศาสตร์ และทางฟิสิกส์นั้นมีนิยามดังนี้

$$\sigma(t) = 0 \quad t \neq 0$$

$$\text{และ} \quad \int_{-\infty}^{\infty} f(t)\sigma(t)dt = f(0) \tag{9}$$

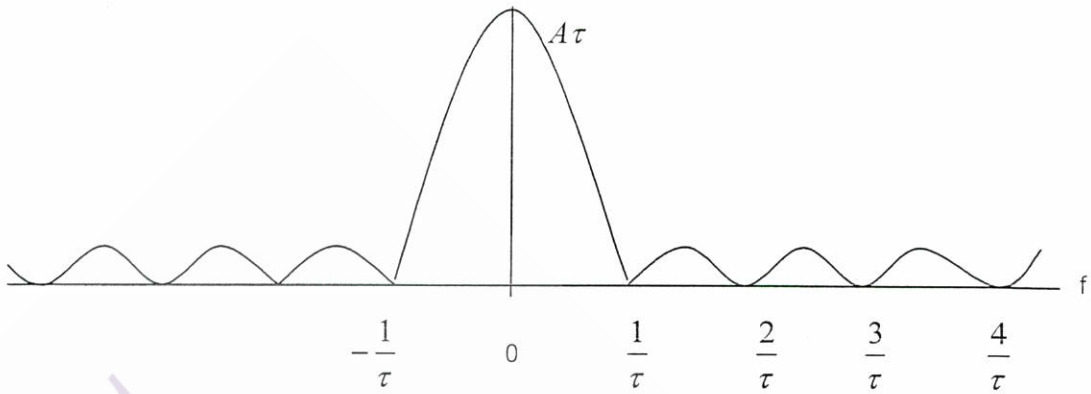
ในกรณีที่ $f(t) = 1$ จะได้

$$\int_{-\infty}^{\infty} \sigma(t)dt = \int_{-\infty/2}^{\infty/2} \sigma(t)dt = 1 \tag{10}$$

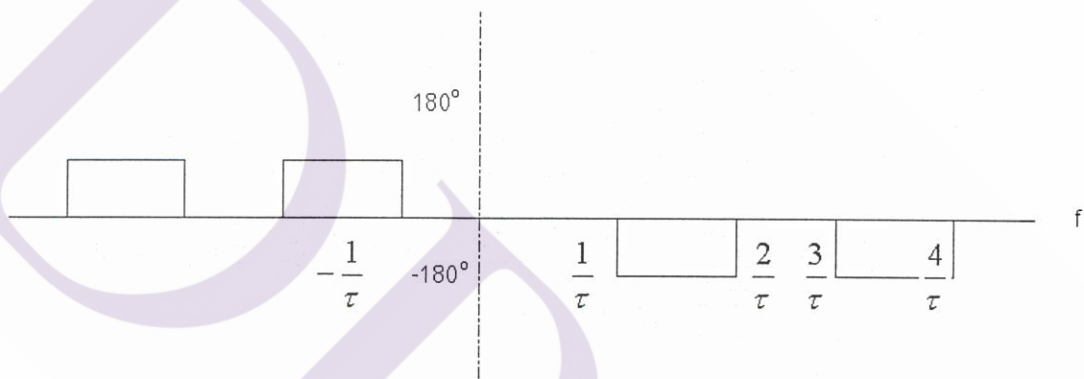
โดยที่ σ มีค่าที่เล็กมาก ๆ ตามคำนิยามและผลในสมการ (9) นี้จะกล่าวได้ว่าเคลด้าฟังก์ชันเป็นฟังก์ชันที่มีค่าเวลา

$t = 0$ และมีพื้นที่ใต้กราฟเป็น 1 ดังนั้นถ้าให้ $f(t)$ เป็นสัญญาณพัลส์สี่เหลี่ยมซึ่งมีช่วงกว้าง ε และสูง $1/\varepsilon$ หรือเป็นสัญญาณ sinc ที่มีความสูง $1/\varepsilon$ และจุดตัดแกนเป็น ε ดังรูปที่ 2.5 เคลด้าฟังก์ชันอาจเขียนได้ดังนี้

$$\sigma(t) = \lim_{\varepsilon \rightarrow 0} f(t) \tag{11}$$

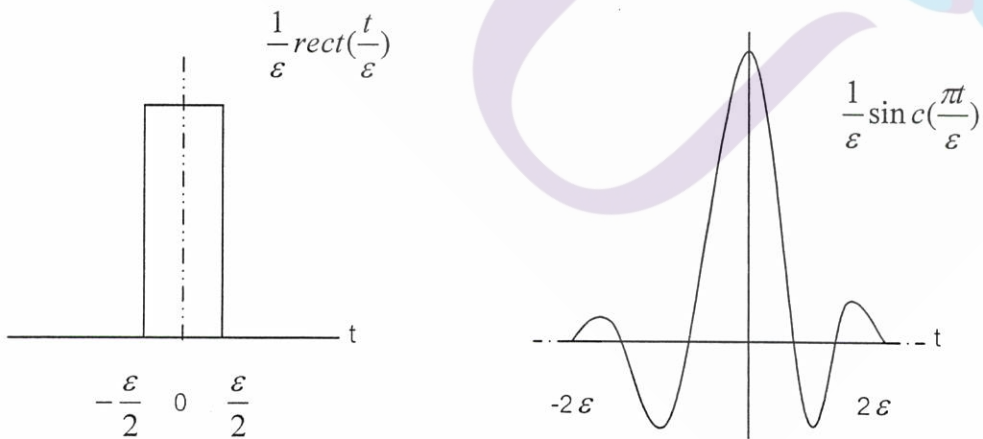


(ก) ขนาดของสเปกตรัม



(ข) เฟสของสเปกตรัม

รูปที่ 2.5 สเปกตรัมเชิงขนาดของสัญญาณพัลส์สี่เหลี่ยมลูกเดียว



รูปที่ 2.6 รูปสัญญาณอิมพัลส์เมื่อ $\epsilon \rightarrow 0$

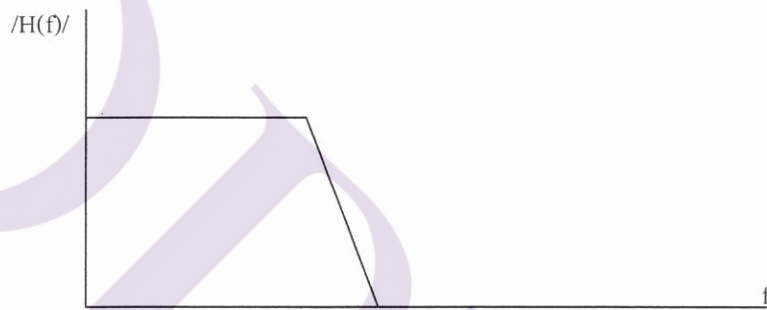
ดังนั้นเมื่อทำการแปลงฟูรีเยร์ของเดตาฟังก์ชันจะได้ผลดังนี้

$$D(f) = F[\sigma(t) = \int_{-\infty}^{\infty} \sigma(t)e^{-j2\pi ft} dt = 1 \quad (12)$$

ซึ่งหมายความว่าเดตาฟังก์ชันจะมีสเปกตรัมขนาดเท่ากันตลอดช่วงความถี่จาก $-\infty$ ถึง ∞ และสิ่งนี้เป็นเหตุผลที่สัญญาณที่มีลักษณะเป็นอิมพัลส์ เช่น สัญญาณรบกวนแบบสไปคีนอยส์ (Spike Noise) มีสเปกตรัมของสัญญาณรบกวนกว้าง

2.2.3 วงจรสมมูลผ่านต่ำของสัญญาณผ่านแถบ

คือวงจรไฟฟ้าที่ออกแบบให้สามารถส่งความถี่ในช่วงที่ต้องการ เรียกว่า ฟิลเตอร์ โดยฟิลเตอร์มีอยู่หลายแบบ เช่น



รูปที่ 2.7 ฟิลเตอร์แบบความถี่ต่ำ

ฟิลเตอร์แบบความถี่ต่ำ, แบบความถี่สูง, แบบผ่านแถบความถี่, และแบบตัดเฉพาะแถบความถี่, และแบบตัดเฉพาะความถี่เป็นต้นซึ่งแต่ละแบบจะมีคุณสมบัติ ในการส่งผ่านความถี่หรือผลตอบสนองอิม-พัลส์เชิง-ความถี่ การวิเคราะห์สเปกตรัมของสัญญาณที่ผ่านฟิลเตอร์ทางอุดมคตินั้น จะทำได้ดังนี้ ก่อนอื่น $H(f)$ จะเขียนได้ดังนี้

$$H(f) = \begin{cases} Ke^{-j2\pi ft} & f_l \leq f \leq f_u \\ 0 & f < f_l, f > f_u \end{cases} \quad (13)$$

โดยที่ f_l เป็นความถี่ส่งผ่านต่ำสุด และ f_u เป็นความถี่ส่งผ่านสูงสุดของฟิลเตอร์ และแบนด์วิดท์ของฟิลเตอร์จะเขียนได้ดังนี้

$$B = f_u - f_l \quad ; \quad \text{เมื่อ } B = \text{แบนด์วิดท์} \quad (14)$$

ถ้าให้สเปกตรัมขาเข้าเป็น $X(f)$ สเปกตรัมขาออกเขียนได้ดังนี้

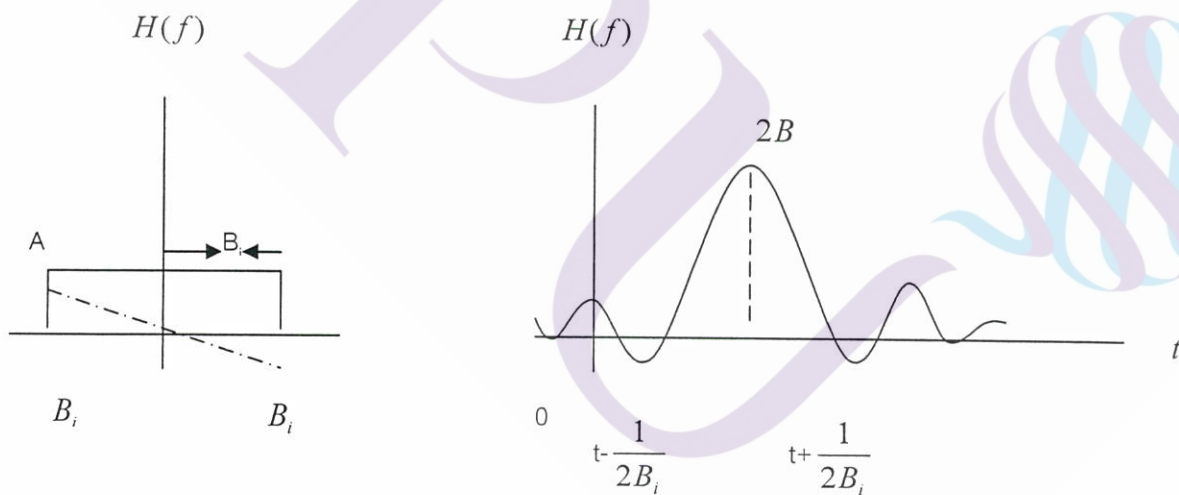
$$Y(f) = Ke^{-j2\pi f t_0} X(f) \quad (15)$$

กรณีที่สเปกตรัมของสัญญาณขาเข้ามีความถี่กว้างกว่าแบนด์วิดท์ของฟิลเตอร์ ฟิลเตอร์ก็จะตัดสัญญาณที่อยู่นอกแบนด์วิดท์ออก ทำให้สัญญาณขาออกมีรูปร่างแตกต่างออกไป การผิดเพี้ยนนี้ไม่ใช่ว่าจะไม่ดีเสมอไปเพราะว่าถึงแม้สัญญาณจริงจะผิดเพี้ยนไปเล็กน้อย แต่ถ้าเราสามารถสกัดสัญญาณรบกวนที่อยู่นอกแบนด์วิดท์ออกได้ก็จะเป็นผลดีมากกว่า เพราะสามารถตัดสัญญาณรบกวนออกไปได้ด้วย

ถ้าสัญญาณขาเข้าเป็นอินพัลส์ $\delta(t)$ ซึ่งทำให้ $X(f) = 1$ เมื่อทำการแปลงฟูริเยร์กลับสมการที่ (12) ผลที่ได้ก็จะเป็นผลตอบสนองอิมพัลส์ของวงจรฟิลเตอร์

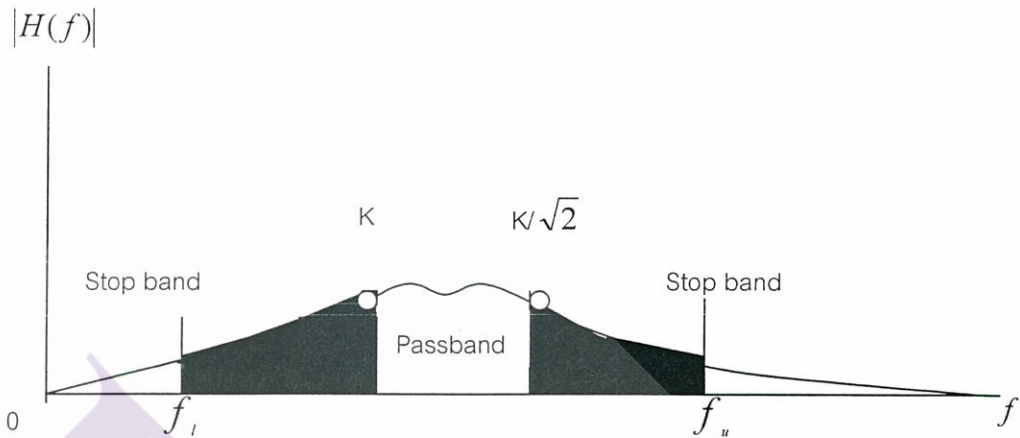
$$h(f) = F^{-1}[H(f) = 2B_i K \sin[2\pi B_i(t-t_0)]] \quad (16)$$

โดยที่ B_i คือ แบนด์วิดท์ของฟิลเตอร์ผ่านความถี่ต่ำแบบอุดมคติ



รูปที่ 2.8 ฟังก์ชันถ่ายโอนและผลตอบสนองอิมพัลส์ของวงจรฟิลเตอร์

จากรูปจะเห็นว่ารูปร่างของสัญญาณขาออกกว้างออกไปจากสัญญาณอิมพัลส์ และมีค่าสูงสุดเกิดขึ้นที่เวลา t_0 และสัญญาณขาออกจะเกิดขึ้นที่ก่อนเวลา $t=0$ คือ เกิดขึ้นก่อนสัญญาณขาเข้า ดังนั้นจะเห็นว่าไม่มีทางเป็นไปได้ ที่เราจะสร้างฟิลเตอร์ในลักษณะนี้ขึ้นมาได้ ฟิลเตอร์ที่สร้างได้จริงนั้นมักจะมีรูปร่างของทรานสเฟอ์ฟังก์ชันตามที่ได้แสดงดังนี้



รูปที่ 2.9 ฟังก์ชันถ่ายโอนที่สร้างได้จริง

ซึ่ง $|H(f)|$ จะมีค่าไม่สม่ำเสมอ ช่วงสัปดาห์ความถี่จะไม่ชัดเจน การนิยามแบนด์วิดท์ของฟิลเตอร์มักจะนิยามจากช่วงความถี่ที่ $|H(f)| = K/\sqrt{2}$ หรือความถี่ที่ f_l และ f_u ที่เรียกว่าคัตออฟ (cutoff -frequency) สำหรับช่วงความถี่ที่นิยามไม่แน่นอนส่วนใหญ่จะใช้ค่าที่ $|H(f)| < K/10$ สำหรับช่วงความถี่ที่อยู่ระหว่างช่วงส่งผ่านและช่วงสัปดาห์ความถี่นั้นเป็นช่วงความถี่เปลี่ยนแปลงช่วงความถี่นี้จะไม่ส่งผ่านเต็มที่ ฟิลเตอร์ที่ดีนี้จะต้องมีช่วงนี้แคบ เช่นพวงวงจร RC ฟิลเตอร์นี้มีคัตออฟเป็น $f_c = 1/2\pi RC$ และแบนด์วิดท์ $B = 1/2\pi RC$ เช่นเดียวกัน

$$H(f) = \frac{1}{G(jf/B)} \quad (17)$$

โดยที่ B เป็นแบนด์วิดท์ของระบบ และ $G(jf/B)$ เป็นโพลีโนเมียลเชิงซ้อนและมีคุณสมบัติดังนี้

$$|G(jf/B)|^2 = 1 + (f/B)^{2n} \quad (18)$$

ดังนั้นเราจะได้ว่า

$$|H(f)|^2 = \frac{1}{1 + (f/B)^{2n}} \quad (19)$$

2.2.4 ความหมายของกระบวนการเรณดัม

ในการพิจารณาเกี่ยวกับความน่าจะเป็นของเหตุการณ์ของการทดลองเชิงสถิติ ความน่าจะเป็นที่จะเกิดตัวแปรเรณดัมตัวหนึ่งนั้นมักจะมีค่าเปลี่ยนแปลงตามเวลาเสมอ ตัวแปรเรณดัมจะเป็นฟังก์ชันที่แปลงเหตุการณ์ทางสถิติให้เป็นตัวเลขค่าจริง และเซตที่เป็นไปได้ทั้งหมดนี้เรียกว่า Ensemble ตัวอย่าง สมการของการสื่อสารได้แก่เซตของฟังก์ชัน $X(t)$

$$X(t) = \sum_{n=-N}^N [a_n \cos(n\omega t) + b_n \sin(n\omega t)] \quad (20)$$

โดยที่ a_n , b_n เป็นตัวแปรเรณดัมที่อิสระจากกัน มีการกระจายแบบเกาส์เซียน และมีค่าเฉลี่ยเป็นศูนย์ ตัวแปรเรณดัมในเชิงสถิติไม่ได้หมายถึงตัวแปรที่เกิดขึ้นอย่างอิสระ โดยไม่มีกฎเกณฑ์แต่จะหมายถึงฟังก์ชันซึ่งสร้างค่าจริงออกจากเหตุการณ์ที่เกิดขึ้นได้ในการทดลอง ถ้าให้ α เป็นเหตุการณ์ที่เกิดขึ้นได้ในการทดลอง และ $X(\alpha)$ เป็นฟังก์ชันที่จะกำหนดค่าจริงจาก α ค่า $X(\alpha)$ นี้จะถูกเรียกว่าตัวแปรเรณดัม ตัวแปรเรณดัมจะมี 2 แบบ คือ แบบที่มีค่าไม่ต่อเนื่อง และแบบที่มีค่าต่อเนื่อง สำหรับกรณีตัวแปรเรณดัมที่มีค่าต่อเนื่องนั้น จะเกิดจากการคำนวณเหตุการณ์ที่เกิดขึ้นได้ในการทดลองที่ไม่จำกัดซึ่งจะส่งผลให้ $X(\alpha)$ มีค่าอย่างต่อเนื่อง ตัวอย่างที่ดีของตัวแปรเรณดัมประเภทนี้ก็คือสัญญาณรบกวน รูปที่ 2.10 แสดงตัวอย่างสัญญาณรบกวนแบบความร้อน สัญญาณรบกวนประเภทนี้จะมีค่าสูงสุดและค่าต่ำสุดอยู่ค่าหนึ่ง และค่าที่อยู่ระหว่างสองค่านี้ก็จะเกิดขึ้นได้ทั้งหมดอย่างต่อเนื่อง ดังตัวแปรเรณดัมของสัญญาณประเภทนี้ก็จะมีความต่อเนื่อง



รูปที่ 2.10 สัญญาณรบกวนแบบความร้อน

2.2.5 ฟังก์ชันความหนาแน่นความน่าจะเป็น (Probability Density Function, PDF)

ในกรณีที่ตัวแปรเรณดัมเป็นแบบมีค่าต่อเนื่องนั้น ในหลาย ๆ โอกาสเราจะพบว่า การให้นิยามความหนาแน่นของความน่าจะเป็นนั้นมีประโยชน์มาก นิยามของความหนาแน่นของความน่าจะเป็นซึ่งเรียกว่าฟังก์ชันความหนาแน่นความน่าจะเป็น (ย่อว่า PDF) เป็นดังนี้

$$P_x(x) \triangleq \frac{dF_x(x)}{dx} \quad (21)$$

ตามคำนิยามในสมการที่ (21) $P_x(x)$ ก็จะเป็นความน่าจะเป็นที่ X อยู่ระหว่าง $x-dx$ ถึง x จากสมการที่ (21) ถ้าเราต้องการหาฟังก์ชันแจกแจง ก็จะทำให้ได้โดยการอินทิเกรตกลับมาเป็น

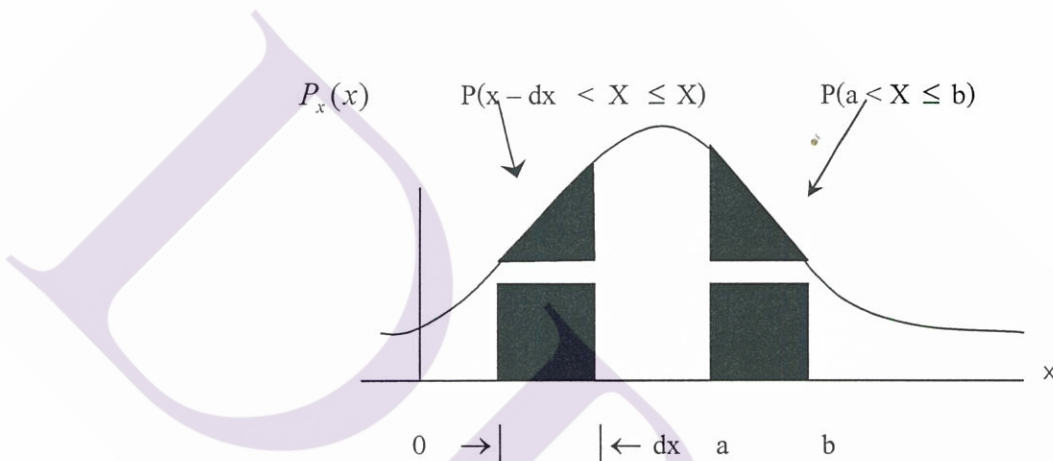
$$F_x(x) = \int_{-\infty}^x P_x(x) dx \quad (22)$$

คุณสมบัติของ PDF ที่เห็นได้ชัดเจนมีดังนี้คือ

$$P_x(x) \geq 0 \tag{23}$$

$$\int_{-\infty}^{\infty} P_x(x) dx = 1 \tag{24}$$

$$\begin{aligned} P(a < X \leq b) &= F_x(b) \\ &= \int_a^b P_x(x) dx \end{aligned} \tag{25}$$



รูปที่ 2.11 ตัวอย่างของ PDF พื้นที่ใต้เส้นโค้งตรงบริเวณ $a < X \leq b$ เป็นความน่าจะเป็นรวมในช่วงนั้นตามสมการที่ (25)

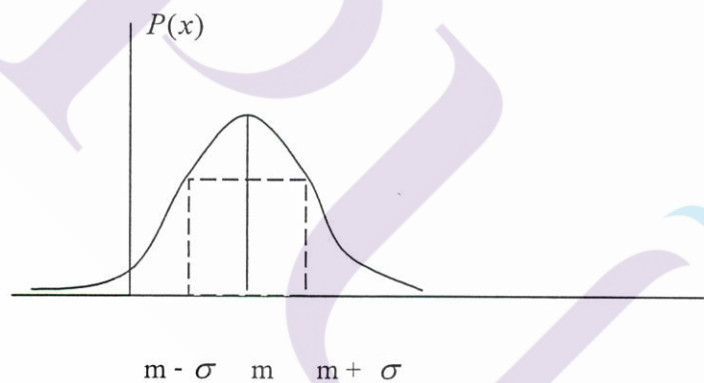
2.2.6 กระบวนการเกาส์และเกาส์มาร์คอฟ

ฟังก์ชันความหนาแน่นความน่าจะเป็นแบบเกาส์เซียนนี้บางครั้งเราจะเรียกว่า แบบนอร์มอล (Normal PDF) เนื่องจากจะเกิดขึ้นและใช้ได้กับเหตุการณ์ที่เกิดขึ้นตามธรรมชาติต่าง ๆ และในด้านการสื่อสารก็จัดว่ามีความสำคัญยิ่งยวด เพราะว่าสัญญาณรบกวนที่เกิดขึ้นในระบบสื่อสารมักจะมีส่วนเกี่ยวข้องกับการกระจายแบบเกาส์เซียนเสมอ ซึ่งอันที่จริงฟังก์ชันความหนาแน่นความน่าจะเป็นแบบเกาส์เซียนหรือแบบนอร์มอลมีรูปแบบที่เกิดขึ้นได้มากที่สุด ทั้งนี้เป็นเพราะทฤษฎีบททางสถิติที่เรียกว่า ทฤษฎีบทเซ็นทรัลลิมิต (Central limit theorem) สนับสนุนอยู่ ทฤษฎีบทนี้กล่าวไว้ว่า ถ้าเรามีตัวแปรแรนดัมที่เป็นอิสระจากกันอยู่ N ชุด และแต่ละชุดมีค่าเฉลี่ย และแวกเรียนซ์เป็น $m_1, m_2, m_3, \dots, m_N$ และ $\sigma_1^2, \sigma_2^2, \sigma_3^2, \dots, \sigma_N^2$ ตามลำดับผลบวกของตัวแปรแรนดัม N ชุดนี้ จะมีฟังก์ชันความหนาแน่นความน่าจะเป็นแบบการกระจายนอร์มอลในขณะที่ N มีค่าใหญ่ขึ้นทั้งนี้จะไม่ขึ้นอยู่กับ การกระจายของแต่ละตัวแปรแรนดัม และจะไม่ได้ค่าเฉลี่ยและแวกเรียนซ์ของความ

น่าจะเป็นฟังก์ชันความหนาแน่นความน่าจะเป็นแบบนอร์มอลนี้เป็น $m = \sum_{i=1}^N mi$ และ $\sigma^2 = \sum_{i=1}^N \sigma_i^2$ ตามลำดับเนื่องจากการพิสูจน์ทฤษฎีบทนี้เป็นเรื่องค่อนข้างยุ่งยากจึงไม่ขอกล่าว แต่ขอเน้นอีกครั้งหนึ่งว่าความน่าจะเป็นฟังก์ชันความหนาแน่นความน่าจะเป็นแบบนอร์มอลนี้เป็นรูปแบบที่พบมากที่สุด ไม่ว่าจะเป็นสายงานวิทยาศาสตร์หรือสายสังคมศาสตร์ ซึ่งตัวอย่างทางด้านวิทยาศาสตร์นั้นการวัดค่าทางวิทยาศาสตร์ต้องมีการผิดพลาดจากค่าจริงซึ่งการกระจายของข้อผิดพลาดรอบ ๆ ค่าจริงนั้นจะมีลักษณะการกระจายแบบนอร์มอล นอกจากนี้ในระบบไฟฟ้าสื่อสารสัญญาณรบกวนที่เกิดจากความร้อนก็มีลักษณะขนาดกระจายอยู่ในรูปของนอร์มอล เช่นเดียวกันฟังก์ชันความหนาแน่นความน่าจะเป็นฟังก์ชันแบบนอร์มอลนี้จะเขียนในรูปสูตรคณิตศาสตร์ได้ดังนี้

$$P(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(x-m)^2/2\sigma^2} \quad (26)$$

โดยที่ m และ σ^2 เป็นค่าเฉลี่ยและแวกเรียนซ์ตามลำดับ รูปที่ 2.15 แสดงรูปร่างของการกระจายแบบนอร์มอลลักษณะเฉพาะของการกระจายแบบนอร์มอล ก็จะสมมาตรกันทางซ้ายขวาของค่าเฉลี่ย นอกจากนี้ถ้าเราหาค่าเฉลี่ยและแวกเรียนซ์ของสมการ (26) จะได้



รูปที่ 2.12 ความน่าจะเป็นฟังก์ชันที่มีการกระจายแบบเกาส์เซียนหรือแบบนอร์มอล

$$E[x] = \int_{-\infty}^{\infty} \frac{x e^{-(x-m)^2/2\sigma^2}}{\sqrt{2\pi\sigma^2}} dx = m \quad (27)$$

$$E[x-m]^2 = \int_{-\infty}^{\infty} \frac{(x-m)^2 e^{-(x-m)^2/2\sigma^2}}{\sqrt{2\pi\sigma^2}} dx = \sigma^2 \quad (28)$$

และถ้าให้ $F(x)$ เป็นคิวมูเลตีฟดิสทริบิวชันฟังก์ชันของ $p(x)$ ก็จะเขียน $F(x)$ ได้ดังนี้

$$F(x) = P(x < x) = \int_{-\infty}^x \frac{e^{-(x-m)^2/2\sigma^2}}{\sqrt{2\pi\sigma^2}} dx \quad (29)$$

และเนื่องจากคุณสมบัติที่สมมาตรกันดังกล่าวข้างต้นจะได้

$$P(x < m) = P(x > m) = \frac{1}{2} \quad (30)$$

ค่า $F(x)$ ในสมการที่ (38) นั้นจะอินทิเกรตหาค่าไม่ได้ง่ายนักอย่างไรก็ตามเนื่องจากอินทิกรัลของ $F(x)$ นี้จะสัมพันธ์กันอย่างใกล้ชิดกับเออร์เรอร์ฟังก์ชัน (Error Function) ซึ่งค่าต่าง ๆ ของเออร์เรอร์ฟังก์ชันนั้นได้ถูกจำกัดไว้อย่างละเอียด ดังนั้นในการที่เราจะแสดงวิธีหาคความสัมพันธ์ระหว่าง $F(x)$ กับเออร์เรอร์ฟังก์ชันดังนี้ นิยามของเออร์เรอร์ฟังก์ชัน

$$\text{erf}(u) \triangleq \frac{2}{\sqrt{\pi}} \int_0^u e^{-t^2} dt \quad (31)$$

ซึ่งจะได้ $\text{erf}(0) = \text{erf}(\infty) = 1$ และเนื่องจากฟังก์ชันที่เป็น $1 - \text{erf}(u)$ ซึ่งเรียกว่าคอมพลิเมนต์ารีเออร์เรอร์ฟังก์ชัน (Complementary Error Function) ก็ถูกใช้บ่อยเช่นเดียวกัน จึงมีการนิยามฟังก์ชันดังกล่าวนี้โดยใช้สัญลักษณ์เป็น $\text{erfc}(u)$ ดังนี้

$$\text{erf}(u) \triangleq 1 - \text{erfc}(u) = \frac{2}{\sqrt{\pi}} \int_0^u e^{-t^2} dt \quad (32)$$

ในการหาความสัมพันธ์ระหว่าง $F(x)$ ของการกระจายแบบนอร์มอลกับเออร์เรอร์ฟังก์ชันนี้ เพื่อความสะดวกเราจะให้ค่าเฉลี่ยของการกระจายแบบนอร์มอล เป็นศูนย์คือ $m = 0$ ซึ่งจะทำให้ $F(x)$ เขียนได้ในรูปต่อไปนี้

$$\begin{aligned} F(x) &= \int_{-\infty}^x \frac{e^{-x^2/2\sigma^2}}{\sqrt{2\pi\sigma^2}} dx \\ &= \int_{-\infty}^0 \frac{e^{-x^2/2\sigma^2}}{\sqrt{2\pi\sigma^2}} dx - \int_x^{\infty} \frac{e^{-x^2/2\sigma^2}}{\sqrt{2\pi\sigma^2}} dx \end{aligned} \quad (33)$$

เนื่องจากเทอมแรกทางขวามือเท่ากับ 1 และเทอมที่สองจะแปลงเป็นรูปคอมพลิเมนต์ารีเออร์เรอร์ฟังก์ชันได้โดยการเปลี่ยนตัวแปรเป็น $t = x/\sqrt{2\sigma}$ ซึ่งจะทำให้สมการ (33) เขียนได้ดังนี้

$$\begin{aligned} F(x) &= 1 - \frac{1}{\sqrt{\pi}} \int_{x/\sqrt{2\sigma}}^{\infty} e^{-t^2} dt \\ &= 1 - \frac{1}{2} \text{erfc}\left(\frac{x}{\sqrt{2\sigma}}\right) \end{aligned} \quad (34)$$

สมการที่ (34) นี้ใช้ได้กับค่า x ที่เป็นบวกเท่านั้น เพราะ $\text{erfc}(u)$ นั้น นิยามไว้ในช่วงที่ u เป็นบวกเท่านั้น ดังนั้นในกรณีที่ $x < 0$ จะต้องมีการตัดแปลงเล็กน้อยดังนี้

$$\begin{aligned} F(x) &= F(-|x|) = \int_{-\infty}^{-|x|} \frac{e^{-x^2/2\sigma^2}}{\sqrt{2\pi\sigma^2}} dx \\ &= \frac{1}{\sqrt{\pi}} \int_{\infty}^{|x|/\sqrt{2}\sigma} e^{-t^2} dt \end{aligned} \quad (35)$$

และเมื่อแปลงตัวแปรให้ $u = -t$ จะได้

$$\begin{aligned} F(x) &= \frac{1}{2} \left(\frac{2}{\sqrt{\pi}} \int_{|x|/\sqrt{2}\sigma}^{\infty} e^{-u^2} du \right) \\ &= \frac{1}{2} \text{erfc} \left(\frac{|x|}{\sqrt{2}\sigma} \right) \end{aligned} \quad (36)$$

ความสัมพันธ์ของ $F(x)$ กับเออร์เรอร์ฟังก์ชันตามสมการที่ (32) และสมการที่ (34) นี้มีที่ใช้อย่างกว้างขวางในงานด้านวิทยาศาสตร์และในระบบสื่อสาร ยกตัวอย่างเช่น การวัดค่าในการทดลองทางวิทยาศาสตร์ที่กล่าวไว้ในตอนต้นว่าจะมีการผิดพลาดจากค่าจริงคือ m ในที่มีลักษณะการกระจายแบบนอร์มอล นั้นเราจะสามารถรู้ได้ว่าความน่าจะเป็นที่จะผิดพลาดมากกว่าค่า ๆ หนึ่ง เช่น $k\sigma$ มีค่ามากเท่าไรกล่าวคือ จากสมการที่ (34)

$$P(X < m - k\sigma) = \frac{1}{2} \text{erfc} \left(\frac{k}{\sqrt{2}} \right) \quad (37)$$

และจากสมการที่(33)

$$P(X > m + k\sigma) = 1 - F(k\sigma) = \frac{1}{2} \text{erfc} \left(\frac{k}{\sqrt{2}} \right) \quad (38)$$

เมื่อรวม $P(X < m - k\sigma)$ กับ $P(X > m + k\sigma)$ เข้าด้วยกันก็จะได้ความน่าจะเป็นรวมที่มีการผิดพลาดมากกว่า $k\sigma$ ซึ่งจะได้ผลดังนี้

$$P(x < m - k\sigma) + P(X > m + k\sigma) = \text{erfc} \left(\frac{k}{\sqrt{2}} \right) \quad (39)$$

ในกรณีเดียวกันนี้ เราจะสามารถหาความน่าจะเป็นที่ X ตกอยู่ในช่วง $m - k\sigma < X < m + k\sigma$ ได้โดยนำสมการ (31) ลบออกจากค่า $F(\infty)$ ซึ่งเท่ากับ 1 ผลที่ได้จะเป็นดังนี้

$$P(m - k\sigma < X < m + k\sigma) = 1 - \text{erfc} \left(\frac{k}{\sqrt{2}} \right) = \text{erf} \left(\frac{k}{\sqrt{2}} \right) \quad (40)$$

k	$P(m - k\sigma < X < m + k\sigma)$
0.5	0.383
1.0	0.683
1.5	0.866
2.0	0.955
2.5	0.988
3.0	0.997
3.5	0.9995
4.0	0.99994

ตารางที่ 1 แสดงตัวอย่างผลการคำนวณสมการที่ (40) ที่มีค่า k ต่าง ๆ กัน

จากตารางจะเห็นได้ว่า เมื่อ $k = 1$, $P(m - \sigma < X < m + \sigma) = 0.683$ และเมื่อ $k = 2$, $P(m - 2\sigma < X < m + 2\sigma) = 0.955$ นั้นหมายถึงความน่าจะเป็นที่ X จะตกออกนอกช่วงดังกล่าวจะเป็น 0.317 และ 0.045 ตามลำดับ

ในระบบสื่อสาร โดยเฉพาะการส่งสัญญาณดิจิทัลแบบเบสแบนด์นั้น ถึงแม้อุปกรณ์ต้นทางจะส่งสัญญาณที่มีระดับเป็น V_0 V และ 0 V ทำให้การตัดสินใจว่าสัญญาณที่รับมาเป็น “1” หรือเป็น “0” นั้นมีโอกาสผิดพลาด ความน่าจะเป็นที่จะผิดพลาดได้ก็จะเป็นไปตามที่ได้อธิบายมาข้างต้น

2.2.7 การแก้ไขปัญหาจากสัญญาณรบกวนแบบเกาส์เซียน

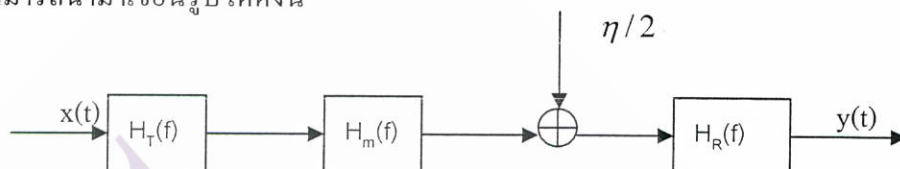
สัญญาณรบกวนที่เกิดขึ้นในระบบส่งสัญญาณดิจิทัลแบบเบสแบนด์นั้น นอกจากสัญญาณรบกวนแบบเกาส์เซียนแล้ว จะมีสัญญาณแบบสไปคี้ด้วย และเนื่องจากสัญญาณรบกวนแบบสไปคี้จะมีระดับสูงและจะทำให้เกิดการผิดพลาดขึ้นได้เสมอไม่ว่าจะใช้ไลน์โค딩แบบต่าง ๆ จึงมักจะพิจารณาเฉพาะผลกระทบจากสัญญาณรบกวนแบบเกาส์เซียนเป็นหลัก

ในการลดสัญญาณรบกวนแบบเกาส์เซียนนั้น โดยหลักการแล้วก็ต้องทำการลดแบนด์วิธของฟิลเตอร์ทางภาครับให้เหลือแคบที่สุด โดยที่ไม่ทำให้เกิดการรบกวนระหว่างสัญญาณขึ้น เมื่อพิจารณาเช่นนี้แล้วจะเห็นได้ว่า ช่องรับสัญญาณแบบเรสคโคไซน์ก็จัดว่าเป็นช่องสัญญาณที่มีคุณ

สมบัติที่เหมาะสมแบบหนึ่ง ในกรณีนี้ถ้าเราใช้รูปพิจารณาผลกระทบของสัญญาณรบกวนโดยรวม ทราบสเฟอ์ฟังก์ชันของวงจรถ้ารับเข้าด้วยกันหมดและเขียนเป็น $H_R(f)$ จากสมการ

$$H_c(f) = \frac{Y_r(f)}{X_s(f)} \quad (41)$$

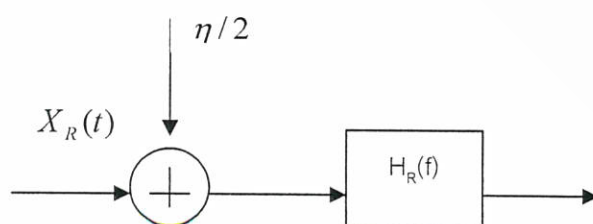
สามารถนำมาเขียนรูปได้ดังนี้



รูปที่ 2.13 บล็อกไดอะแกรมที่ใช้วิเคราะห์ผลกระทบของสัญญาณรบกวนเกาส์เซียน

$$H_R(f) = \frac{Y_r(f)}{X_s(f)H_T(f)H_m(f)} \quad (42)$$

ในสมการนี้ $X_s(f)$ เป็นฟูริเยร์สเปกตรัมของสัญญาณที่ส่งออกและ $H_m(f)$ เป็นทราบสเฟอ์ฟังก์ชันของสายนำสัญญาณ เมื่อฟังก์ชันทั้งสองนี้ก็จะสามารถกำหนด $H_R(f)$ ควบคู่กับ $H_T(f)$ เพื่อให้ช่องสัญญาณมีทราบสเฟอ์ฟังก์ชันแบบเรนต์โคไซน์ได้ เมื่อพิจารณาจากรูปที่ 2.10 จะมีข้อสังเกตประการหนึ่งคือ ถึงแม้ $H_T(f)$ และ $H_R(f)$ จะมีส่วนในการปรับแต่งรูปพัลส์ทั้งคู่ แต่ $H_R(f)$ เท่านั้นที่จะมีผลในการสกัดสัญญาณรบกวนที่เข้าสู่ภาครับ เพราะฉะนั้นแบนด์วิดท์สมมูลของสัญญาณรบกวนจะถูกกำหนดจาก $H_R(f)$ เท่านั้น การลดระดับของสัญญาณรบกวนจะสามารถทำได้โดยใช้ฟิลเตอร์ที่เรียกว่า แมทซ์ฟิลเตอร์ (Matched Filter) แมทซ์ฟิลเตอร์จะอาศัยหลักการที่ว่าวงจรถ้ารับจะรู้รูปร่างของสัญญาณล่วงหน้าถึงแม้จะไม่รู้ขนาดและเวลาที่สัญญาณจะผ่านเข้ามา เมื่อรูปร่างของสัญญาณล่วงหน้ามีโอกาที่จะสร้างฟิลเตอร์ที่มีทราบสเฟอ์ฟังก์ชันแมทซ์ กับรูปร่างของสัญญาณที่จะผ่านเข้ามานั้น เพื่อให้มีระดับของสัญญาณต่อสัญญาณรบกวนสูงสุด การหาทราบสเฟอ์ฟังก์ชันของแมทซ์พัลส์นั้นจะทำได้ตามขั้นตอนต่อไปนี้ ให้รูปร่างของสัญญาณพัลส์เป็น $X_c(t)$ และให้สัญญาณพัลส์นั้นเข้ามาถึงภาครับมีขนาดเป็น A_R และเวลาที่มาถึงเป็น t_0 สามารถเขียนได้ดังนี้



รูปที่ 2.14 บล็อกไดอะแกรมของการหาทราบสเฟอ์ฟังก์ชันของแมทซ์ฟิลเตอร์

$$X_R(t) = A_R X_c(t-t_0) \quad (43)$$

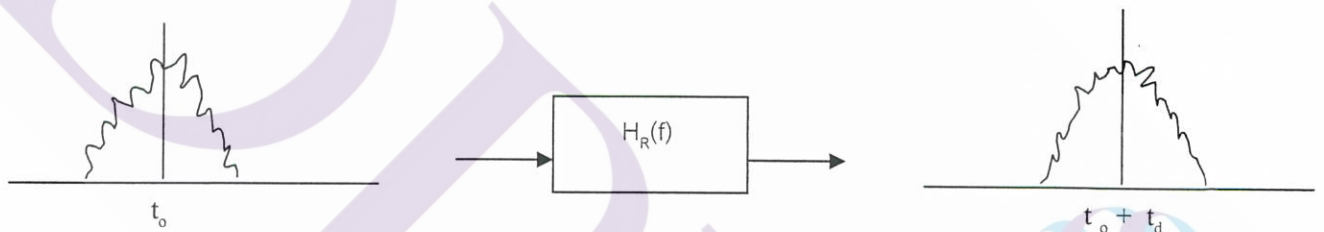
ถ้าให้ $x_c(f)$ เป็นการแปลงฟูรีเยร์ของ $X_c(t)$ เมื่อทำการแปลงฟูรีเยร์สมการที่ (43) จะได้

$$X_R(f) = A_R X_c(f) e^{-j\omega t_0} \quad (44)$$

และเราสามารถหาพลังงานของพัลส์สัญญาณได้ในรูปต่อไปนี้

$$E_R = \int_{-\infty}^{\infty} |X_R(f)|^2 df = A_R^2 \int_{-\infty}^{\infty} |X_c(f)|^2 df \quad (45)$$

ในขณะที่สัญญาณรบกวนแบบแอดคิทีฟเข้ามาพร้อมกับสัญญาณ $X_R(t)$ ฟิลเตอร์ทางภาครับจะต้องสามารถบีบพลังงานของพัลส์นั้นให้เกิดจุดที่สูงสุดขึ้นมา ในขณะที่เดียวกันนั้นก็จะต้องทำให้พลังงานของสัญญาณรบกวนที่ผ่านออกมา ได้มีขนาดต่ำที่สุด ถ้าให้ตำแหน่งที่เกิดจุดสูงสุดเป็น $t_0 + t_d$ ดังแสดงไว้ในรูปที่ 2.18



รูปที่ 2.15 ลักษณะการเกิดจุดสูงสุดเมื่อใช้เมทซ์ฟิลเตอร์รับสัญญาณ

ขนาดของสัญญาณที่จุดสูงสุดเป็น A เราสามารถหาขนาดสูงสุด A นี้โดยนำสเปกตรัมของสัญญาณขาออกมาทำการแปลงฟูรีเยร์กลับแล้วสังเกตที่เวลา $t_0 + t_d$ ถ้าให้ทรานสเฟอร์ฟังก์ชันของฟิลเตอร์เป็น $H_R(f)$ ดังรูปที่ 2.15 จะเขียนได้ดังนี้

$$\begin{aligned} A &= \int_{-\infty}^{\infty} H_R(f) X_R(f) e^{j\omega t} df \Big|_{t=t_0+t_d} \\ &= A_R \int_{-\infty}^{\infty} H_R(f) X_c(f) e^{j\omega t_d} df \end{aligned} \quad (46)$$

เมื่อเราพิจารณาในด้านสัญญาณรบกวนถ้าให้สัญญาณรบกวนนั้น มีขนาดกระจายอยู่ในรูปฟังก์ชันความหนาแน่นความน่าจะเป็นฟังก์ชันแบบเกาส์เซียน โดยจะมีค่าเฉลี่ยเป็นศูนย์ความหนาแน่นสเปกตรัมกำลังของสัญญาณรบกวนเป็น $G_n(f)$ กำลังของสัญญาณรบกวนที่ผ่านฟิลเตอร์ออกไป ซึ่งจะเท่ากับค่าแวลเรียนซ์ของสัญญาณรบกวนแบบเกาส์เซียนที่ผ่านออกไป ได้จะเขียนได้ในสรุปต่อไปนี้

$$\sigma^2 = \int_{-\infty}^{\infty} |H_R(f)|^2 G_n(f) df \quad (47)$$

และเนื่องจาก σ แสดงค่า rms ขนาดสัญญาณรบกวนแบบเกาส์เซียน สิ่งที่เราต้องการมากที่สุดก็คือ ทำให้อัตราส่วนระหว่างระดับสัญญาณ A กับค่า rms ของสัญญาณรบกวน σ สูงที่สุดเพื่อให้การตัดสินใจมีโอกาสผิดพลาด น้อยที่สุดจากสมการที่ (46) และสมการที่ (47) จะสามารถเขียนค่า A/σ ได้ในรูปต่อไปนี้

$$\left(\frac{A}{\sigma}\right)^2 = \frac{A_R^2 \left| \int_{-\infty}^{\infty} H_R(f) X_c(f) e^{j\omega_d t} df \right|^2}{\int_{-\infty}^{\infty} |H_R(f)|^2 G_n(f) df} \quad (48)$$

ในสมการที่ (48) นี้จะเห็นได้ว่ามีแต่ค่า $H_R(f)$ เท่านั้น ที่เราเลือกได้ในการเลือก $H_R(f)$ ในสมการที่ (48) ให้มีค่าสูงสุดนั้น จะทำได้โดยใช้กฎความไม่เท่ากันของชวอร์ ถ้าให้ $V(f)$ และ $W(f)$ เป็นฟังก์ชันใด ๆ ของ f จากกฎความไม่เท่ากันของชวอร์จะให้ความสัมพันธ์ดังนี้

$$\frac{\left| \int_{-\infty}^{\infty} V(f) W^*(f) df \right|^2}{\int_{-\infty}^{\infty} |V(f)|^2 df} \leq \int_{-\infty}^{\infty} |W(f)|^2 df \quad (49)$$

ข้างซ้ายมือของสมการที่ (49) นี้จะเหมือนกับข้างขวามือของสมการที่ (49) เมื่อให้

$$V(f) = H_R(f) \sqrt{G_n(f)} \quad (50)$$

$$\begin{aligned} W^*(f) &= A_R H_R(f) X_c(f) e^{j\omega_d t} / v(f) \\ &= A_R X_c(f) e^{j\omega_d t} / \sqrt{G_n(f)} \end{aligned} \quad (51)$$

และเนื่องจากทางซ้ายมือของสมการและข้างขวามือของสมการที่ (51) จะเท่ากันก็ต่อเมื่อ $V(f)$ และ $W(f)$ เป็นอัตราส่วนกันโดยตรง ดังนั้นถ้าให้ $V(f) = KW(f)/A_R$ โดยที่ K เป็นค่าคงที่ และแทนสมการ (50) ลงในสมการ (48) ก็จะได้ค่าสูงสุดของสมการ (47) ในรูปต่อไปนี้

$$\left(\frac{A}{\sigma}\right)_{\max}^2 = A_R^2 \int_{-\infty}^{\infty} \frac{|X_c(f)|^2}{G_n(f)} df \quad (52)$$

ในขณะเดียวกันก็จะได้ทรานสฟอร์มของฟิลเตอร์ที่เหมาะสมที่สุดในรูปต่อไปนี้

$$H_m(f) = \frac{K X_c^*(f) e^{-j\omega_d t}}{G_n(f)} \quad (53)$$

ฟิลเตอร์ที่มีทรานสฟอร์มฟังก์ชันตามสมการที่ (53) นี้จะถูกเรียกว่าเมทซ์ฟิลเตอร์ ถ้าเราพิจารณาสมการที่ (53) ให้ดีจะเห็นได้ว่า $|H_m(f)|$ จะแปรตาม $X_c(f)$ และแปรผกผันกับ $G_n(f)$ ซึ่งหมายความว่าสเปกตรัมของสัญญาณ $X_c(f)$ ส่วนที่มีค่าใหญ่จะถูกเน้นในขณะที่สเปกตรัมของ

$G_n(f)$ ที่มีค่าใหญ่จะถูกลด ซึ่งก็เป็นเรื่องที่สมเหตุสมผลมาก ในกรณีที่สัญญาณรบกวนเป็นแบบเกาส์เซียนซึ่ง $G_n(f) = \eta/2$ สมการ (41) ก็จะได้ในรูปต่อไปนี้

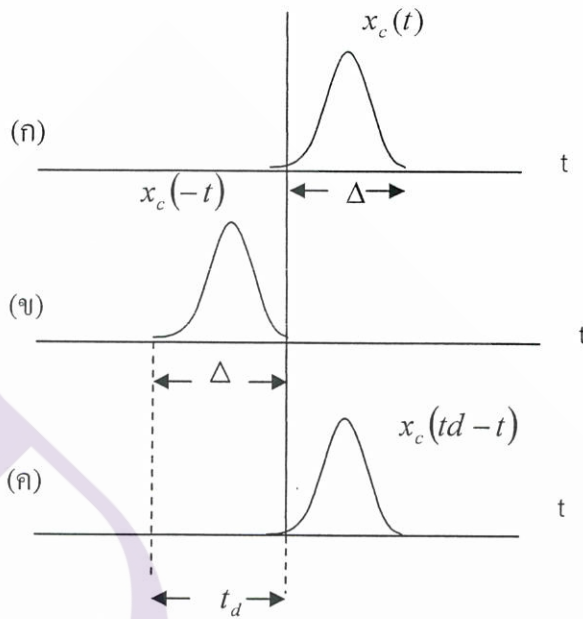
$$\left(\frac{A}{\sigma}\right)_{\max}^2 = \frac{2A_R^2}{\eta} \int_{-\infty}^{\infty} |X_c(f)|^2 df = \frac{2E_R}{\eta} \quad (54)$$

จากสมการที่ (54) จะเห็นได้ว่าพลังงานที่มีอยู่ในสัญญาณพัลส์ที่เข้ามานั้นจะมีผลโดยตรงต่อการทำให้ $(A/\sigma)_{\max}^2$ สูงขึ้นซึ่งก็จะเป็นผลดีต่อการตัดสินระดับสัญญาณมากขึ้น

ในการหาผลตอบสนองอิมพัลส์ของแมทซ์ฟิลเตอร์ในกรณีที่ $G_n(f) = \eta/2$ นั้นจะทำได้โดยการหาการแปลงฟูริเยร์กลับของสมการที่ (52) ซึ่งจะได้ผลดังนี้

$$\begin{aligned} H_M(t) &= \int_{-\infty}^{\infty} \frac{2K}{\eta} X_c^*(f) e^{-j\omega t_d} e^{j\omega t} df \\ &= \frac{2K}{\eta} \int_{-\infty}^{\infty} X_c^*(f) e^{j\omega(t-t_d)} df \\ &= \frac{2K}{\eta} \int_{-\infty}^{\infty} X_c(f) e^{j\omega(t_d-t)} df \\ &= \frac{2K}{\eta} X_c(t_d - t) \end{aligned} \quad (55)$$

$h_M(t)$ เป็นฟังก์ชันค่าจริงจึงมี $h_M^*(t) = h_M(t)$ จะเห็นได้ว่าผลตอบสนองอิมพัลส์ของแมทซ์ฟิลเตอร์จะมีรูปร่างเหมือนกับสัญญาณ ที่เข้ามาแต่จะมีเครื่องหมายของ t เป็น - ถ้าให้ $x_c(t)$ มีรูปร่างตามที่แสดงไว้ในรูปที่ 2.16 (ก) ถ้า $t_d = 0$ ผลตอบสนองอิมพัลส์จะเป็น $x_c(-t)$ ก็จะมีรูปร่างเป็นรูปที่ (ข) ผลตอบสนองอิมพัลส์ตามรูปที่ (ข) นี้จะไม่สามารถสร้างได้เพราะขัดแย้งกับกฎของความเป็นเหตุเป็นผล ผลตอบสนองอิมพัลส์ที่สร้างได้จริงนั้นจะเป็นดังที่แสดงไว้ในรูปที่ (ค) คือ t_d จะต้องใหญ่กว่าช่วงกว้างพื้นฐานของสัญญาณพัลส์ จากรูปร่างของผลตอบสนองอิมพัลส์ที่เป็นไปได้นี้ ถ้าเราทำการคำนวณสัญญาณขาออกของแมทซ์ฟิลเตอร์เฉพาะส่วนที่มาจากสัญญาณขาเข้า $A_R x_c(t)$ ก็จะได้ผลเป็นคอนโวลูชันของ $x_c(t)$ กับ $h_M(t)$ ในรูปต่อไปนี้

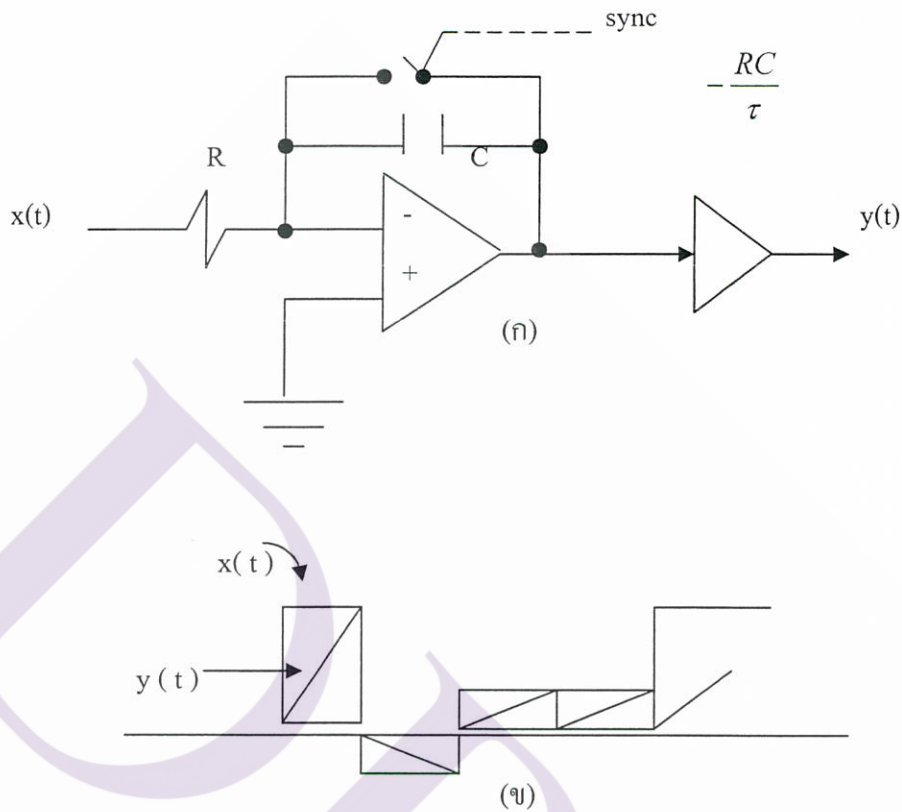


รูปที่ 2.16 รูปร่างของสัญญาณพัลส์ผลตอบสนองของอิมพัลส์ของแมทซ์ฟิลเตอร์

$$\begin{aligned}
 y(t) &= \int_{-\infty}^{\infty} A_R x_c(\tau) h_M(t-\tau) d\tau \\
 &= \frac{2KA_R}{\eta} \int_{-\infty}^{\infty} x_c(\tau) x_c(t_d - t + \tau) d\tau
 \end{aligned}
 \tag{56}$$

อินทิกรัลที่ปรากฏในสมการที่ (56) นี้จะอยู่ในรูปของการหาอโตคอร์รีเลชันของ $x_c(t)$ จากผลที่ได้นี้จะเห็นได้ว่า $y(t)$ จะมีค่าสูงสุดที่ตำแหน่ง $t = t_d$ จึงเป็นการยืนยัน สมมติฐานที่ว่า แมทซ์ฟิลเตอร์จะอัดพลังงานของสัญญาณให้เกิดค่าสูงสุดขึ้นที่เวลา t_d

การสร้างแมทซ์ฟิลเตอร์ให้มี ฟังก์ชันถ่ายโอนอยู่ในรูปสมการที่ (54) และมีผลตอบสนองอิมพัลส์ตามสมการที่ (54) นั้น จะทำได้โดยวงจรแพสซีฟ อย่างไรก็ตามถ้ารูปร่างของสัญญาณพัลส์ที่เข้ามามีความยุ่งยากมาก การสร้างก็จะยุ่งยากตามไปด้วย และที่สำคัญคือ ถ้าสร้างได้ไม่ดียังจะทำให้เกิดการรบกวนระหว่างสัญญาณได้มาก เพราะ $h_M(t)$ จะถูกร่วงออกไปกว่าความกว้างของสัญญาณพัลส์แต่ละสัญญาณ วิธีที่พยายามสร้างแมทซ์ฟิลเตอร์นี้มีโอกาสที่จะยุ่งยากกว่าการสร้างฟิลเตอร์ในช่องสัญญาณแบบเรสคโคไซน์ได้มากและไม่สามารถรับประกันได้ว่าจะไม่เกิดสัญญาณรบกวนระหว่างสัญญาณขึ้น กรณีที่สัญญาณขาเข้าเป็นสัญญาณรูปพัลส์สี่เหลี่ยม 2 ระดับ หรือหลายระดับ เราจะสามารถสร้างแมทซ์ฟิลเตอร์ได้โดยใช้วงจรแอดทีฟ วงจรนี้มีชื่อเรียกว่าวงจรแบบอินทิเกรตและดัมพ์ (Integrate and dump circuit)



รูปที่ 2.17 วงจรฟิลเตอร์แบบอินทิเกรตและคัมพัทธ์รูปร่างของสัญญาณขาเข้าและสัญญาณขาออก

2.2.8 การส่งสัญญาณเรณดัมผ่านฟิลเตอร์

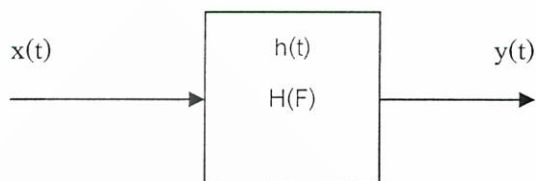
ถ้าเราผ่านสัญญาณเรณดัม $x(t)$ เข้าสู่วงจรฟิลเตอร์ที่มีผลตอบสนองอิมพัลส์เป็น $h(\tau)$ ดังที่แสดงไว้ในรูปที่ 2.18 ในกรณีนี้ที่สัญญาณ $x(t)$ เป็นแบบสเตรนาร์และมื่อโตคอร์รี่เรชันฟังก์ชันและความหนาแน่นสเปกตรัมกำลังเป็น $R_x(\tau)$ และ $G_x(f)$ ตามลำดับ เราสามารถหาสัญญาณขาออก $y(t)$ ได้ดังนี้

$$y(t) = h(t) * x(t) \tag{57}$$

ในช่วงที่ $|t| < T/2$ เมื่อใช้ $x_T(t)$ ตามนิยามในสมการ

$$x(t) = \begin{cases} x(t) & |t| < T/2 \\ 0 & |t| > T/2 \end{cases} \tag{58}$$

จะได้ $y_T(t)$ ในรูปต่อไปนี้



รูปที่ 2.18 การป้อนสัญญาณเรณคัมตู้วงจรฟลเตอร์

$$y_T(t) = h(t) * x_T(t) \quad (59)$$

เมื่อทำการแปลงฟูรีเยร์สมการ 58 นี้จะได

$$Y_T(f) = H(f)X_T(f) \quad (60)$$

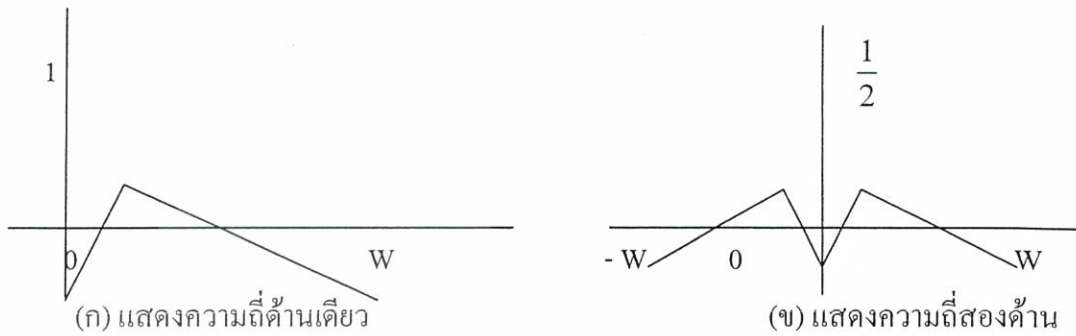
และจากนิยามความหนาแน่นสเปกตรัมกำลังตามสมการจะเขียน $G_Y(f)$ ได้ในรูปต่อไปนี้

$$\begin{aligned} G_Y(f) &= \lim_{T \rightarrow \infty} \frac{1}{T} E[|Y(t)|^2] \\ &= \lim_{T \rightarrow \infty} \frac{|H(f)|^2}{T} E[|X_T(f)|^2] \\ &= |H(f)|^2 G_X(f) \end{aligned} \quad (61)$$

ผลที่ได้ตามสมการที่ 60 นี้เป็นการบ่งบอกว่าความหนาแน่นของสเปกตรัมกำลังของสัญญาณขาออกจะขึ้นอยู่กับขนาดของ $H(f)$ เท่านั้น โดยไม่ขึ้นอยู่กับคุณสมบัติทางด้านเฟสของ $H(f)$ ในกรณีที่สัญญาณขาเข้ามี $G_X(f)$ ค่าคงที่ เช่น สัญญาณ ไวท์นอยส์ สัญญาณขาออกก็จะแสดงคุณสมบัติเชิงขนาดของ $H(f)$ นั้นหมายความว่าถ้าเราอยากรู้คุณสมบัติเชิงขนาดของฟิลเตอร์เราก็ทำได้ โดยทำการป้อนสัญญาณไวท์นอยส์เข้าสู่ฟิลเตอร์แล้วตรวจสอบสเปกตรัมของสัญญาณขาออกก็จะได้ $|H(f)|^2$ ออกมา การตรวจสอบความหนาแน่นสเปกตรัมอาจทำได้โดยใช้เครื่องวิเคราะห์สเปกตรัมหรือเครื่องวัดแรงดันที่มีฟิลเตอร์ผ่านแถบความถี่แคบ ๆ อยู่สองด้านหน้า

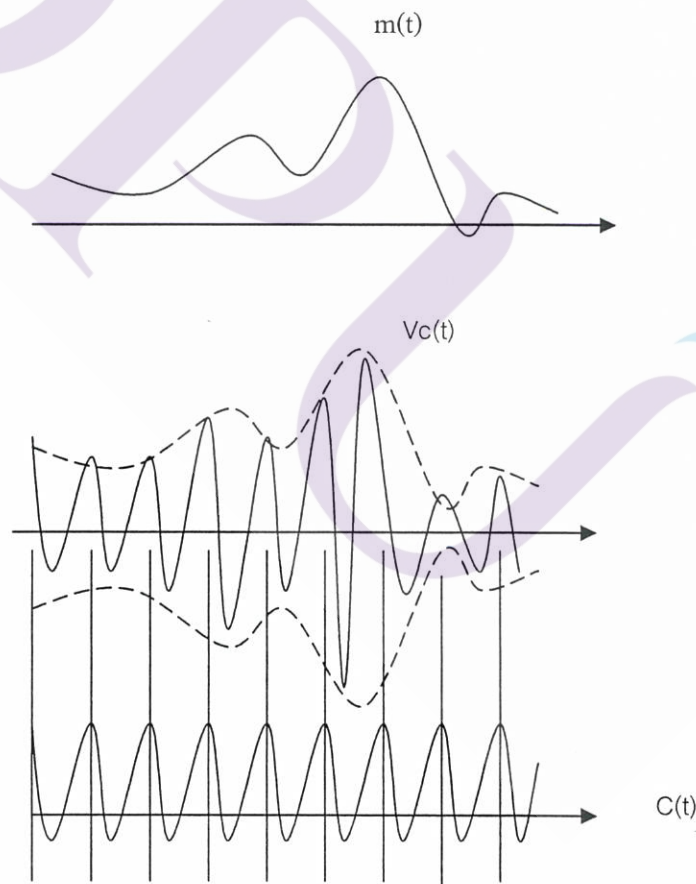
2.2.9 การมอดูเลตสัญญาณแอนะล็อก

โดยทั่วไปสัญญาณแอนะล็อกเบสแบนด์ที่ส่งกันนั้นจะมีความถี่สูงสุด และมีค่าที่จำกัด เช่น สัญญาณอডিโอที่หูคนเราฟังได้อาจจะสูงเพียง 15 kHz และเสียงที่คนเราพูดกันมีความถี่ไม่เกิน 4 kHz เป็นต้น เมื่อเป็นเช่นนี้ถ้าเราให้ความถี่สูงสุดของสัญญาณเบสแบนด์เป็น W Hz เมื่อเขียนสเปกตรัมของสัญญาณนั้นแบบแสดงความถี่ด้านบวกด้านเดียว และแบบแสดงความถี่ทั้งสองด้านจะเป็นดังที่แสดงดังรูปที่ 19 (ก) และ (ข)



รูปที่ 2.19 ตัวอย่างสเปกตรัมของสัญญาณแอนะล็อกที่มีความถี่สูงสุดต่ำกว่า W

สเปกตรัมทางด้านบวก และด้านลบจะสมมาตรกัน และมีขนาดเป็น $1/2$ ของสเปกตรัมที่แสดง ความถี่บวกด้านเดียว การมอดูเลตแบบ AM ที่ใช้ในวิทยุ AM โดยทั่วไปนั้นจะเป็นการสร้าง สัญญาณที่มีลักษณะเหมือนกับที่แสดงไว้ในรูปที่ 2.20 กล่าวคือเหมือนกับนำสัญญาณเบสแบนด์ไปขึ้น บนสัญญาณคลื่นพาห้



รูปที่ 2.20 รูปร่างของสัญญาณเบสแบนด์และสัญญาณ AM

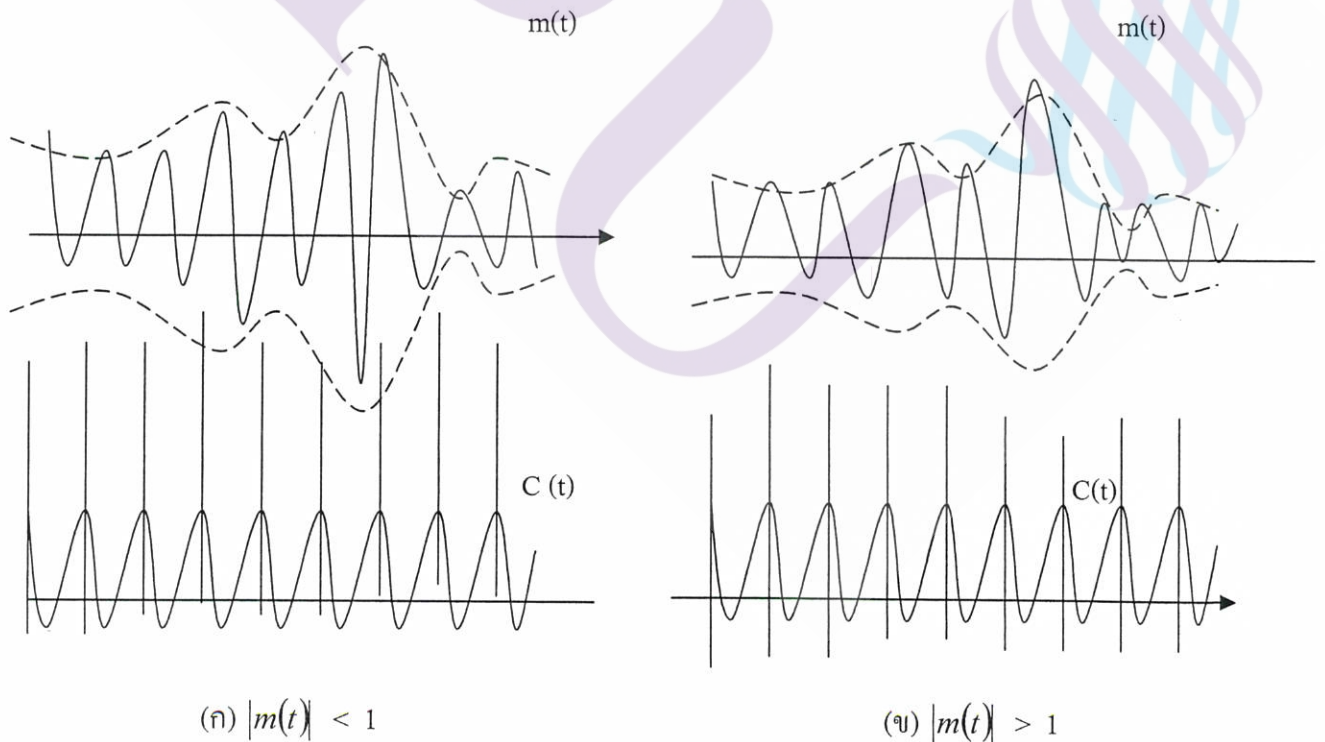
ทำให้ยอดคลื่นพาห์หรือเอ็นเวลโลป (Envelope) เปลี่ยนไปตามสัญญาณเบสแบนด์นั้น เราสามารถตรวจวัดสัญญาณเบสแบนด์กลับได้โดยใช้วงจรดีเทคเตอร์แบบง่าย ๆ ซึ่งเป็นเหตุผลสำคัญที่ทำให้วิทยุ AM ถูกนำมาใช้งานก่อน การตรวจวัดยอดคลื่นของคลื่นพาห์นี้ เราเรียกว่าเอ็นเวลโลปดีเทคชัน (Envelope Detection) สัญญาณ AM ที่มีรูปร่างดังที่แสดงไว้ในรูปที่ 2.20 นี้เมื่อเขียนเป็นฟังก์ชันของเวลาจะเป็นดังนี้

$$V_c(t) = \{A_c + V(t)\} \text{COS}\omega_c t \tag{62}$$

โดย $A_c \text{COS}\omega_c t$ เป็นสัญญาณคลื่นพาห์ และ $V(t)$ เป็นสัญญาณเบสแบนด์ โดยทั่วไปเราจะแสดงสัญญาณเดียวกันนี้ในรูปต่อไปนี้

$$V_c(t) = \{A_c + m(t)\} \text{COS}\omega_c t \tag{63}$$

โดยที่ $m(t)$ เป็นสัญญาณเบสแบนด์ที่ถูกนอร์มอลไลซ์ไว้ด้วยค่า A_c และ $m(t)$ จะถูกเรียกว่า มอดูเลชันอินเด็กซ์ มอดูเลชันอินเด็กซ์มักจะหมายถึงค่า $m(t)$ ที่มีขนาดสูงสุด และสัญญาณ AM ที่สามารถหาค่าด้วยวิธีเอ็นเวลโลปดีเทคชันจะต้องมี $|m(t)| < 1$ เสมอ ถ้า $m(t)$ มีค่าสูงกว่า 1 ก็จะทำให้เกิดการกลับเฟสตรงที่ $m(t) = -1$ จึงดีเทคได้เอ็นเวลโลปทางด้านบวกเสมอ ซึ่งจะทำให้ไม่สามารถดีเทคสัญญาณเบสแบนด์เดิมกลับมา



(ก) $|m(t)| < 1$

(ข) $|m(t)| > 1$

รูปที่ 2.21 สัญญาณ AM ที่มีมอดูเลชันอินเด็กซ์ต่ำกว่าและสูงกว่า 1

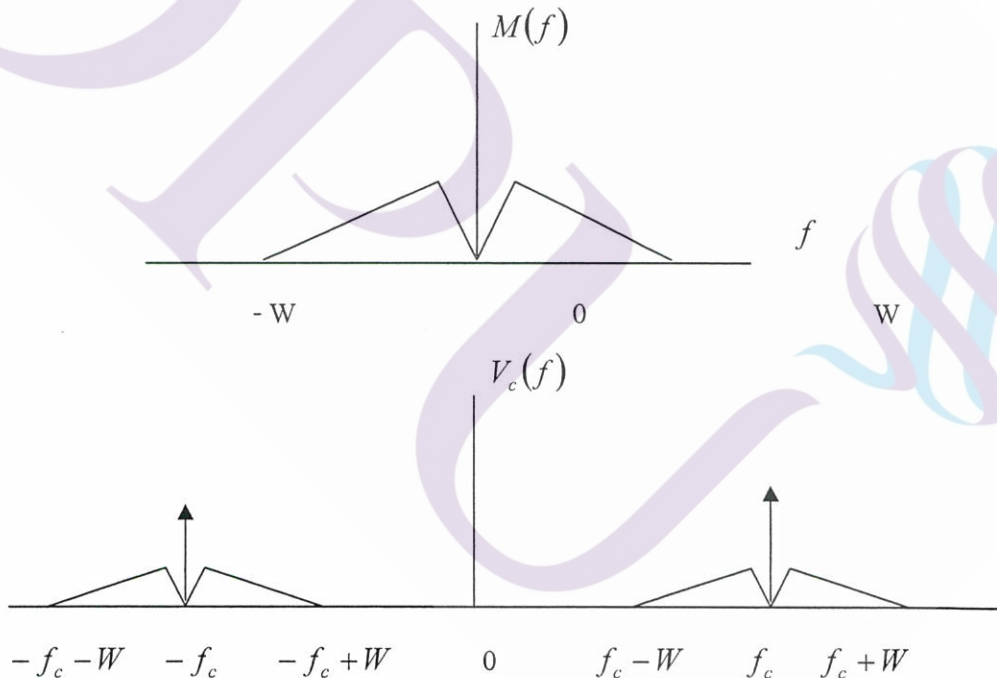
ในกรณีที่ $m(t)$ เป็นสัญญาณรูปไซน์จะมีการนิยามเปอร์เซ็นต์ของมอดูเลชันในรูปต่อไปนี้

$$m_a = \frac{(V_{\max} - V_{\min})/2}{(V_{\max} + V_{\min})/2} \times 100\% \quad (64)$$

2.2.10 การวิเคราะห์สเปกตรัมของสัญญาณเอเอ็ม

สัญญาณ AM ที่มีรูปร่างตามสมการที่ (46) หรือสมการที่ (47) นี้ ประกอบด้วย 2 เทอม เราจะสามารถทำการแปลงฟูรีเยร์ของสัญญาณได้โดยการแปลงฟูรีเยร์ของ $m(t)$ เป็น $M(f)$ เราจะได้การแปลงฟูรีเยร์ของสมการที่ (47) ออกมาในรูปนี้

$$V_c(f) = \frac{A_c}{2} \{\delta(f + f_c) + \delta(f - f_c)\} + \frac{A_c}{2} \{M(f + f_c) + M(f - f_c)\} \quad (65)$$



รูปที่ 2.22 สเปกตรัมของสัญญาณ AM

สิ่งที่น่าสังเกต คือสเปกตรัมของสัญญาณ AM ที่ได้นี้จะมีสเปกตรัมของสัญญาณเบสแบนด์โอบสเปกตรัมคลื่นพาห้สมมาตรซ้ายขวา ความถี่คลื่นพาห้จะสูงกว่าแบนด์วิดท์ของสัญญาณเบสแบนด์

มากคือ $f_c \gg W$ เพราะฉะนั้นแบนด์วิดท์ของสัญญาณ AM ก็จะกลายเป็น $2W$ ในขณะที่สัญญาณเบสแบนด์วิดท์เป็น W เพราะฉะนั้นสัญญาณ AM จึงเป็นสัญญาณที่มีประสิทธิภาพในการใช้แบนด์วิดท์ต่ำ ในลำดับต่อไปเราจะพิจารณากำลังเฉลี่ยของสัญญาณ AM จากสมการที่ (64) เราจะสามารถหาค่ากำลังเฉลี่ยของสัญญาณ AM ได้ดังนี้

$$\begin{aligned} P_{AM} &\triangleq \langle V_c^2(t) \rangle \\ &= \frac{A_c^2}{2} \langle 1 + 2m(t) + m^2(t) \rangle + \frac{A_c^2}{2} \langle [1 + m(t)]^2 \cos^2 \omega_c t \rangle \\ &= \frac{A_c^2}{2} [1 + 2 \langle m(t) \rangle + \langle m^2(t) \rangle] \end{aligned} \quad (66)$$

เพราะว่า $\langle \cos 2\omega_c t \rangle = 0$ และ $m(t)$ เป็นสัญญาณแรนด้อมที่มี $W \ll f_c$ และถ้าเราให้สัญญาณเบสแบนด์ไม่มีส่วนประกอบกระแสตรงอยู่ซึ่งจะเป็นจริงในกรณีของสัญญาณออดิโอและสัญญาณอื่นๆ อีกหลายแบบจะได้ $\langle m(t) \rangle = 0$ สมการ (66) ก็จะเขียนได้เป็น

$$P_{AM} = \frac{A_c^2}{2} + \frac{A_c^2}{2} \langle m^2(t) \rangle \quad (67)$$

เทอมที่ หนึ่งในสมการ (66) นี้จะเห็นได้ชัดว่าเป็นกำลังของคลื่นพาห้ ส่วนเทอมที่สองนั้นเป็นกำลังสัญญาณส่วนที่เหลือคือกำลังของไซด์แบนด์ด้านล่างรวมกัน และเนื่องจากไซด์แบนด์ด้านบนและด้านล่างมีความสมมาตรกันทุกประการ กำลังสัญญาณที่อยู่ในไซด์แบนด์ด้านบนและด้านล่างก็จะมีขนาดเท่ากันด้วย ให้ p_c และ p_{sb} แสดงกำลังของคลื่นพาห้และกำลังของสัญญาณที่อยู่ในไซด์แบนด์ด้านเดียว

$$P_{AM} = p_c + 2p_{sb} \quad (68)$$

โดยที่

$$p_{sb} = \frac{A_c^2}{4} \langle m^2(t) \rangle = \frac{p_c}{2} \langle m^2(t) \rangle \quad (69)$$

และเนื่องจาก $|m(t)| \leq 1$ เสมอ ดังนั้น $\langle m^2(t) \rangle \leq 1$ เสมอด้วย และจะส่งผลให้ $p_{sb} \leq p_c / 2$ เสมอเมื่อแทนเงื่อนไขดังกล่าวนี้ลงในสมการ (69) จะได้ดังนี้

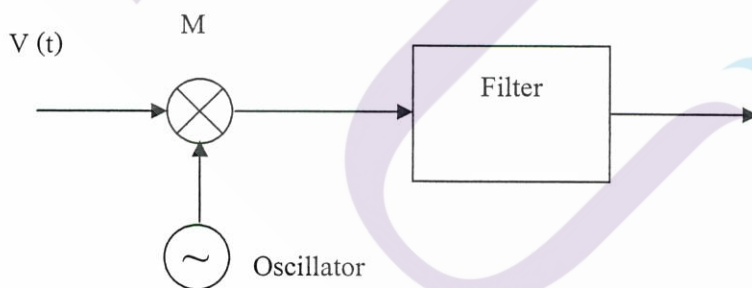
$$\begin{aligned} p_c &= P_{AM} - 2p_{sb} \geq P_{AM} - p_c \\ \therefore p_c &\geq \frac{1}{2} P_{AM} \\ \text{และ} \quad p_{sb} &\leq \frac{1}{4} P_{AM} \end{aligned} \quad (70)$$

2.2.10.1 การคิมอดูเลตสัญญาณแอนะล็อก

สัญญาณที่ได้รับการคิมอดูเลตเชิงขนาดนั้น จะเป็นสัญญาณเบสแบนด์ที่ถูกเลื่อนความถี่ให้สูงขึ้น ดังนั้นการคิมอดูเลตสัญญาณก็เป็นกระบวนการในการเลื่อนความถี่นั้นให้กลับมาอยู่ที่ความถี่เบสแบนด์อย่างเดิม ซึ่งอาจจะทำได้หลายวิธี เช่น ทำการคิมอดูเลตเป็นความถี่เบสแบนด์เลย หรือทำการแปลงเป็นความถี่ที่ต่ำลงขั้นหนึ่งก่อน แล้วค่อยขยายคิมอดูเลตสัญญาณเบสแบนด์ต่อไป การแปลงความถี่จึงจัดว่าเป็นกระบวนการพื้นฐานในการคิมอดูเลตสัญญาณ

เราอาจทำได้โดยใช้วงจรถูก เพราะวงจรถูกจะทำการคูณสัญญาณเบสแบนด์กับสัญญาณคลื่นพาห์ทำให้เกิดการเลื่อนความถี่สูงขึ้นได้ ในทำนองเดียวกันถ้าเรามีสัญญาณ AM หรือสัญญาณแบบอื่นเข้ามา เราก็สามารถใช้วงจรถูกในการเลื่อนสัญญาณนั้นไปอยู่ที่ความถี่ใหม่ได้ รูปที่ 2.23 แสดงบล็อกไดอะแกรมของระบบแปลงความถี่ของสัญญาณที่กล่าวถึงนี้ ให้สัญญาณขาเข้าเป็นสัญญาณ AM ที่เขียนได้เป็น $v(t) = A_c [1 + V_m(t)] \cos \omega_c t$ เมื่อเราคูณสัญญาณจากออสซิลเลเตอร์ที่มีความถี่เป็น ω_0 และมีขนาดเป็น A_0 เราก็จะได้

$$\begin{aligned} V(t) \cos \omega_0 t &= A_0 A_c [1 + V_m(t)] \cos \omega_c t \cos \omega_0 t \\ &= A_0 \frac{A_c}{2} [1 + V_m(t)] \cos(\omega_c + \omega_0) t \\ &\quad + A_0 \frac{A_c}{2} [1 + V_m(t)] \cos(\omega_c - \omega_0) t \end{aligned} \quad (71)$$



รูปที่ 2.23 ระบบแปลงความถี่ของสัญญาณที่ใช้วงจรถูก

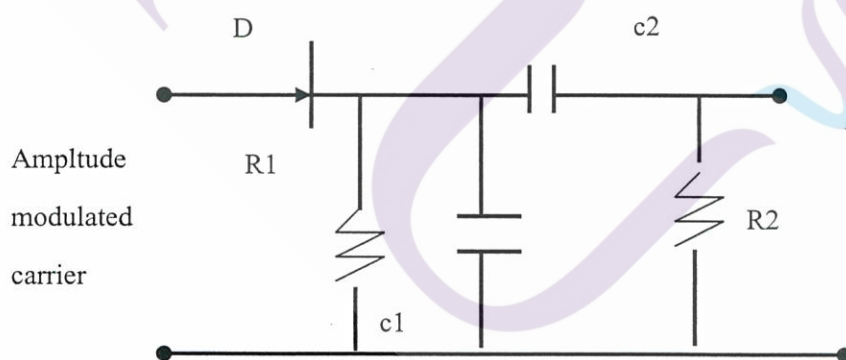
ได้ว่ารูปร่างของสัญญาณก็ยังคงเป็นสัญญาณ AM แต่ขนาดจะขึ้นอยู่กับ A_0 และความถี่ของสัญญาณคลื่นพาห์จะถูกแปลงไปอยู่ที่ $\omega_c + \omega_0$ และ $\omega_c - \omega_0$ ใหญ่กว่าแบนด์วิดท์ของสัญญาณเบสแบนด์เราจะใช้ฟิลเตอร์กรองเอาเฉพาะความถี่ด้านสูง หรือความถี่ด้านต่ำถ้าฟิลเตอร์เอาสัญญาณด้านความถี่สูงขึ้นมา เรียกวงจรที่แปลงความถี่นี้ว่า (up-converter) ถ้ากรองเอาสัญญาณด้านความถี่ต่ำออกมาเรียกวงจรนี้ว่า (down-converter) กรณีของเครื่องรับ

ทั่วไปนั้นเรามักใช้ดาวนคอนเวอร์เตอร์มากกว่าอัปคอนเวอร์เตอร์ เพราะ จัดการกับความถี่ต่ำลงมักจะง่ายกว่ากระบวนการที่เราแปลงความถี่นี้เรียกกันว่า (mixing) หรือ (heterodyning) การแปลงความถี่ที่กล่าวมานั้นนอกจากจะใช้ประโยชน์ในการดีมอดูเลตสัญญาณสื่อสารแล้วสามารถใช้ประโยชน์ในทางด้านอื่นได้อีกมากมายหลายอย่างนั้น ซึ่งได้แก่ใช้ในการวิเคราะห์เกี่ยวกับความถี่ (spectrum analyzer) ใช้ในการสแกนเบิลสัญญาณเสียงใช้ในการสร้างออสซิลเลเตอร์ เป็นต้น

2.2.10.2 เอ็นเวลโลปดีเทคชัน (Envelope Detection)

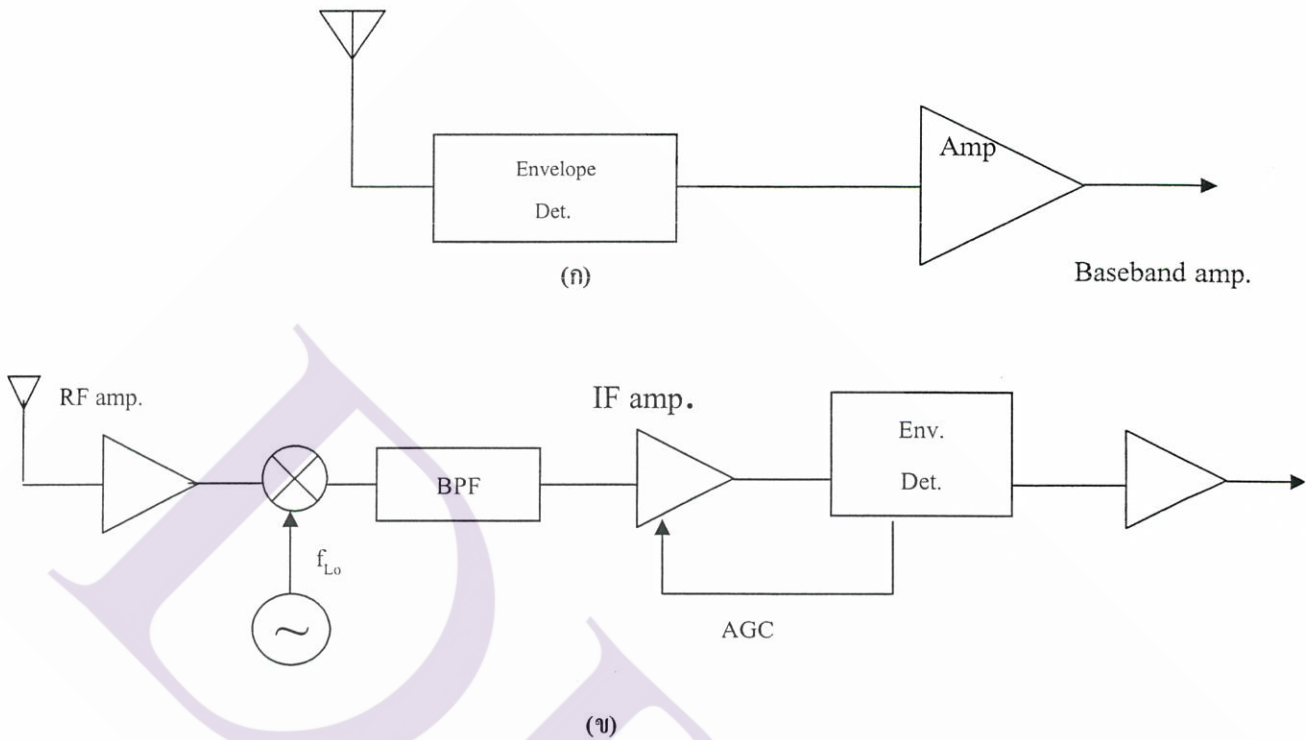
เอ็นเวลโลปดีเทคชันซึ่งหมายถึงการกรองแสงเอ็นเวลโลปของสัญญาณ AM ออกมา วงจรที่แสดงไว้ในรูปที่ 2.24 ไดโอดในวงจรจะกรองสัญญาณด้านบวกให้ผ่านไปได้ ในขณะที่ R_1C_1 ซึ่งทำหน้าที่เป็นฟิลเตอร์ผ่านความถี่ต่ำจะกรองเอาส่วนที่เป็นเอ็นเวลโลปออกมา การที่วงจร R_1C_1 จะทำงานได้ดีนั้นค่าคงที่ (time constant) ของวงจร หรือ R_1C_1 นั้น เมื่อเทียบกับคาบของสัญญาณคลื่นพาห์จะต้องยาวกว่ามาก ๆ ในขณะเดียวกันเมื่อเทียบกับสัญญาณเบสแบนด์ก็ต้องสั้นกว่าหรือมีการเปลี่ยนแปลงที่เร็วกว่า

$$W \ll \frac{1}{R_1C_1} \ll fc \quad (72)$$



รูปที่ 2.24 ตัวอย่างวงจรเอ็นเวลโลปดีเทคชัน

สำหรับ R_2C_2 นั้นจะทำหน้าที่สกัดส่วนประกอบกระแสตรงออกซึ่งเป็นสาเหตุที่ทำให้เอ็นเวลโลปดีเทคชันไม่เหมาะที่จะนำไปใช้งานที่ส่วนประกอบความถี่ต่ำมีความยากมาก



รูปที่ 2.25 การรับสัญญาณ AM (ก) แบบเอนเวลโลปดีเทกชัน (ข) แบบซูเปอร์เฮเตอร์โรไดน์

ในเครื่องรับสัญญาณ AM นั้นเราอาจจะนำคลื่นที่รับได้จากสายอากาศมาทำเอนเวลโลปดีเทกชันโดยตรง แล้วนำสัญญาณดีเทคได้ไปขยายออกถ้าโพงดังแสดงไว้ในรูปที่ 2.25(ก) แต่โดยทั่วไปที่รับมาได้จากสายอากาศมักจะมีกำลังต่ำ ถ้านำไปดีเทคทันทีก็อาจจะมีค่าเล็กเกินไปที่จะนำไปขยายทำการขยายต่อให้มีคุณภาพดีได้ดั่งนั้น โดยทั่วไป จึงมักมักจะใช้วิธีที่แสดงไว้ในรูปที่ 2.25(ข) คือใช้วงจรขยายทำการขยายสัญญาณที่รับมา ได้ชั้นหนึ่งก่อน และเพื่อจะให้คุณสมบัติของการรับดีขึ้น เรามักจะใช้มิกเซอร์แปลงความถี่ให้ต่ำลงเป็นความถี่ IF (Intermediate Frequency) แล้วทำการขยายสัญญาณในย่านความถี่ IF อีกชั้นหนึ่งก่อน แล้วจึงทำเอนเวลโลปดีเทกชัน การรับสัญญาณในลักษณะที่กล่าวมานี้เราเรียกว่าแบบซูเปอร์เฮเตอร์โรไดน์ (super-heterodyne) อนึ่งวิทย์ AM ที่ใช้อยู่ในปัจจุบันมีความถี่ IF เป็น 455 kHz

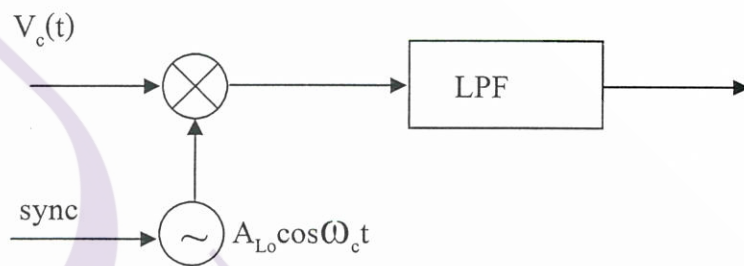
2.2.10.3 ซิงโครนัสดีเทกชัน (synchronous detection)

ซิงโครนัสดีเทกชัน หรือบางครั้งเรียกว่า โคฮีเรนต์ดีเทกชัน (Coherent Detection) คือวิธีการดีเทคสัญญาณจากออสซิลเลเตอร์ (Local Oscillator) คูณเข้ากับสัญญาณที่รับเข้ามาโดยที่สัญญาณ

จากโลกออสซิลเลเตอร์จะซิงโครไนซ์กับสัญญาณคลื่นพาห้ทั้งความถี่และเฟส ถ้าความถี่ของโลกออสซิลเลเตอร์คือ ω_0 เท่ากับ ω_c เราจะได้สัญญาณผลคูณออกมาในรูปต่อไปนี้

$$v(t)\cos\omega_c t = \frac{A_c}{2}[1+v_m(t)] + \frac{A_c}{2}[1+v_m(t)]\cos 2\omega_c t \quad (73)$$

ประกอบด้วยสัญญาณเบสแบนด์กับสัญญาณ AM ที่มีความถี่คลื่นพาห้เป็น $2\omega_c$ เราก็จะสามารถกรองเอาเฉพาะส่วนที่เป็นสัญญาณเบสแบนด์ออกมา โดยใช้ฟิลเตอร์ผ่านความถี่ต่ำที่มีแบนด์วิดท์เป็น W หรือ ใหญ่กว่า W เล็กน้อย



รูปที่ 2.26 บล็อกไดอะแกรมของซิงโครนัสดีเทกชัน

ถึงแม้สมการที่ (73) จะแสดงกรณีของสัญญาณในระบบ AM ในกรณีของสัญญาณ DSB ก็-จะเห็นได้ว่าใช้ได้เช่นเดียวกัน โดยที่เทอมเป็น $A_a/2$ และ $(A_a \cos 2\omega_c t)/2$ จะไม่ปรากฏออกมาทำนองนั้นเอง

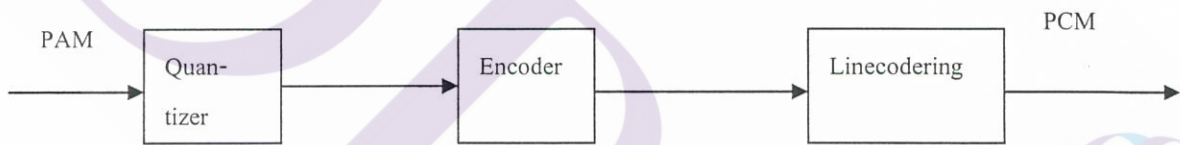
2.2.11 หลักการในการเปลี่ยนสัญญาณแอนะล็อกไปเป็นสัญญาณดิจิทัล

พัลส์โค้ดมอดูเลชันหรือที่เรียกกันย่อ ๆ ว่า PCM นั้น เป็นระบบสื่อสารแบบดิจิทัลที่ได้รับการพัฒนาและนำมาใช้ในระบบโทรศัพท์เมื่อต้นทศวรรษของ พ.ศ.2500 ระบบ PCM เป็นระบบที่ใช้ประโยชน์จากทฤษฎีสุ่มตัวอย่างโดยจัดการกับสัญญาณ PAM ที่ได้รับจากการสุ่มตัวอย่างให้เป็นสัญญาณที่เหมาะสมกับการส่งผ่านระบบมากขึ้น เพราะสัญญาณ PAM เมื่อส่งผ่านระบบจะประสบกับปัญหาการผิดเพี้ยนซึ่งทำให้สเปกตรัมของสัญญาณเบสแบนด์ที่ปลายทางผิดเพี้ยนไปจากเดิมและเมื่อทำการดีมอดูเลตได้ก็จะผิดเพี้ยนไปจากสัญญาณเดิม และเมื่อทำการดีมอดูเลตด้วยฟิลเตอร์ผ่านความถี่ต่ำ สัญญาณที่ดีมอดูเลตได้ก็จะผิดเพี้ยนไปจากสัญญาณเดิมอย่างหลีกเลี่ยงไม่ได้ ระบบ PCM นั้นเป็นการจัดการกับสัญญาณ PAM นี้ โดยการนำสัญญาณ PAM ไปทำการเข้ารหัส (coding) เป็นสัญญาณดิจิทัลแล้วจึงนำสัญญาณดิจิทัลที่ได้นั้นไปส่งผ่านระบบต่อไป และทางภาครับก็จะทำการถอดรหัส (decoding) เป็นสัญญาณ PAM แล้วนำสัญญาณ PAM นั้นไปดีมอดูเลตเป็นสัญญาณเบสแบนด์กลับคืนมา การส่งสัญญาณดิจิทัลนั้นจะส่งได้ทั้งแบบเบสแบนด์และแบบเบสพาส โดยที่

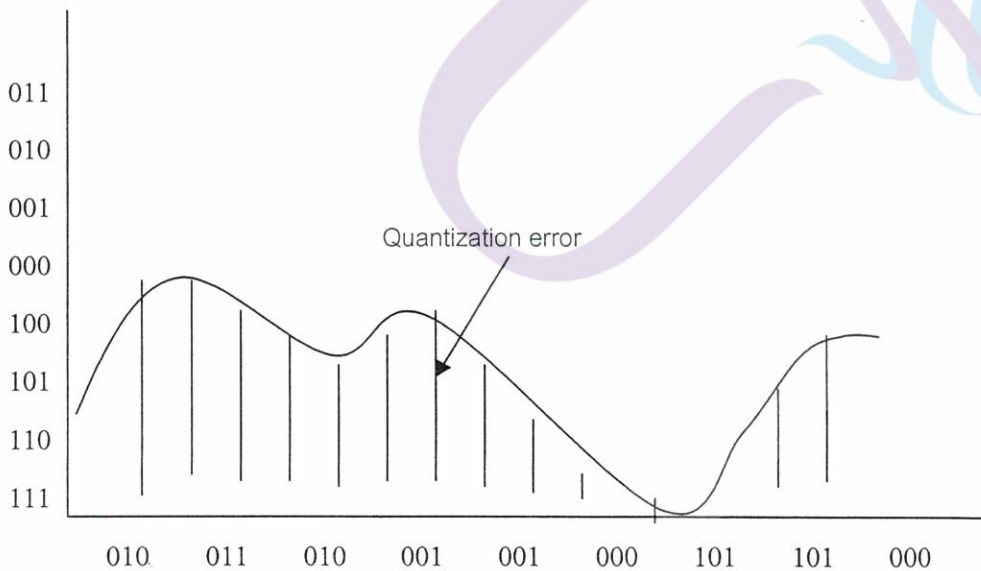
แบบแบนด์พาสนั้นเป็นการมอดูเลตสัญญาณดิจิทัลเข้ากับความถี่คลื่นพาห้ สำหรับในหัวข้อนี้จะกล่าวถึงขั้นตอนในการสร้างสัญญาณ PCMมาตรฐานที่ใช้ในงานโทรศัพท์ และสัญญาณรบกวนที่เกิดขึ้นในกระบวนการนี้

2.11.1 ขั้นตอนในการสร้างสัญญาณ PCM

สัญญาณ PCM นั้นจะสร้างจากสัญญาณ PAM ที่ได้จากการสุ่มตัวอย่างของสัญญาณจริง โดยใช้จำนวนบิตจำกัดโดยทั่วไป จำนวนบิตที่ใช้ในการเข้ารหัสจะเป็น 8 บิต ซึ่งสามารถแสดงระดับสัญญาณได้ทั้งหมด 256 ระดับ ดังนั้นจึงจำเป็นต้องปัดระดับของสัญญาณ PAM ให้ลงตามระดับที่เตรียมไว้ขั้นตอนในการจัดให้ระดับลงตัวนี้เรียกว่า การแควนไตซ์สัญญาณ (quantization) ในการแควนไตซ์สัญญาณนี้ถ้าให้แต่ละขั้นของระดับสัญญาณมีขนาดเท่ากัน ก็จะเป็นการแควนไตซ์สัญญาณแบบเชิงเส้น รูปที่ 2.28 แสดงตัวอย่างของการแควนไตซ์สัญญาณแบบเชิงเส้นนี้ สัญญาณ PAM ที่ถูกแควนไตซ์เรียบร้อยแล้วจะถูกนำไปเข้ารหัสด้วยสัญญาณดิจิทัล 8 บิต เป็นสัญญาณ PCM ออกมา



รูปที่ 2.27 ขั้นตอนการสร้างสัญญาณ PCM

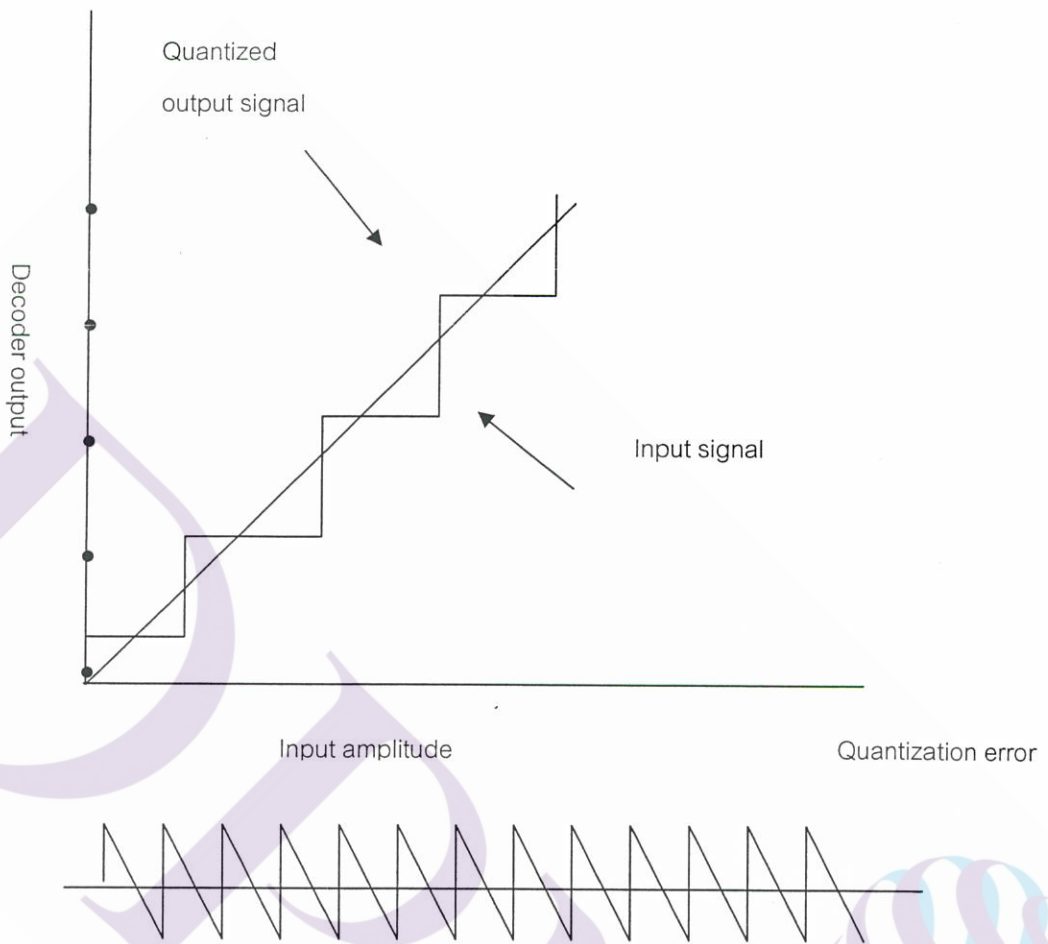


รูปที่ 2.28 ตัวอย่างการแควนไตซ์สัญญาณแบบเชิงเส้น

จากสัญญาณ PCM ที่รับมาได้ก็จะทำการถอดรหัสออกมาเป็นสัญญาณ PAM ที่เป็นระดับลงตัว และเมื่อผ่านสัญญาณ PAM นี้เข้าสู่วงจรฟิลเตอร์ความถี่ต่ำ ก็จะได้สัญญาณแอนะล็อกที่ ใกล้เคียงกับสัญญาณเดิมออกมา ที่กล่าวว่าใกล้เคียงกับสัญญาณเดิมก็เพราะว่าสัญญาณ PAM ที่ได้จากการถอดรหัสนั้นจะมีระดับลงตัวและอาจจะไม่เท่ากับสัญญาณเดิมก็ได้ ลักษณะเช่นนี้เหมือนกับการเกิดสัญญาณรบกวนแล้วทำให้ระดับของสัญญาณของระดับเปลี่ยนไป สัญญาณรบกวนในลักษณะนี้เรียกว่า แควนไทเซชันนอยส์ (quantization) แควนไทเซชันนอยส์เป็นสัญญาณรบกวนที่เกิดขึ้นจากการแควนไทซ์ จะพิจารณากรณีที่การแควนไทซ์เป็นแบบเชิงเส้น โดยให้ช่วงกว้างของแต่ละขั้นของระดับสัญญาณที่ถูกแควนไทซ์เป็น q ในการวิเคราะห์หาแควนไทเซชันนอยส์นี้จะสมมติฐานดังต่อไปนี้

1. ระดับสัญญาณที่ถอดรหัสได้จะแตกต่างจากระดับของสัญญาณจริงสูงสุดไม่เกิน $q/2$
2. ระดับของสัญญาณที่ถูกสุ่มตัวอย่างมาจะมีพรอบบะบิลิตีที่จะอยู่ที่ระดับต่าง ๆ ในช่วง q ได้เท่า ๆ กันนั่นคือ พรอบบะบิลิตีเดอนซิตีฟังก์ชันจะเป็น $1/q$
3. ระดับสัญญาณสูงสุดจะต้องอยู่ในช่วงการทำแควนไทซ์ ถ้าระดับสัญญาณสูงกว่าค่าสูงสุดที่จะทำการแควนไทซ์ได้ ก็จะเกิดการผิดเพี้ยนเนื่องจากสาเหตุนี้ขึ้นอีก





รูปที่ 2.29 การเกิดแควนไตเซชันนอยส์กรณีของการแควนไตซ์แบบเชิงเส้น

รูปที่ 2.29 การเกิดแควนไตเซชันนอยส์ตามสมมติฐานดังกล่าวนี้ ถ้าให้ $v(t)$ เป็นสัญญาณแอนะล็อก และ $V_D(t)$ เป็นสัญญาณที่ถอดรหัสได้ กำลังของสัญญาณรบกวนแบบแควนไตเซชันนอยส์นี้จะเขียนได้เป็น

$$\begin{aligned}
 N_Q &= E\{[v_D(t) - v(t)]^2\} \\
 &= \int_{-q/2}^{q/2} \frac{1}{q} \varepsilon^2 d\varepsilon \\
 &= \frac{q^2}{12}
 \end{aligned}
 \tag{74}$$

ตามสมมติฐานที่กล่าวไว้ข้างต้น กำลังของแควนไตเซชันนอยส์นี้จะเท่ากันทุก ๆ ช่วงของการแควนไตซ์โดยไม่ขึ้นอยู่กับระดับของสัญญาณแอนะล็อก ดังนั้นถ้าให้สัญญาณแอนะล็อกมีค่า rms ของสัญญาณเป็น v_{rms} เราจะคำนวณกำลังของสัญญาณต่อกำลังของแควนไตเซชันนอยส์หรือ SQR (signal-to-quantization noise ratio) ได้ดังต่อไปนี้

$$\begin{aligned} \text{SQR} &= 10 \log \frac{E[v^2(t)]}{E\{[v_D(t) - v(t)]^2\}} \\ &= 10 \log \left(\frac{v_{rms}^2}{q^2 / 12} \right) = 10.8 + 20 \log \left(\frac{v_{rms}}{q} \right) \end{aligned} \quad (75)$$

กรณีที่สัญญาณแอนะล็อกนั้นเป็นคลื่นรูปไซน์มีแอมพลิจูดเป็น A สมการ (84) ก็จะเขียนได้เป็น

$$\text{SQR} = 10 \log \left(\frac{A^2 / 2}{q^2 / 12} \right) = 7.8 + 20 \log \left(\frac{A}{q} \right) \quad (76)$$

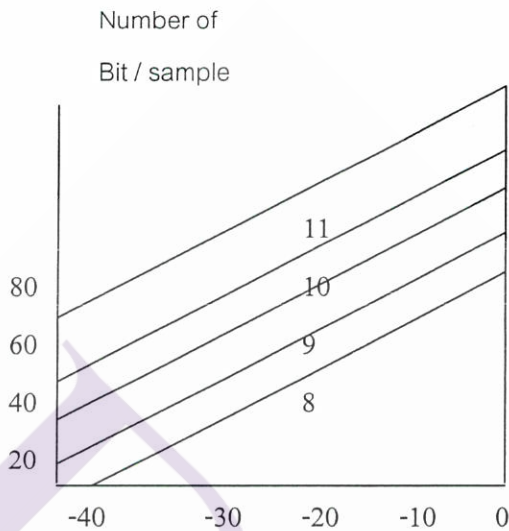
เนื่องจากตามเงื่อนไขข้อ (3) ช่วงของการแควนไตซ์จะต้องครอบคลุมระดับสูงสุดของสัญญาณอนาล็อก ดังนั้นถ้าให้ A_{max} เป็นระดับสัญญาณสูงสุด แล้วทำการแควนไตซ์ด้วยขั้นละ q เราจะต้องการำนวนขั้นทั้งหมดคือ M เป็นดังนี้

$$M = \frac{2A_{max}}{q} \quad (77)$$

จำนวนขั้นที่ได้ตามสมการที่ 77 นั้นจะเป็นจำนวนลงตัวโดยการปิดขึ้นหรือปิดลง เมื่อเราใช้สัญญาณดิจิทัลแบบ ไบนารีซึ่งมี 2 ระดับสัญญาณคือ "0" กับ "1" เราจะต้องใช้จำนวนบิต N เป็นดังนี้

$$\begin{aligned} M &= 2^N \\ \text{หรือ} \quad N &= \log_2 M \end{aligned} \quad (78)$$

ในกรณีที่จำนวนระดับ M ไม่ทำให้ N เป็นจำนวนลงตัว ก็ต้องปิดขึ้นให้เป็นจำนวนลงตัว ยกตัวอย่างเช่น ถ้า $M=200$ เมื่อคำนวณ N จะได้ 7.6 ซึ่งเป็นจำนวนเลขไม่ลงตัว และจำเป็นต้องทำการปิดขึ้นให้เป็นตัวเลขลงตัวคือ 8 ซึ่งก็คือต้องใช้สัญญาณไบนารีจำนวน 8 บิต ในการแควนไตซ์สัญญาณที่มี 200 ระดับ การใช้จำนวนบิต ตามตัวอย่างที่กล่าวมานี้ นับว่าเป็นการใช้บิตข้อมูลที่ไม่มีประสิทธิภาพ ฉะนั้นการเข้าโดยทั่วไปมักจะแบ่งจำนวนระดับในการแควนไตซ์ ให้เหมาะสมกับจำนวนบิตที่จะใช้ในการเข้ารหัส ตามตัวอย่างเช่น ถ้าใช้สัญญาณไบนารี 8 บิต ก็จะทำการแควนไตซ์เป็น 256 ระดับ เป็นต้น



รูปที่ 2.30 SQR ของสัญญาณ PCM ที่มีขึ้นของการแควนไตซ์สม่ำเสมอ

เมื่อเราทำการหาที่ q จะได้ $q = 2A_{\max}/2^N$ และ SQR ของกรณีที่ระดับสัญญาณเป็น A ในขณะทีระดับสัญญาณสูงสุดเป็น A_{\max} ในรูปต่อไปนี

$$\text{SQR} = 10 \log \left[\frac{(A^2/2) \times 12}{4A_{\max}^2/2^{2N}} \right] = 1.76 + 6.02N + 20 \log \left(\frac{A}{A_{\max}} \right) \quad (79)$$

เมื่อนำสมการที่ (79) นี้ไปพล็อตกราฟจะได้ตามรูปที่ 2.30 จากรูปจะเห็นได้ว่าค่า SQR จะสูงขึ้นเมื่อระดับของสัญญาณ A เข้าใกล้สัญญาณสูงสุดคือ A_{\max} และเนื่องจากในมาตรฐานของระบบโทรศัพท์กำหนดไว้ว่า SQR จะต้องสูงกว่า 26 dB ได้ จำนวนบิตที่ต้องการใช้นั้นจะคำนวณได้จากสมการที่ (79) แล้วเรากำหนด N โดยการปัดขึ้นผลที่ได้ให้เป็นเลขจำนวนลงตัวซึ่งในกรณีนี้คำนวณออกมาได้ว่า $N = 10$ เมื่อใช้สัญญาณไบนารีจำนวน 10 บิต ในการเข้ารหัสปรากฏว่า SQR ที่ได้ทีระดับสัญญาณสูงสุดจะได้ถึงประมาณ 62 dB ซึ่งจัดได้ว่าเกินสูงเกินกว่าความจำเป็น และที่สำคัญคือสัญญาณระดับสูง ๆ เกือบเท่า A_{\max} ไม่ค่อยจะเกิดขึ้นบ่อยนัก

การแควนไตซ์สัญญาณแบบเชิงเส้นนั้นถึงแม้จะทำได้ง่ายแต่ก็มักข้อเสียหลัก 2 ประการ คือ มีโอกาสที่จะใช้จำนวนบิตในการลงรหัสอย่างไม่มีประสิทธิภาพค่า SQR นั้นจะเปลี่ยนไปตามระดับสัญญาณโดยที่ระดับสัญญาณที่สูงกว่าจะมี SQR ข้อเสียดังกล่าวนี้ทำให้ระบบ PCM ที่มีแควนไตซ์สัญญาณแบบเป็นเชิงเส้นไม่ถูกนำมาใช้ในระบบโทรศัพท์วิธีการที่ถูกนำมาใช้โดยทั่วไปคือ ระบบที่มีการทำคอมแพนดิง (companding)

2.3 ทฤษฎีเบื้องต้นของระบบสื่อสารแบบดิจิทัล

2.3.1 การมอดูเลตและการดีมอดูเลตสัญญาณ

การมอดูเลตและการดีมอดูเลตสัญญาณหมายถึง การนำสัญญาณไฟฟ้าที่ได้จากทรานสดิวเซอร์ทางด้านต้นทางผสมเข้าไปกับสัญญาณคลื่นพาห์ เพื่อฝากข่าวสารที่มีอยู่ในสัญญาณไฟฟ้าเข้าไปกับสัญญาณคลื่นพาห์ที่มีความถี่สูงกว่ามาก ๆ การมอดูเลตสัญญาณเป็นสิ่งที่จำเป็นอย่างยิ่ง ถ้าต้องการส่งสัญญาณออกไปโดยคลื่นแม่เหล็กไฟฟ้าส่งผ่านทางบรรยากาศ ทั้งนี้เพราะเราไม่สามารถสร้างระบบสายอากาศให้ทำงานได้ดีที่ความถี่ต่ำ ๆ

ประโยชน์ของการมอดูเลตนั้นมีหลายประการด้วยกัน ประโยชน์ที่เด่นชัดมาก ๆ ก็คือทำให้เราส่งสัญญาณหลาย ๆ สัญญาณพร้อม ๆ กันได้ โดยการมอดูเลตเข้าไปกับคลื่นพาห์ ที่มีความถี่ต่างกัน ซึ่งเราเรียกวิธีการนี้ว่า การมัลติเพลกซ์เชิงความถี่ (Frequency Division Multiplexing ย่อว่า FDM) การสื่อสารที่ใช้การมัลติเพลกซ์เชิงความถี่ที่เห็นได้ชัดคือ ระบบวิทยุกระจายเสียงและวิทยุโทรทัศน์ที่เราใช้อยู่ในปัจจุบัน

อนึ่งการส่งสัญญาณแบบที่ใช้การมอดูเลตมักเรียกว่า ระบบส่งสัญญาณแบบแบนด์พาส (Band-pass Transmission System) และระบบที่ใช้การมัลติเพลกซ์เชิงความถี่ด้วยนั้นมักจะเรียกว่า ระบบส่งสัญญาณแบบบรอดแบนด์ (Broadband Transmission System) สำหรับการส่งสัญญาณโดยส่งในรูปของสัญญาณเดิมนั้นเรียกว่า ระบบส่งสัญญาณแบบเบสแบนด์ (Baseband Transmission System)

2.3.2 การเข้ารหัสและการถอดรหัส

เนื่องจากข่าวสารที่ทำการส่งผ่านระบบสื่อสาร นั้นเมื่อทำการศึกษาคุณสมบัติเชิงสถิติให้ดีจะพบว่ามีความซ้ำซ้อนของข่าวสารอยู่มาก ดังนั้นการเข้ารหัสจึงเป็นสิ่งที่จำเป็น เพราะจะทำให้สามารถลดความซ้ำซ้อนของข่าวสารลงได้ซึ่งหมายถึง จะสามารถส่งข่าวสารนั้นได้ด้วยปริมาณบิตที่น้อยลงและส่งได้เร็วขึ้น ทางด้านภาครับก็จะมีระบบถอดรหัสที่สอดคล้องกับทางภาคส่ง ซึ่งก็จะทำให้ได้ข่าวสารเดิมกลับคืนมา

2.3.3 การผิดเพี้ยนของสัญญาณจากการส่งผ่านระบบเชิงเส้น

ในกรณีที่สัญญาณที่ส่งผ่านระบบแล้ว ไม่ผิดเพี้ยนไปจากสัญญาณขาออกจะต้องเขียนในรูปของสัญญาณขาเข้าได้ดังนี้

$$y(t) = Kx(t - t_d) \quad (80)$$

โดยที่ K เป็นค่าคงที่ และ t_d เป็นเวลาหน่วง ค่า K นี้สำหรับวงจรไฟฟ้าที่ประกอบด้วยชิ้นส่วนแพสซีฟ (passive component) เท่านั้นจะมีค่าต่ำกว่า 1 เสมอ

และทรานสเฟอ์ฟังก์ชัน $H(f)$ จะเขียนได้เป็น

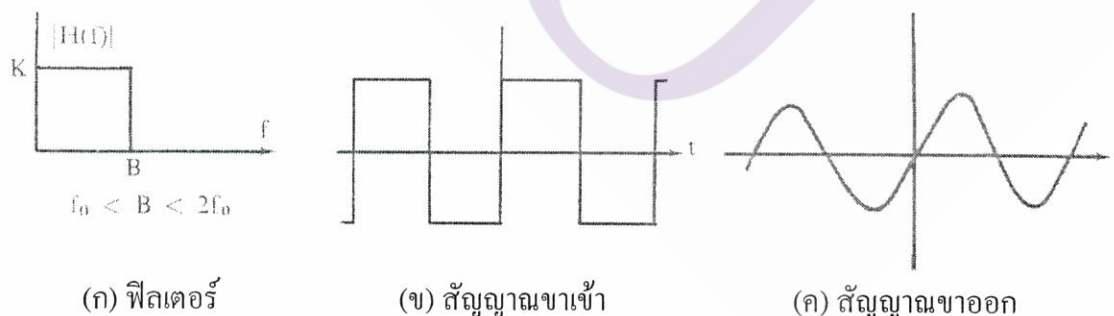
$$|H(f)| = K \quad (81)$$

และ
$$\arg H(f) = -2\pi_d f \quad (82)$$

เงื่อนไขในสมการที่ 81 และ 82 เป็นการระบุว่าค่า $|H(f)|$ เป็นค่าคงที่ และ $\arg H(f)$ จะต้องเป็นฟังก์ชันเชิงเส้นของความถี่โดยที่สมการที่ 81 และ 82 นี้จะต้องเป็นจริงจากช่วงความถี่ $-\infty$ ถึง ∞ อย่างไรก็ตาม สัญญาณที่ใช้งานอยู่ในภาคปฏิบัติมักจะประกอบด้วย สเปกตรัมของช่วงความถี่ที่จำกัด ดังนั้น ถ้าสมการ 81 และ 82 เป็นจริงในช่วงความถี่ที่เกี่ยวข้อง สัญญาณขาออกก็ จะไม่ผิดเพี้ยนไปจากสัญญาณขาเข้า แต่ถ้าคุณสมบัติตามสมการที่ 81 และ 82 ไม่เป็นจริงในช่วง ความถี่ที่เกี่ยวข้อง สัญญาณขาออกก็จะผิดเพี้ยนไปจากสัญญาณขาเข้าอย่างแน่นอนตามสมการที่ 81 และ 82 จะเห็นได้ว่า การผิดเพี้ยนจะเกิดขึ้นได้จาก 2 สาเหตุคือ

- (1) การผิดเพี้ยนเนื่องจาก $|H(f)| \neq K$ เรียกว่าการผิดเพี้ยนเชิงขนาด (amplitude distortion)
- (2) การผิดเพี้ยนเนื่องจาก $\arg H(f) \neq -2\pi_d f$ เรียกว่าการผิดเพี้ยนเชิงเวลาหน่วง (delay distortion)

การผิดเพี้ยนเชิงขนาดนั้นเป็นสิ่งที่เข้าใจง่าย เพราะถ้าส่งสัญญาณขาเข้าที่ประกอบด้วยหลาย ๆ ความถี่เข้าไปในระบบ แล้วระบบต้องสนองเชิงขนาดต่อความถี่ต่าง ๆ ไม่เหมือนกัน การกระจายของสเปกตรัมของสัญญาณขาออกย่อมแตกต่างไปจากสัญญาณขาเข้า ผลก็คือทำให้รูปร่างของสัญญาณขาออกผิดเพี้ยนไปจากสัญญาณขาเข้า รูปที่ 2.31 แสดงตัวอย่างของระบบเชิงเส้นที่ให้ ความถี่พื้นฐานของสัญญาณคลื่นสี่เหลี่ยมผ่านเท่านั้น เมื่อฮาร์โมนิกส์ของสัญญาณผ่านไม่ได้ สัญญาณขาออกก็จะเหลือเพียงความถี่เดียวเป็นสัญญาณไซน์ ดังที่แสดงไว้ในรูป (ค)



(ก) ฟิลเตอร์

(ข) สัญญาณขาเข้า

(ค) สัญญาณขาออก

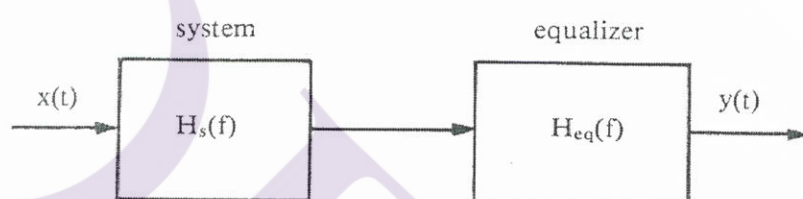
รูปที่ 2.31 ตัวอย่างการผิดเพี้ยนเชิงขนาด

2.3.4 อีควอไลเซอร์

อีควอไลเซอร์คือ วงจรที่ใช้ชดเชยการบิดเบี้ยวของสัญญาณให้กลับไปมีรูปเดิม ดังนั้นถ้าพิจารณาในอาณาจักรของความถี่ ผลตอบสนองอิมพัลส์ของวงจรอีควอไลเซอร์จะต้องสัมพันธ์กับผลตอบสนองอิมพัลส์ของระบบเชิงเส้นก่อนหน้านั้น ในลักษณะที่ทำให้ผลตอบสนองอิมพัลส์รวมมีคุณสมบัติตามสมการที่ 81 และ 82 กล่าว ดังสมการที่ 83

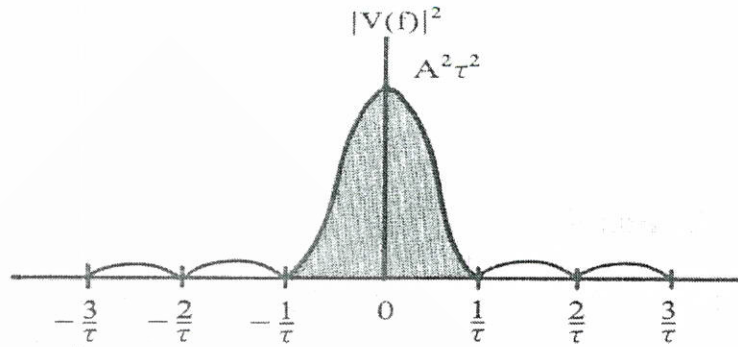
$$H_s(f)H_{eq}(f) = Ke^{-j2\pi ftd} \quad (83)$$

โดยที่ $H_s(f)$ และ $H_{eq}(f)$ เป็นผลตอบสนองอิมพัลส์ของระบบและของอีควอไลเซอร์ตามลำดับ สำหรับอีควอไลเซอร์ที่สมบูรณ์แบบตามสมการที่ 83 นี้จะต้องเป็นจริงที่ทุก ๆ ความถี่ที่ $X(f) \neq 0$



รูปที่ 2.32 การใช้อีควอไลเซอร์แก้ปัญหาการบิดเบี้ยวของสัญญาณ

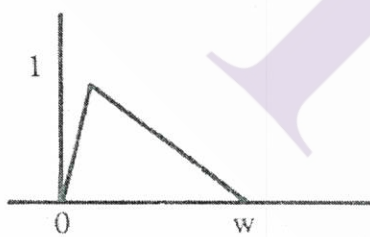
แสดงการส่งผ่านสัญญาณที่มีอีควอไลเซอร์ กรณีที่สัญญาณขาเข้ามีแบนด์วิดท์แคบและ $H_s(f)$ ของระบบไม่ซับซ้อน การสร้าง อีควอไลเซอร์ให้ได้ $H_{eq}(f)$ ตามสมการ 83 ก็อาจทำได้ไม่ยากนัก แต่ถึงกระนั้นอีควอไลเซอร์ที่ได้ก็มักจะบั่นทอนกำลังของสัญญาณไปอีกไม่มากนัก ซึ่งเป็นค่าใช้จ่ายที่หลีกเลี่ยงไม่ได้ ในกรณีที่สัญญาณขาเข้ามีแบนด์วิดท์กว้าง เช่น เป็นสัญญาณรูปพัลส์สี่เหลี่ยมอีควอไลเซอร์ชนิดสมบูรณ์แบบจะสร้างได้ยากขึ้น ในกรณีเช่นนี้มักจะทำการสร้างอีควอไลเซอร์ให้จัดการกับสเปกตรัมส่วนใหญ่ของสัญญาณได้ เช่น สามารถจัดการกับ 90% ของกำลังของสัญญาณก็อาจจะเพียงพอเพราะการบิดเบี้ยวก็จะลดลงไปได้มาก รูปที่ 2.33 แสดงความหนาแน่นสเปกตรัมพลังงานของสัญญาณพัลส์สี่เหลี่ยม และตำแหน่งที่พลังงานของสัญญาณรวมได้ 90% ซึ่งจะแคบลงมาก เมื่อเปรียบเทียบกับสเปกตรัมของสัญญาณ ดังนั้นในกรณีทั่วไปเรามักจะทำตามลักษณะที่กล่าวมานี้ นอกจากนี้แล้วเราอาจจะช่วยให้การออกแบบของอีควอไลเซอร์ง่ายขึ้น ส่งสัญญาณต้นทางให้มีแบนด์วิดท์แคบลง โดยใช้ฟิลเตอร์ช่วย



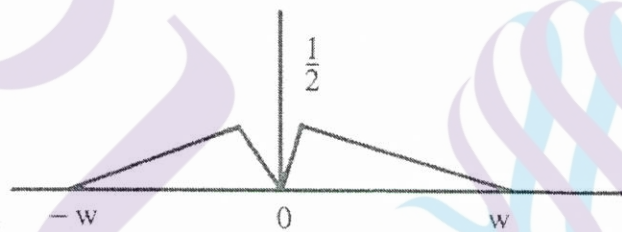
รูปที่ 2.33 ความหนาแน่นสเปกตรัมพลังงานของพัลส์สี่เหลี่ยม

2.3.5 การมอดูเลตแบบ AM (Amplitude modulation)

โดยทั่วไปสัญญาณเบสแบนด์ที่ทำการส่งนั้นจะมีความถี่สูงสุดที่มีค่าจำกัด เช่น สัญญาณออกดีโอดิโคนที่หูคนเรารับฟังได้อาจจะสูงเพียง 15 kHz และถ้าเป็นสัญญาณเสียงที่คนเราพูดกัน โดยทั่วไปก็มักจะมีค่าไม่เกิน 4 kHz เป็นต้น เมื่อเป็นเช่นนี้ถ้าเราให้ความถี่สูงสุดของสัญญาณเบสแบนด์เป็น W Hz เมื่อเขียนสเปกตรัมของสัญญาณนั้นแบบแสดงความถี่ด้านบวกด้านเดียวและแบบแสดงความถี่ทั้งสองด้านจะเป็นดังที่แสดงไว้ในรูปที่ 2.34 (ก) และ (ข) ตามลำดับ



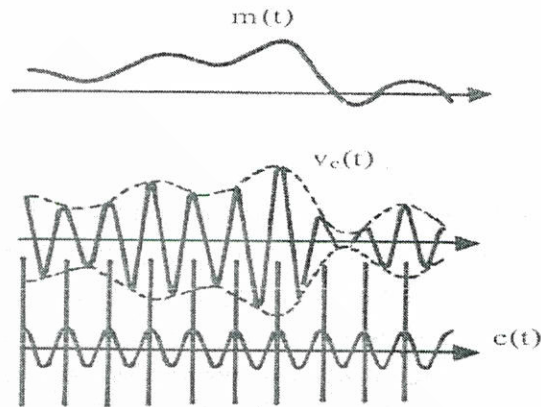
(ก) แสดงความถี่ด้านเดียว



(ข) แสดงความถี่สองด้าน

รูปที่ 2.34 ตัวอย่างสเปกตรัมของสัญญาณอนาล็อกที่มีความถี่สูงสุดต่ำกว่า W

ในกรณีที่แสดงความถี่สองด้านนั้น สเปกตรัมทางด้านบวกด้านลบจะสมมาตรกัน และมีขนาดเป็น $1/2$ ของสเปกตรัมที่แสดงความถี่บวกด้านเดียว การมอดูเลตแบบ AM ที่ใช้ในวิทยุ AM โดยทั่วไปนั้นจะเป็นการสร้างสัญญาณที่มีลักษณะเหมือนกับที่แสดงไว้ในรูปที่ 2.35 กล่าวคือ เหมือนกับนำสัญญาณเบสแบนด์ไปซ้อนอยู่บนสัญญาณคลื่นพาห้



รูปที่ 2.35 รูปร่างของสัญญาณเบสแบนด์และสัญญาณ AM

ทำให้ยอดคลื่นของคลื่นพาห้หรือเ็นเวลโลป (Envelope) เปลี่ยนไปตามสัญญาณเบสแบนด์นั้น ในสภาพ ดังกล่าวนี้เราจะสามารถดีเทคสัญญาณเบสแบนด์กลับมาได้โดยใช้วงจรีเทคเตอร์ ซึ่งเป็นเหตุผลสำคัญที่ทำให้วิทยุ AM ถูกนำมาใช้งานก่อน การดีเทคยอดคลื่นของคลื่นพาห้นี้ เราเรียกว่า เ็นเวลโลปดีเทคชัน (envelope detection) ซึ่งเป็นวิธีดีเทคสัญญาณ AM วิธีหนึ่ง

สัญญาณ AM ที่มีรูปร่างดังที่แสดงไว้ในรูปที่ 2.35 นี้เมื่อเขียนเป็นฟังก์ชันของเวลาจะเป็นดังนี้

$$v_c(t) = \{A_c + v(t)\} \cos \omega_c t \quad (84)$$

โดย $A_c \cos \omega_c t$ เป็นสัญญาณคลื่นพาห้ และ $v(t)$ เป็นสัญญาณเบสแบนด์ และโดยทั่วไปเราจะแสดงสัญญาณเดียวกันนี้ในรูปต่อไปนี้

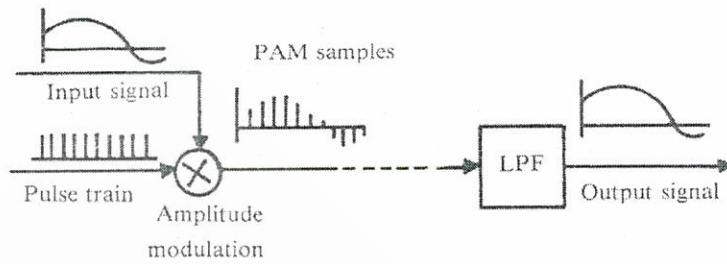
$$v_c(t) = A_c \{1 + m(t)\} \cos \omega_c t \quad (85)$$

โดยที่ $m(t)$ เป็นสัญญาณเบสแบนด์ที่ถูกนอร์มอลไลซ์ไว้ด้วยค่า A_c และ $m(t)$ จะถูกเรียกว่ามอดูเลชันอินเด็กซ์ (modulation index) ซึ่งจะเป็นตัวชี้บ่งว่าสัญญาณคลื่นพาห้ถูกทำให้มีการเปลี่ยนแปลงไปอย่างไร

2.3.6 ผลกระทบที่เกิดขึ้นในกรณีที่อัตราสุ่มตัวอย่างสูงไม่เพียงพอ

การสุ่มตัวอย่างของสัญญาณมีความถี่ของการสุ่มคือ $f_s = 2W$ อันที่จริงในระบบทั่วไปที่ไม่ใช่ทางอุดมคตินั้น ความถี่ของการสุ่มจะต้องสูงกว่า $2W$ เสมอ ในหัวข้อนี้จะขอกกล่าวถึงความจำเป็นที่ f_s จะต้องสูงกว่า $2W$ ดังกล่าวนี้ และผลกระทบที่จะเกิดขึ้นในกรณีที่ $f_s < 2W$

ในการพิจารณาการหาค่าฟูเรียร์ของสัญญาณสเปกตรัมของสัญญาณสุ่มตัวอย่าง โดยใช้ระบบอุดมคติในรูปที่ 3.32 ตามรูปสัญญาณที่ออกจากวงจรคูณ



รูปที่ 2.36 บล็อกไดอะแกรมแสดงการสุ่มตัวอย่างและการสร้างสัญญาณเดิมขึ้นมาแบบอุดมคติ

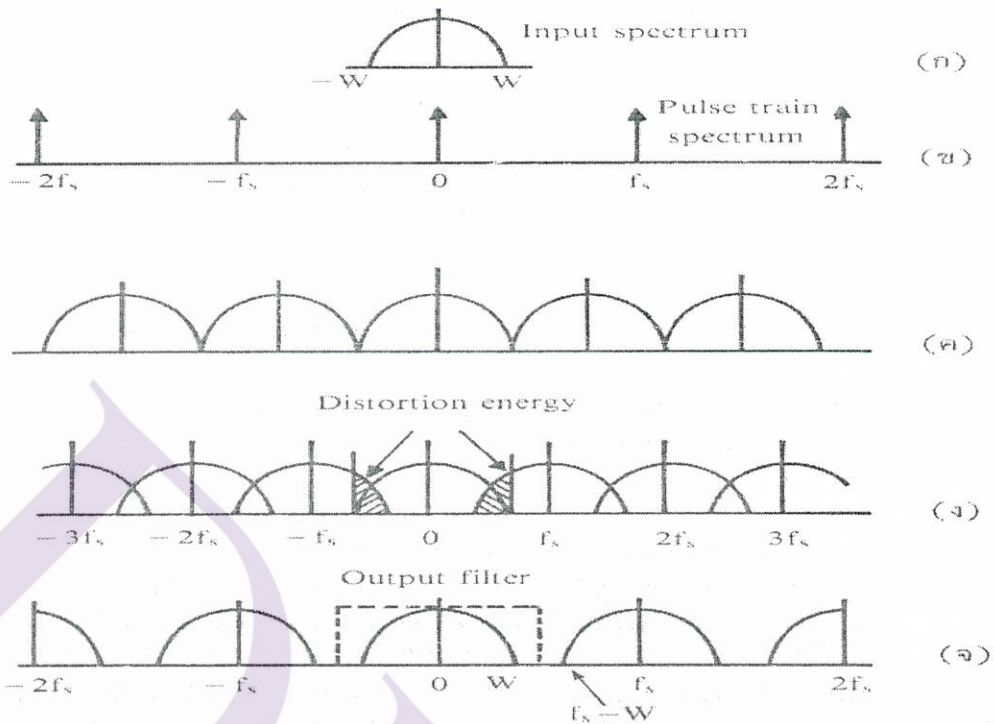
ดังนั้น จะเขียนได้เป็นสมการที่ 86

$$v_s(t) = v(t) \sum_{n=-\infty}^{\infty} \delta(t - nT_s) \quad (86)$$

เมื่อทำการหาฟูรีเยร์ทรานสฟอร์มของสมการที่ 86 นี้จะได้

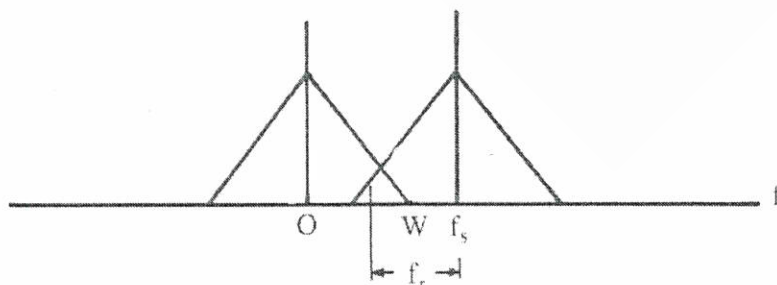
$$V_s(f) = \sum_{n=-\infty}^{\infty} f_s V(f - nf_s) \quad (87)$$

เมื่อนำฟูรีเยร์สเปกตรัมที่ได้ตามสมการ 87 นี้ไปเขียนเป็นรูปก็จะได้ดังรูปที่ 2.37 โดยที่รูป (ก) แสดงสเปกตรัมของสัญญาณอนาล็อกเบสแบนด์คือ $V(f)$ รูป (ข) แสดงสเปกตรัมของขบวนอิมพัลส์ที่มีความถี่เป็น nf_s รูป (ค) แสดงการกระจายของสเปกตรัมกรณีที่ทำกรสุ่มตัวอย่างด้วย $f_s = 2W$ รูป (ง) และ รูป (จ) นั้นแสดงกรณีที่ $f_s < 2W$ และ $f_s > 2W$ ตามลำดับ สเปกตรัมของสัญญาณที่แสดงไว้ในรูป (ค) ถึงรูป (จ) คือ สเปกตรัมของสัญญาณที่ถูกมอดูเลตแล้ว ลักษณะการมอดูเลตแบบนี้เรียกว่าพัลส์แอมพลิจูดมอดูเลชัน (Pulse Amplitude Modulation ย่อว่า PAM) เมื่อสัญญาณที่ถูกมอดูเลตแล้วมีสเปกตรัมดังที่แสดงไว้ในรูป (ค) ถึงรูป (จ) การทำดีมอดูเลชันซึ่งก็คือการดึงเอาเฉพาะสเปกตรัมของสัญญาณเบสแบนด์ออกมาก็อาจจะทำได้โดยใช้ฟิลเตอร์ผ่านความถี่ต่ำ อย่างไรก็ตาม รูป (ค) นั้นจะต้องใช้ฟิลเตอร์ผ่านความถี่ต่ำแบบอุดมคติจึงจะสามารถสกัดสเปกตรัมของความถี่ที่อยู่ติดกันได้



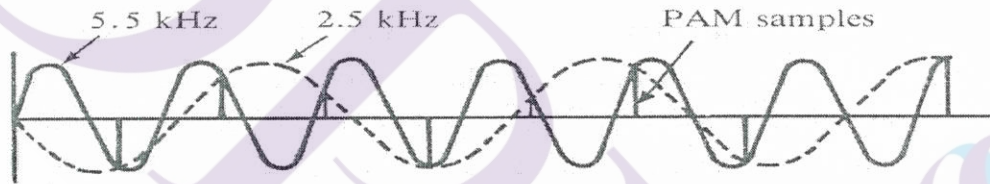
รูปที่ 2.37 สเปกตรัมของสัญญาณเบสแบนด์และสเปกตรัมของสัญญาณสุ่มตัวอย่างที่มีอัตราสุ่มต่าง ๆ กัน

แต่เนื่องจากฟิลเตอร์แบบอุดมคติ นั้นสร้างไม่ได้ เพราะฉะนั้น จึงไม่อาจป้องกันการรบกวนจากสเปกตรัมที่มีความถี่สูงกว่าดังกล่าวได้ สำหรับกรณีของรูป (ง) ซึ่ง $f_s < 2W$ นั้นลักษณะจะเหมือนกับการใช้ความถี่คลื่นพาห์ที่สูงไม่เพียงพอในการมอดูเลต ซึ่งทำให้สเปกตรัมของสัญญาณไซด์แบนด์ซ้อนกันอยู่ ในสภาพเช่นนี้ถึงแม้จะใช้ฟิลเตอร์ผ่านความถี่ต่ำแบบอุดมคติก็ไม่สามารถสกัดการรบกวนระหว่างไซด์แบนด์ได้ สำหรับกรณีที่ $f_s > 2W$ ตามรูป (จ) นั้นจะลดความยุ่งยากในการใช้ฟิลเตอร์ลงโดยเฉพาะถ้าช่วงการ์ดแบนด์ (guard band) คือ $f_s - 2W$ มีขนาดกว้างขึ้น ดังนั้นในกรณีทั่วไปจะเป็นไปตามเงื่อนไขในรูป (จ) นี้ สำหรับอัตราสุ่มตัวอย่าง $f_s = 2W$ นั้นเรียกว่าอัตราสุ่มตัวอย่างของไนควิสต์ (Nyquist sampling rate)



รูปที่ 2.38 การซ้อนกันของสเปกตรัมเมื่อ $f_s < 2W$

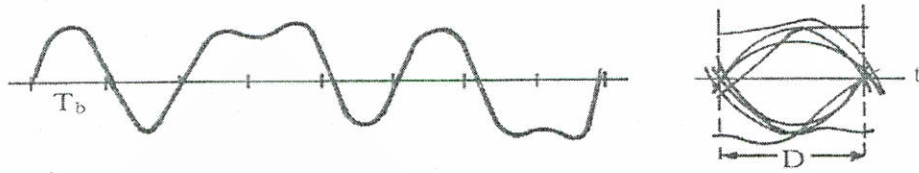
ในลำดับต่อไปจะพิจารณาผลกระทบของกรณีที่ $f_s < 2W$ ซึ่งทำให้ไซด์แบนด์ของสเปกตรัมมาซ้อนกันอยู่ดังรูปที่ 2.37 (ง) เพื่อให้เห็นภาพชัดขึ้นจะขอใช้รูปที่ 2.38 ในการอธิบายการคิมอดูเลตนั้น เมื่อใช้ฟิลเตอร์ผ่านความถี่ต่ำที่มีแบนด์วิดท์เป็น $f_s/2$ มาใช้ในการคิมอดูเลตสัญญาณซึ่งเราจะพบว่าสัญญาณที่คิมอดูเลตมาได้นั้นจะเกิดการบิดเบี้ยวขึ้นเพราะสเปกตรัมในช่วง $f_s/2$ ถึง W ผ่านฟิลเตอร์ไม่ได้ นอกจากนั้นสเปกตรัมที่อยู่ในช่วง $f_s - W$ ถึง $f_s/2$ ก็อาจจะเป็นของปลอมได้ กล่าวคือในช่วงที่สเปกตรัมของไซด์แบนด์ซ้อนกันอยู่นี้ f_r ซึ่งเป็นสเปกตรัมของความถี่เบสแบนด์จะทำให้เกิดสเปกตรัม $f_s - f_r$ ในความถี่เบสแบนด์เพิ่มขึ้นมาได้ดังที่แสดงในรูป นั้นหมายความว่าถึงแม้สัญญาณเดิมจะเป็นของปลอมที่ไม่ได้มีจริงตั้งแต่ต้น ลักษณะเช่นนี้เรียกว่าการเกิดสเปกตรัมปลอม (aliasing) ซึ่งเป็นสิ่งที่ต้องระวังอย่างมากในการออกแบบระบบจริง รูปที่ 2.39 แสดงการเกิดสเปกตรัมปลอมในกรณีที่ทำการสุ่มตัวอย่างสัญญาณไซน์ความถี่ 5.5 kHz ด้วยอัตราการสุ่ม 8 kHz แล้วทำการคิมอดูเลตด้วยฟิลเตอร์ผ่านความถี่ต่ำที่มีแบนด์วิดท์เป็น 4 kHz สเปกตรัมปลอมที่เกิดขึ้นจะมีความถี่เป็น $8 \text{ kHz} - 5.5 \text{ kHz} = 2.5 \text{ kHz}$ ซึ่งแสดงเป็นเส้นประไว้ในรูป 2.39



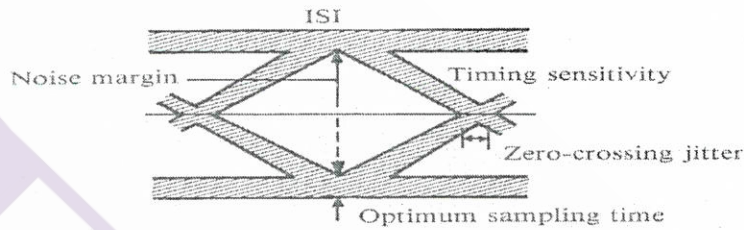
รูปที่ 2.39 การเกิดสเปกตรัมปลอมจาก 5.5 kHz เป็น 2.5 kHz เมื่ออัตราการสุ่มเป็น 8 kHz

2.3.7 แพทเทินรูปดวงตา (eye pattern)

ในการส่งสัญญาณดิจิทัลแบบเบสแบนด์นี้ ถ้าสัญญาณที่ส่งจากต้นทางเป็นสัญญาณเร้นดัมหรือ กึ่งเร้นดัม (pseudo random) เมื่อเราทำการมอนิเตอร์สัญญาณที่ภาครับตรงตำแหน่งก่อนเข้าวงจรตัดสินใจระดับโดยใช้ออสซิลโลสโคป โดยทั่วไปเราจะพบว่าการบิดเบี้ยวของสัญญาณระหว่างสัญลักษณ์ และเฟสจิดเตอร์ที่เกิดขึ้น จะทำให้ขอบบนของสัญญาณดิจิทัลที่มาซ้อน ๆ กันให้เห็นบนจอเกิดเป็นแทนเทินรูปดวงตาขึ้นมา รูปที่ 2.40 แสดงตัวอย่างของแพทเทินรูปดวงตาที่เกิดขึ้นในกรณีที่สัญญาณดิจิทัลมีไลน์โค้ดดิ้งแบบ โพลาร์ NRZ และเมื่อเราทำการเขียนขอบเขตของบริเวณต่าง ๆ ก็จะได้ตามรูปที่ 2.41 โดยที่ส่วนแกลงไวนั้นเป็นบริเวณที่มีสัญญาณมาปรากฏอยู่ เมื่อพิจารณาส่วนต่าง ๆ ในแพทเทินรูปดวงตาก็จะเป็นตามที่ได้แสดงไว้ในรูปที่ 2.41



รูปที่ 2.40 ตัวอย่างของแพทเทินรูปดวงตาในกรณีของไลน์โค้ดดิ้งแบบโพลาร์ NRZ



รูปที่ 2.41 แพทเทินรูปดวงตากับความหมายของส่วนต่าง ๆ

กล่าวคือ ส่วนที่เป็นแถบอยู่ด้านบนสุดและล่างสุด ก็จะเป็นผลกระทบจากการรบกวนระหว่างสัญลักษณ์ ส่วนที่เป็นแถบอยู่ตรงกลางจะเกิดจากเฟสจิตเตอร์ เมื่อพิจารณาจากแพทเทินรูปดวงตาที่เกิดขึ้นนี้ จะเห็นได้ว่าการตัดสินใจระดับสัญญาณนั้น ตำแหน่งการสุ่มตัวอย่างที่เหมาะสมที่สุดก็คือ ตำแหน่งตรงกลางของดวงตาซึ่งเป็นส่วนที่กว้างที่สุดของดวงตา และระดับสัญญาณตัดสินใจที่เหมาะสมที่สุดคือ ระดับตรงกลางของดวงตาเพราะเป็นระดับที่ทำให้มีอิมมาร์จิน (noise margin) ทางด้าน “ 1 ” และ “ 0 ” เท่ากัน ซึ่งจะส่งผลให้พروبะบิลิตีของการตัดสินใจผิดพลาดเนื่องจากสัญญาณรบกวนแบบเกาส์เซียนต่ำที่สุด สำหรับความชันของเส้นรอบดวงตานี้จะบ่งบอกความไวที่มีต่อการคลาดเคลื่อนของจังหวะในการสุ่มตัวอย่าง โดยทั่วไปสัญญาณที่มีอัตราข้อมูลสูงก็มักจะมี ความชันสูงด้วย ซึ่งหมายถึงผลกระทบของจังหวะการสุ่มตัวอย่างที่คลาดเคลื่อนไปก็จะมีสูงตามไปด้วย

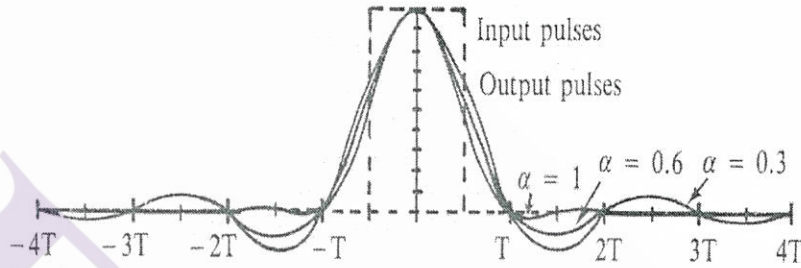
เมื่อก้าวโดยสรุปแล้ว แพทเทินรูปดวงตานี้จะบ่งบอกคุณสมบัติของช่องสัญญาณที่ทำการส่งผ่านสัญญาณดิจิทัลอยู่ และแพทเทินรูปดวงตาที่มีดวงตาเปิดกว้างทั้งในแนวตั้งและแนวนอนก็จะหมายถึงคุณสมบัติที่ดีของช่องสัญญาณ ซึ่งจะส่งผลให้การตัดสินใจในระดับสัญญาณมีโอกาสที่จะผิดพลาดน้อยลงด้วย

2.3.8 การแก้ปัญหาการผิดเพี้ยนของสัญญาณและการรบกวนข้ามสัญลักษณ์

การผิดเพี้ยนของสัญญาณเป็นสิ่งที่หลีกเลี่ยงไม่ได้ในระบบส่งสัญญาณดิจิทัล อย่างไรก็ตาม การรบกวนข้ามสัญลักษณ์นั้นมีโอกาสที่จะลดความรุนแรงลงหรือทำให้ไม่เกิดขึ้นได้ จากการศึกษาในเชิงทฤษฎีพบว่า รูปแบบหนึ่งของสัญญาณขาออกตรงตำแหน่งก่อนเข้าวงจรตัดสินใจระดับที่จะไม่ทำให้เกิดการรบกวนระหว่างสัญลักษณ์จะเขียนได้ในรูปต่อไปนี้

$$y_r(t) = \frac{\sin(\pi t/T) \cos(\alpha \pi t/T)}{\pi t/T \cdot 1 - (\alpha \pi t/T)^2} \tag{88}$$

เมื่อนำสัญญาณในสมการ 88 นี้ไปเขียนเป็นรูปออกมาจะเป็นดังที่แสดงไว้ในรูปที่ 2.42

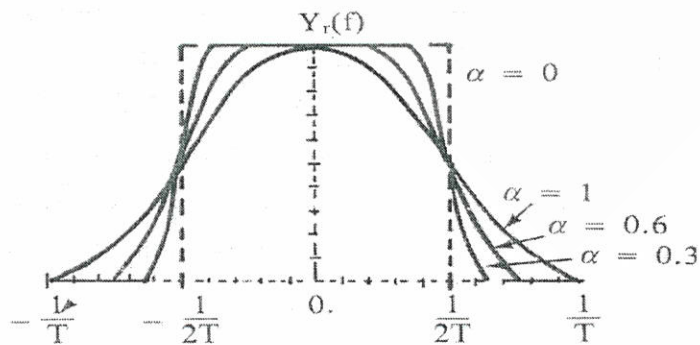


รูปที่ 2.42 สัญญาณพัลส์ขาออกที่ไม่เกิดการรบกวนระหว่างสัญลักษณ์

จากรูปจะเห็นได้ว่าที่ตำแหน่งกึ่งกลางของสัญลักษณ์ข้างเคียงสัญญาณตามสมการที่ 88 นี้ จะมีค่าเป็นศูนย์ซึ่งหมายถึงจะไม่เกิดการรบกวนระหว่างสัญลักษณ์ขึ้น ถ้าการสุ่มตัวอย่างมีจังหวะที่ถูกต้องเมื่อทำฟูเรียร์ทรานสฟอร์มเพื่อดูสเปกตรัมของสัญญาณในสมการที่ 88 นั้น จะพบว่ามีการกระจายของสเปกตรัมในรูปที่ 2.43

$$\begin{aligned}
 y_r(f) &= 1, & |f| &\leq \frac{(1-\alpha)}{2T} \\
 &= \frac{1}{2} \left[1 + \cos \left\{ \frac{\pi|f|T}{\alpha} - \frac{\pi(1-\alpha)}{2\alpha} \right\} \right], & \frac{(1-\alpha)}{2T} &\leq |f| \leq \frac{1+\alpha}{2T} \\
 &= \cos^2 \left[\frac{\pi|f|T}{2\alpha} - \frac{\pi(1-\alpha)}{4\alpha} \right] \\
 &= 0, & f &: \text{ค่าอื่นๆ}
 \end{aligned} \tag{89}$$

รูปสเปกตรัมตามสมการที่ 89 นี้ เมื่อนำไปเขียนกราฟจะได้ตามรูปที่ 2.43



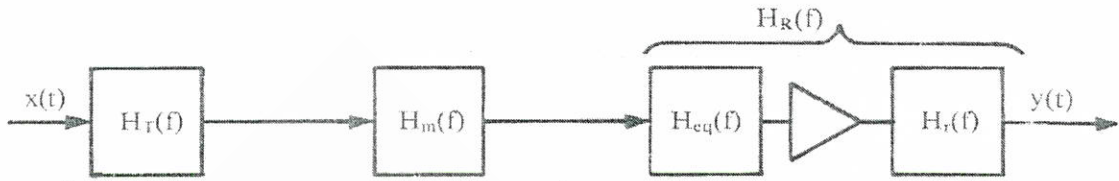
รูปที่ 2.43 ฟูเรียร์สเปกตรัมของสัญญาณพัลส์ในรูปที่ 2.42

ช่องสัญญาณที่ให้สัญญาณขาออกในรูปสมการ 88 ซึ่งมีผลตอบสนองความถี่ตามสมการ 89 นี้ เรียกว่าช่องสัญญาณแบบ “เรสต์โคไซน์” (raised cosine) ซึ่งหมายถึงรูปร่างของโคไซน์กำลังสองที่ปรากฏอยู่ในฟูเรียร์สเปกตรัม ช่วงที่ $(1-\alpha)/2T \leq |f| \leq (1+\alpha)/2T$ ค่า α นั้นจะอยู่ระหว่าง 0 ถึง 1 ถ้าเท่ากับ 0 ช่อง สัญญาณนั้นก็จะเป็ นฟิลเตอร์ผ่านความถี่ต่ำแบบอุดมคติที่มีแบนด์วิดท์ เท่ากับ $1/2T$ ซึ่งหมายความว่า เราสามารถใช้ช่องสัญญาณแบบอุดมคติที่มีแบนด์วิดท์เป็น $1/2$ ของอัตราการส่งสัญญาณโดยไม่ทำให้เกิดการรบกวนระหว่างสัญลักษณ์ได้ ในกรณีที่ α มีค่ามากกว่า 0 จากรูปที่ 2.43 จะเห็นได้ว่าช่องสัญญาณจะต้องมีแบนด์วิดท์ที่กว้างออกและไม่เป็นแบบอุดมคติ ในกรณีที่ช่องสัญญาณเป็นแบบผ่านความถี่ต่ำแบบอุดมคติ ในควิสต์ได้เป็นผู้ที่แสดงให้เห็นว่าแบนด์วิดท์ที่แคบจะไม่ทำให้เกิดการรบกวนระหว่างสัญลักษณ์ก็คือ $1/2T$ ถ้าแบนด์วิดท์แคบกว่านี้ก็จะทำให้เกิดการรบกวนระหว่างสัญลักษณ์ขึ้น ในกรณีที่แบนด์วิดท์กว้างกว่า $1/2T$ และเป็นจำนวนเท่าลงตัวของ $1/2T$ ก็จะสามารถให้สัญญาณขาออกที่ไม่มีการรบกวนระหว่างสัญลักษณ์ได้เช่นเดียวกัน

จากที่อธิบายมานี้จะเห็นได้ว่า สัญญาณที่มีฟูเรียร์สเปกตรัมตามสมการ 89 จะให้สัญญาณเชิงเวลาที่ไม่มีการรบกวนระหว่างสัญลักษณ์ อย่างไรก็ตามในระบบทั่วไปนั้นสัญญาณพัลส์ที่ส่งออกจากเครื่องส่งนั้น มักจะไม่ใช่อิมพัลส์แต่จะเป็นพัลส์สี่เหลี่ยมที่มีความกว้างจำกัดค่าหนึ่ง ถ้าให้ฟูเรียร์สเปกตรัมของสัญญาณขาออกนี้เป็น $X_s(f)$ ในการสร้างสัญญาณที่ไม่มีการรบกวนระหว่างสัญลักษณ์นั้นคุณสมบัติในการตอบสนองความถี่ของช่องสัญญาณนี้ คือ $H_c(f)$ จะเขียนได้ดังนี้

$$H_c(f) = \frac{y_r(f)}{X_s(f)} \quad (90)$$

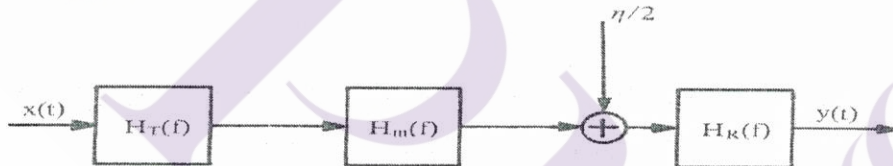
$H_c(f)$ ในสมการ 90 นี้จะแสดงคุณสมบัติในการตอบสนองความถี่รวมของสัญญาณ ก็จะเป็นผลคูณของทรานสเฟอร์ฟังก์ชันที่ฟิลเตอร์ของภาคส่งทรานสเฟอร์ฟังก์ชันของสายนำสัญญาณ และทรานสเฟอร์ฟังก์ชันของวงจรทางภาครับซึ่งได้แก่ วงจรอีควอไลเซอร์ วงจรขยายและวงจรฟิลเตอร์ รูปที่ 2.2-14 แสดงส่วนประกอบต่าง ๆ ที่กล่าวมานี้ จากรูปจะได้ข้อสังเกตว่า ทรานสเฟอร์ฟังก์ชันที่ภาคส่งและภาครับนั้น เราสามารถออกแบบวงจรส่วนต่าง ๆ ให้มีคุณสมบัติตามต้องการได้ แต่ทรานสเฟอร์ฟังก์ชันของสายนำสัญญาณนั้นจะขึ้นอยู่กับชนิดของสายนำสัญญาณ นอกจากนั้นยังขึ้นอยู่กับความยาวของสายนำสัญญาณ เมื่อทรานสเฟอร์ฟังก์ชันของสายนำสัญญาณเปลี่ยนไป ทรานสเฟอร์ฟังก์ชันของวงจรภาครับซึ่งมักจะเป็นอีควอไลเซอร์ก็ต้องเปลี่ยนแปลงให้สอดคล้องกันด้วย ดังนั้นในระบบ PCM จึงมักจะพยายามรักษาระยะห่างระหว่างเครื่องทวนสัญญาณเพื่อลดความยุ่งยากในการปรับแต่งวงจรอีควอไลเซอร์ดังกล่าวข้างต้น



รูปที่ 2.44 ส่วนประกอบต่าง ๆ ในช่องสัญญาณที่มีผลต่อการตอบสนองความถี่ของช่องสัญญาณ

2.3.9 แมทซ์ฟิลเตอร์ (matched filter)

การลดระดับของสัญญาณรบกวนจะสามารถทำได้โดยใช้ฟิลเตอร์ที่เรียกว่า แมทซ์ฟิลเตอร์ (matched filter) แมทซ์ฟิลเตอร์จะอาศัยหลักการที่ว่าวงจรรับจะรู้รูปร่างของสัญญาณล่วงหน้าถึงแม้จะไม่รู้ขนาด และในเวลาทีสัญญาณจะผ่านเข้ามาเมื่อมีรูปร่างของสัญญาณล่วงหน้าก็มีโอกาสที่จะสร้างฟิลเตอร์ที่มีทรานสเฟอร์ฟังก์ชันแมทซ์กับรูปร่างของสัญญาณที่จะผ่านเข้ามานั้นเพื่อให้มีระดับของสัญญาณต่อสัญญาณรบกวนสูงสุด การหาทรานสเฟอร์ฟังก์ชันของแมทซ์ฟิลเตอร์นั้นจะทำได้ตามขั้นตอนต่อไปนี้ ก่อนอื่นให้รูปร่างของสัญญาณพัลส์เป็น $x_c(t)$ และให้สัญญาณพัลส์นั้นเข้ามาถึงภาครับตามรูปที่ 2.45 มีขนาดเป็น A_R และเวลาที่มาถึงเป็น t_0 และ เขียนได้เป็น



รูปที่ 2.45 บล็อกไดอะแกรมของการหาทรานสเฟอร์ฟังก์ชันของแมทซ์ฟิลเตอร์

$$x_R(t) = A_R x_c(t - t_0) \quad (91)$$

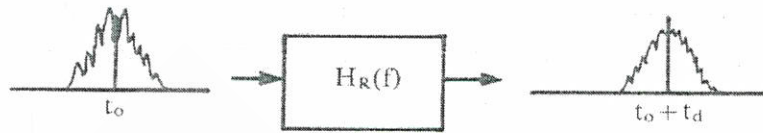
ถ้าให้ $X_c(f)$ เป็นฟูเรียร์ทรานสฟอร์มของ $x_c(t)$ เมื่อทำฟูเรียร์ทรานสฟอร์มสมการ 91 จะได้

$$X_R(f) = A_R X_c(f) e^{-j\omega t_0} \quad (92)$$

และเราสามารถหาพลังงานของพัลส์ได้ในรูปต่อไปนี้

$$E_R = \int_{-\infty}^{\infty} |X_R(f)|^2 df = A_R^2 \int_{-\infty}^{\infty} |X_c(f)|^2 df \quad (93)$$

ในขณะที่มีสัญญาณรบกวนแบบแอดคิทีฟเข้ามาพร้อมกับสัญญาณ $x_R(t)$ ฟิลเตอร์ทางภาครับจะต้องสามารถบีบพลังงานของพัลส์นั้นให้เกิดจุดสูงสุดขึ้น ในขณะเดียวกันก็ต้องทำให้พลังงานของสัญญาณรบกวนที่ผ่านออกมาได้มีขนาดต่ำที่สุด ถ้าให้ตำแหน่งที่เกิดจุดสูงสุดเป็น $t_0 + t_d$ ดังแสดงไว้ในรูปที่ 2.46



รูปที่ 2.46 ลักษณะการเกิดจุดสูงสุดเมื่อใช้เมทซ์ฟิลเตอร์รับสัญญาณ

และขนาดของสัญญาณที่จุดสูงสุดเป็น A เราจะสามารถหาขนาดสูงสุด A นี้ได้โดยนำสเปกตรัมของสัญญาณขาออกมาทำฟูเรียร์ทรานสฟอร์มกลับแล้วสังเกตที่เวลา $t_0 + t_d$ ถ้าให้ทรานสเฟอร์ฟังก์ชันของฟิลเตอร์เป็น $H_R(f)$ ดังในรูปที่ 2.45 A จะเขียนได้ดังนี้

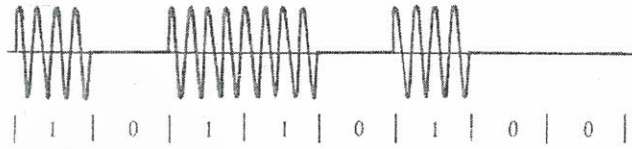
$$A = A_R \int_{-\infty}^{\infty} H_R(f) X_c(f) e^{j\omega t_d} df \quad (94)$$

เมื่อพิจารณาในด้านสัญญาณรบกวน ถ้าให้สัญญาณรบกวนนั้นมีขนาดกระจายอยู่ในรูปพรอบเบบิลิตีเดนซิตีฟังก์ชันแบบเกาส์เซียนโดยมีค่าเฉลี่ยเป็นศูนย์ ความหนาแน่นสเปกตรัมกำลังของสัญญาณรบกวนเป็น $G_n(f)$ กำลังของสัญญาณรบกวนที่จะผ่านฟิลเตอร์ออกไป ซึ่งจะเท่ากับค่าแวลเรียนซ์ของสัญญาณรบกวนแบบเกาส์เซียนที่ผ่านออกไปได้จะเขียนได้ในรูปต่อไปนี้

$$\sigma^2 = \int_{-\infty}^{\infty} |H_R(f)|^2 G_n(f) df \quad (95)$$

2.3.10 การมอดูเลตโดยการเปลี่ยนขนาดของสัญญาณคลื่นพาห้

การมอดูเลตโดยการเปลี่ยนขนาดของสัญญาณคลื่นพาห้ นั้น วิธีที่ง่ายที่สุดก็คือ การมอดูเลตให้ได้สัญญาณ AM ออกมา ในกรณีที่ทำการมอดูเลตให้สัญญาณ AM นั้นมีมอดูเลชันอินเด็กซ์เป็น 1 และไลน์โคดดิ้งของสัญญาณดิจิทัลที่เข้ามาเป็นแบบโพลาร์ NRZ รูปร่างของสัญญาณที่ได้ก็เป็นดังที่แสดงไว้ในรูปที่ 2.47 สัญญาณตามรูปนี้โดยทั่วไปเรียกว่า ออนออฟคีย์อิง (on-off keying ย่อว่า OOK) เพราะเหมือนกับการเปิดและปิดสัญญาณคลื่นพาห้ตามสัญญาณดิจิทัลที่เข้ามามอดูเลตสัญญาณ OOK นี้จัดว่าเป็นรูปแบบเฉพาะอันหนึ่งของสัญญาณแบบแอมปริจูดชิฟท์คีย์อิง (amplitude shift keying ย่อว่า ASK) สัญญาณ OOK นี้มีประวัติการใช้งานยาวนานมาขึ้นตั้งแต่สมัยเริ่มใช้คลื่นวิทยุในการสื่อสารใหม่ ๆ โดยใช้ในการส่งโทรเลข และความถี่ที่ใช้เป็นย่าน HF (High Free Quency) ปัจจุบันก็มีที่ใช้ในระบบไมโครเวฟบ้าง สำหรับที่ใช้ที่เด่นชัดที่สุดก็คือ ใช้ในระบบสื่อสารด้วยเส้นใยแสงซึ่งการมอดูเลตตัวกำเนิดแสงคือ เลเซอร์ไดโอดส่วนใหญ่จะใช้วิธีนี้



รูปที่ 2.47 รูปร่างของสัญญาณอนอ็อฟคีย์อิง (OOK)

ในกรณีของสัญญาณยูนิโพลาร์ NRZ 2 ระดับนั้น สัญญาณในอาณาจักรของเวลาและในอาณาจักรของความถี่จะเขียนได้ดังนี้

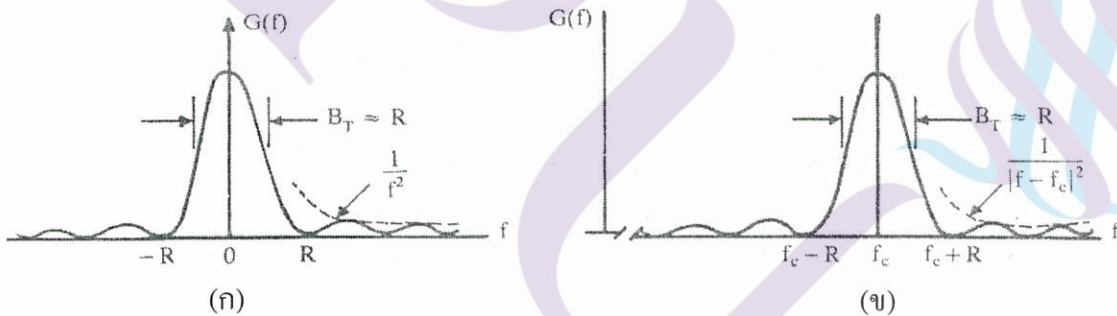
$$x(t) = \sum_k a_k p(t - kD) \quad (96)$$

$$|X(f)|^2 = \frac{1}{4R} \sin^2(\pi f D) + \frac{1}{4} \delta(f) \quad (97)$$

โดยที่ a_k เป็น 1 และ 0 เมื่อสัญญาณดิจิทัลเป็น 1 และ 0 ตามลำดับ และ $R = 1/D$ เป็นอัตราการเปลี่ยนแปลงของสัญญาณ และสัญญาณ OOK ในอาณาจักรของเวลาและความถี่จะเขียนได้ในรูปต่อไปนี้

$$x_c(t) = A_c x(t) \cos \omega_c t \quad (98)$$

$$G(f) = \frac{A_c^2}{4} |X(f - f_c)|^2 + \frac{A_c^2}{4} |X(f + f_c)|^2 \quad (99)$$



รูปที่ 2.48 ความหนาแน่นสเปกตรัมของสัญญาณยูนิโพลาร์ NRZ และความหนาแน่นสเปกตรัมของสัญญาณ OOK

ในการพิจารณาแบนด์วิดท์ที่ต้องการนั้น จากสมการที่ 97 และรูปที่ 2.48 จะเห็นได้ว่ากำลังของสัญญาณส่วนใหญ่จะกระจายในช่วง $f_c \pm R/2$ และการลดต่ำลงของความหนาแน่นสเปกตรัมกำลังจะแปรผกผันกับ $|f - f_c|^2$ ดังนั้นเราสามารถอนุโลมให้แบนด์วิดท์ที่ต้องการ $B_T \approx R$ ได้ จากผลดังกล่าวนี้จะได้ประสิทธิภาพของการใช้แบนด์วิดท์ของกรณีนี้เป็น 1 bps/Hz

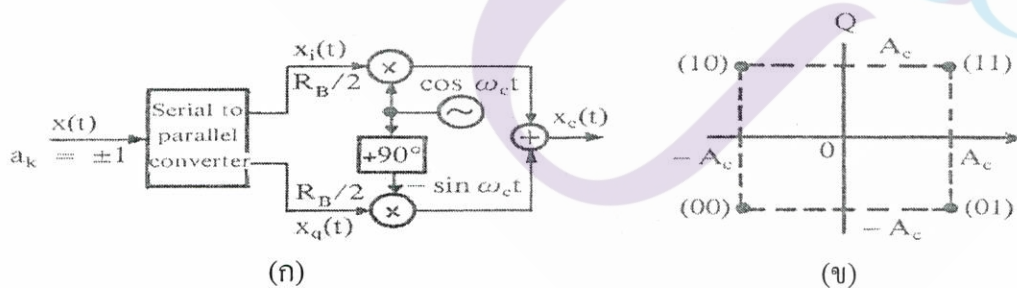
ในกรณีที่สัญญาณดิจิทัลที่เข้ามามี M ระดับ ถ้าให้สัญญาณนั้นเป็นแบบยูนิโพลาร์ NRZ เราจะเขียนสัญญาณในอาณาจักรของเวลาได้ในรูปต่อไปนี้

$$x(t) = \sum_k a_k p(t - kD) \quad a_k = 0, 1, 2, \dots, M-1 \quad (100)$$

และเมื่อทำการหาความหนาแน่นสเปกตรัมกำลังของสัญญาณนี้ อาศัยหลักการของสัญญาณเรณดัม โดยพิจารณาว่า $m_a = (M-1)/2$ และ $\sigma^2 = E[m^2] - m_a^2 = (M^2 - 1)/12$ ซึ่งจะทำให้ได้ $|X(f)|^2$ อยู่ในรูปต่อไปนี้

$$|X(f)|^2 = \frac{M^2 - 1}{12R} \text{sinc}^2\left(\frac{\pi f}{R}\right) + \frac{(M-1)^2}{4} \delta(f) \quad (101)$$

2.3.11 การมอดูเลตเชิงขนาดที่ใช้สัญญาณคลื่นพาห์อินเฟส และคลื่นพาห์ควอดเรเจอร์เฟส
การมอดูเลตเชิงขนาดที่ใช้สัญญาณคลื่นพาห์ที่มีเฟสต่างกัน 90° ซึ่งมีชื่อย่อว่า QAM (quadrature carrier amplitude modulation) นี้ เป็นการใช้คลื่นพาห์ที่มีเฟสต่างกัน 90° พร้อม ๆ กัน ตามรูปที่ 2.49 (ก) แสดงบล็อกไดอะแกรมของระบบมอดูเลตแบบ QAM นี้ ตามรูปสัญญาณดิจิทัลที่เข้ามาเป็นสัญญาณอนุกรมจะถูกแปลงให้เป็นสัญญาณขนานทีละ 2 บิต บิตแรกจะถูกส่งไปมอดูเลตสัญญาณคลื่นพาห์อินเฟส บิตที่สองจะถูกส่งไปมอดูเลตสัญญาณคลื่นพาห์ควอดเรเจอร์เฟส อัตราข้อมูลที่เข้ามาเป็น R_B bps อัตราข้อมูลที่ไปมอดูเลตคลื่นพาห์แต่ละด้านก็จะเป็น $R_B/2$ bps และสัญญาณที่ไปมอดูเลตคลื่นพาห์อินเฟส และคลื่นพาห์ควอดเรเจอร์เฟสจะเขียนได้ตามลำดับในรูปที่ 2.49



รูปที่ 2.49 บล็อกไดอะแกรมของการมอดูเลตแบบ QAM และการจัดตำแหน่งของสัญญาณ

$$x_i(t) = \sum_k a_{2k} p(t - kD) \quad (102)$$

$$x_q(t) = \sum_k a_{2k+1} p(t - kD) \quad (103)$$

สัญญาณ QAM ที่สร้างขึ้นก็จะเขียนได้ในรูปต่อไปนี้

$$x_c(t) = A_c [x_i(t) \cos \omega_c t - x_q(t) \sin \omega_c t] \quad (104)$$

ถ้าให้ a_k ที่มีค่าเป็น 1 และ -1 เมื่อสัญญาณดิจิทัลที่มีค่าเป็น 1 และ 0 ตามลำดับ สัญญาณ $x_c(t)$ ในสมการ 104 ซึ่งเป็นสัญญาณที่มีข้อมูลเข้ามามอดูเลตทีละ 2 บิตพร้อม ๆ กันก็就会有การจัดตำแหน่งของสัญญาณตามที่แสดงไว้ในรูปที่ 2.49 (ข)

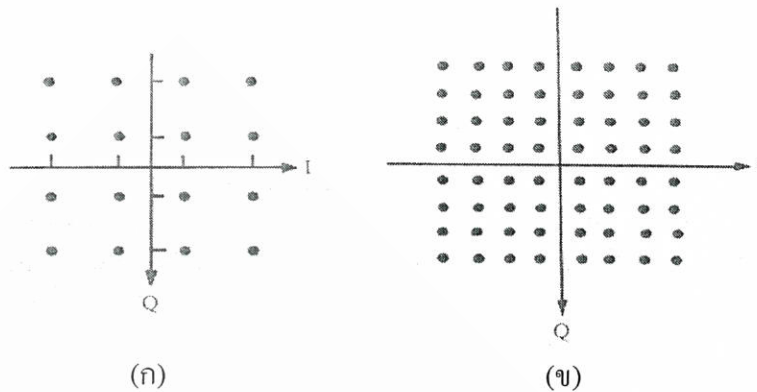
ในการหาความหนาแน่นสเปกตรัมกำลังของสัญญาณในสมการ 104 นี้ จะทำได้ดังนี้ ก่อนอื่นพิจารณาว่า $x_i(t)$ และ $x_q(t)$ ต่างก็มีรูปร่างของสัญญาณเหมือนกัน และมีความอิสระเชิงสถิติต่อกัน ดังนั้นความหนาแน่นสเปกตรัมกำลังของสัญญาณทั้งสองจึงเหมือนกันเราจะสามารถเขียนความหนาแน่นสเปกตรัมกำลังของสัญญาณ $x_i(t)$ และ $x_q(t)$ ได้ในรูปต่อไปนี้ $m_a = 0$, $\sigma_a^2 = 1$ และ

$$|x_i(f)|^2 = |x_q(f)|^2 = \frac{2}{R_B} \sin^2 \left(\frac{2\pi f}{R_B} \right) \quad (105)$$

และเนื่องจากเทอมที่หนึ่งและเทอมที่สองทางขวามือของสมการ 105 อยู่ในรูปของสัญญาณ DSB ดังนั้นเมื่อหาความหนาแน่นสเปกตรัมกำลังของสมการ 104 จะเขียนได้ในสมการ 106

$$G(f) = \frac{A_c^2}{R_B} \left[\sin^2 \left\{ \frac{2\pi(f - f_c)}{R_R} \right\} + \sin^2 \left\{ \frac{2\pi(f + f_c)}{R_R} \right\} \right] \quad (106)$$

เมื่อใช้หลักเกณฑ์ในการพิจารณาแบนด์วิดท์ที่ต้องการ จะพบว่าในกรณีนี้กำลังของสัญญาณส่วนใหญ่จะกระจายในช่วง $f_c \pm R_B/4$ ดังนั้นแบนด์วิดท์ที่ต้องการก็จะเป็น $B_T \approx R_B/2$ ซึ่งทำให้ประสิทธิภาพในการใช้แบนด์วิดท์เป็น 2 bps/Hz ประสิทธิภาพในการใช้แบนด์วิดท์นี้จะสูงกว่าสัญญาณ ASK แบบ 2 ระดับ 2 เท่าตัว ดังนั้นการมอดูเลตแบบ QAM นี้ถึงแม้จะยุ่งยากขึ้นกว่าแบบ ASK ก็ทำให้ประสิทธิภาพในการมอดูเลตสูงขึ้น ในกรณีที่สัญญาณดิจิทัลที่เข้ามามอดูเลตเป็นแบบ L ระดับ เช่น 4 ระดับ หรือ 8 ระดับ การจัดตำแหน่งของสัญญาณก็จะเป็นดังที่แสดงไว้ในรูปที่ 2.50 และเนื่องจากแบนด์วิดท์ของสัญญาณยังเท่าเดิมอยู่ ประสิทธิภาพในการใช้แบนด์วิดท์ก็จะสูงขึ้นเป็น 4 bps/Hz และ 6 bps/Hz ตามลำดับ



รูปที่ 2.50 การจัดตำแหน่งของสัญญาณ 16 QAM และ 64 QAM

2.3.12 การมอดูเลตแบบเปลี่ยนเฟสของสัญญาณคลื่นพาห้

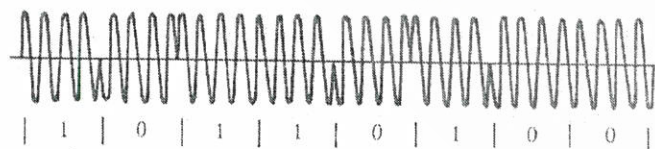
การมอดูเลตแบบเปลี่ยนเฟสของคลื่นพาห้หรือที่เรียกชื่อย่อว่า PSK (phase shift keying) นั้นเป็นการนำเอาสัญญาณดิจิทัลมาเปลี่ยนเฟสของสัญญาณคลื่นพาห้ ในกรณีของสัญญาณดิจิทัลแบบ M ระดับ เฟสของสัญญาณคลื่นพาห้ก็จะถูกแบ่งออกเป็น M ค่า เพื่อใช้แทนสัญญาณแต่ละระดับ สัญญาณที่ถูกมอดูเลตแล้วจะเขียนได้ในรูปต่อไปนี้

$$x_c(t) = A_c \sum_k \cos(\omega_c t + \phi_k) p(t - kD) \quad (107)$$

โดยที่

$$\phi_k = \frac{\pi(2a_k + 1)}{M} \quad a_k = 0, 1, 2, \dots, M-1 \quad (108)$$

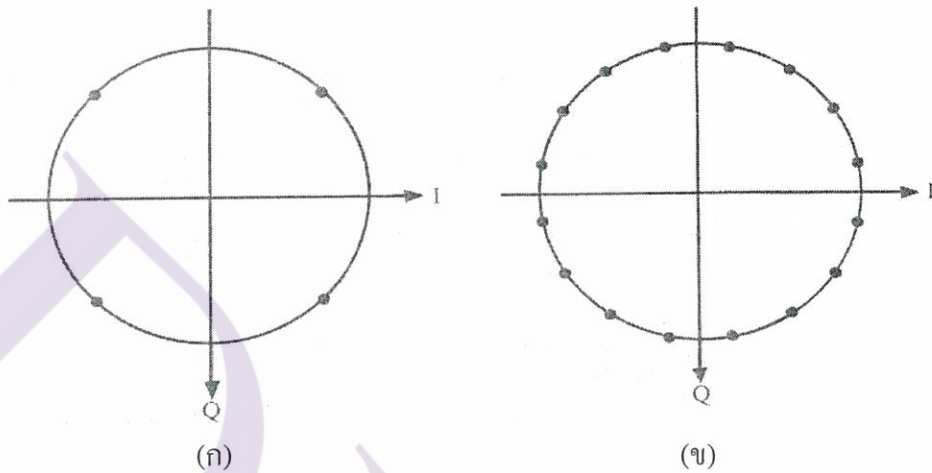
และ $p(t - kD)$ แสดงสัญญาณพัลส์ที่มีค่าเป็น 1 และกว้างเท่ากับ D ลักษณะของสัญญาณ PSK ในสมการที่ 107 แสดงว่าเฟสของสัญญาณจะเปลี่ยนแปลงไปตามระดับของสัญญาณดิจิทัล และขนาดของสัญญาณคลื่นพาห้จะไม่เปลี่ยนแปลง ในกรณีที่เป็นสัญญาณ 2 ระดับ เฟสของสัญญาณ 0 กับสัญญาณ 1 ก็จะอยู่ตรงข้ามกัน รูปที่ 2.51 แสดงลักษณะของรูปคลื่นของกรณีนี้



รูปที่ 2.51 รูปคลื่นของกรณีสัญญาณ PSK แบบ 2 ระดับ

ในกรณีของสัญญาณ 4 ระดับ หรือ 4-PSK นั้น การจัดตำแหน่งของสัญญาณนี้จะเป็นดังที่แสดงไว้ในรูปที่ 2.52 (ก) ซึ่งจะเห็นได้ว่าเหมือนกับของ 4-QAM ทุกประการ อย่างไรก็ตาม ในกรณีที่สัญญาณมากกว่า 4 ระดับ เช่น 16-PSK การจัดตำแหน่งสัญญาณจะไม่เหมือนกับของ 16-QAM ซึ่ง

จะเห็นได้จากการเปรียบเทียบรูปที่ 2.52 (ข) กับ รูปที่ 2.52 (ก) ทั้งนี้เป็นเพราะว่าสัญญาณ 16-QAM มีขนาดของสัญญาณคลื่นพาห์เปลี่ยนแปลงไปตามระดับของสัญญาณที่เข้ามามอดูเลต อนึ่งสัญญาณ 4-PSK นั้นมักจะถูกรเรียกว่า QPSK ซึ่งย่อมาจาก Quaternary PSK หรือ Quadriphase PSK



รูปที่ 2.52 การจัดตำแหน่งของสัญญาณ 4-PSK และ 16-PSK

และเมื่อทำการเขียนความหนาแน่นสเปกตรัมกำลังของสัญญาณ PSK ก็จะได้ผลดังนี้

$$G(f) = \frac{A_c^2}{4R} \left[\text{sinc}^2 \left\{ \frac{\pi(f - f_c)}{R} \right\} + \text{sinc}^2 \left\{ \frac{\pi(f + f_c)}{R} \right\} \right] \quad (109)$$

2.3.13 การมอดูเลตโดยการเปลี่ยนความถี่ของคลื่นพาห์

การมอดูเลตสัญญาณดิจิทัลโดยการเปลี่ยนความถี่ของสัญญาณคลื่นพาห์นั้น อาจจะทำให้ได้ 2 วิธีด้วยกันตามที่แสดงไว้ในรูปที่ 2.53 กล่าวคือ วิธีแรกเป็นการใช้สวิทช์ตอบสนองสัญญาณดิจิทัลแล้วทำการเลือกออสซิลเลเตอร์ที่มีความถี่ที่ต้องการเพื่อทำการส่งสัญญาณออกไป จำนวนออสซิลเลเตอร์ที่เตรียมไว้จะต้องเท่ากับจำนวนระดับของสัญญาณ การมอดูเลตวิธีนี้เฟสของสัญญาณแต่ละช่วงจะไม่ต่อเนื่องกันเพราะมาจากออสซิลเลเตอร์คนละชุดกัน วิธีนี้โดยทั่วไปเรียกว่า FSK (Frequency Shift Keying) สำหรับวิธีที่สองนั้นเป็นการมอดูเลตสัญญาณจากออสซิลเลเตอร์ตัวเดียวกัน เพราะฉะนั้นเฟสของสัญญาณจะต่อเนื่องบางครั้งจึงถูกเรียกว่า CPFSK (Continuous Phase Frequency Shift Keying) สัญญาณทั้งสองแบบนี้มีความยุ่งยากในการวิเคราะห์สเปกตรัม ในที่นี้จะพิจารณาแบบ CPFSK ซึ่งมีความยุ่งยากน้อยกว่าเป็นหลัก สัญญาณนี้ในอาณาจักรของเวลาจะเขียนได้ดังนี้

$$x_c(t) = A_c \sum_k \cos(\omega_c t + \theta + a_k \omega_d t) p(t - tD) \quad (110)$$

โดยที่ $a_k \omega_d$ เป็นความถี่ที่เลื่อนออกไปจากความถี่คลื่นพาห์ และ a_k แสดงระดับของสัญญาณดิจิทัล โดยมีค่าเป็น $\pm 1, \pm 3, \dots, \pm(M-1)$ นั่นคือ ความถี่ของสัญญาณที่อยู่ถัดจากกันจะห่างกัน $2f_d$ และถ้าต้องการให้สัญญาณ $x_c(t)$ มีเฟสต่อเนื่องกันตรงรอยต่อระหว่างสัญญาณก็จะทำได้โดยจัดให้ $2\omega_d D = 2\pi N$ โดยที่ N เป็นเลขจำนวนเต็ม



รูปที่ 2.53 การมอดูเลตสัญญาณดิจิทัล โดยการเปลี่ยนความถี่ของคลื่นพาห์

เนื่องจากการมอดูเลตแบบ FSK หรือ CPFSK นี้ ในกรณีที่สัญญาณดิจิทัลมีมากกว่า 2 ระดับ จะไม่มีที่ใช้น่ามากนัก ถ้าเปรียบเทียบกับ การมอดูเลตแบบ QAM หรือ PSK ดังนั้นในที่นี้จะกล่าวถึงการวิเคราะห์สเปกตรัมของกรณีสัญญาณดิจิทัล 2 ระดับ ถ้าให้อัตราข้อมูลเป็น R_B bps ตามเงื่อนไขที่จะให้เฟสต่อเนื่องกันระหว่างช่วงการเปลี่ยนสัญญาณ จะหา f_d ได้ดังนี้

$$f_d = \frac{1}{2D} = \frac{R_B}{2} \quad (111)$$

เมื่อความหนาแน่นสเปกตรัมกำลังของสัญญาณ $x_i(t)$ จะได้ผลดังนี้

$$|X_i(f)|^2 = \frac{1}{4} \left[\delta\left(f - \frac{R_B}{2}\right) + \delta\left(f + \frac{R_B}{2}\right) \right] \quad (112)$$

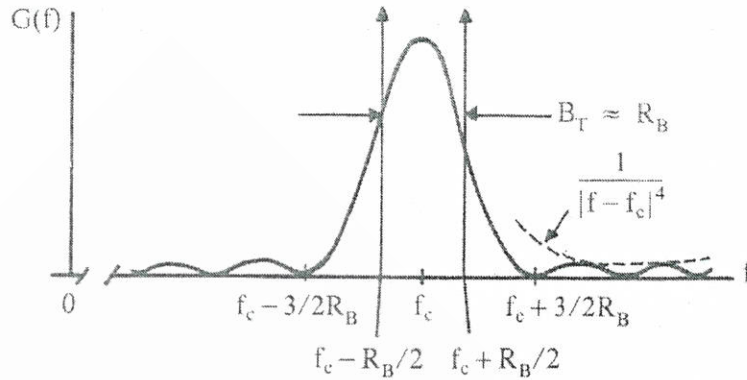
และจะสามารถเขียนความหนาแน่นสเปกตรัมกำลังของส่วนประกอบ $x_q(t)$ ได้ดังนี้

$$|X_q(f)|^2 = \frac{4}{\pi^2 R_B} \left[\frac{\cos(\pi f / R_B)}{(2f / R_B)^2 - 1} \right]^2 \quad (113)$$

จากสมการ 112 และสมการ 113 เราจะสามารถเขียนความหนาแน่นสเปกตรัมกำลังของสัญญาณรวมได้ในรูปต่อไปนี้

$$G(f) = \frac{A_c^2}{4} \frac{1}{4} \left[\delta\left(f - f_c - \frac{R_B}{2}\right) + \delta\left(f - f_c + \frac{R_B}{2}\right) + \delta\left(f + f_c - \frac{R_B}{2}\right) + \delta\left(f + f_c + \frac{R_B}{2}\right) \right] \quad (114)$$

$$+ \frac{A_c^2}{\pi^2 R_B} \left[\frac{\cos\{\pi(f - f_c)/R_B\}}{\{2(f - f_c)/R_B\}^2 - 1} \right]^2 + \frac{A_c^2}{\pi^2 R_B} \left[\frac{\cos\{\pi(f + f_c)/R_B\}}{\{2(f + f_c)/R_B\}^2 - 1} \right]^2$$



รูปที่ 2.54 ความหนาแน่นสเปกตรัมกำลังของสัญญาณ CPFSK

รูปที่ 2.54 แสดงในความสัมพันธ์ด้านบวกด้านเดียว ความหนาแน่นสเปกตรัมกำลังที่ได้นี้จะมีจุดศูนย์กลางแรกที่ตำแหน่ง $f_c \pm 3R_B/2$ เนื่องจากการลดต่ำลงของความหนาแน่นสเปกตรัมกำลังจะแปรตาม $1/|f - f_c|^4$ ดังนั้นแบนด์วิดท์เป็น $B_T \approx R_B$

2.3.14 การดิมอดูเลตสัญญาณแบบเอ็นเวลโลปดิเทกชัน

การดิมอดูเลตสัญญาณแบบเอ็นเวลโลปดิเทกชันในกรณีของสัญญาณดิจิทัลนั้น โดยมีหลักการก็จะเหมือนกับกรณีของสัญญาณอนาล็อก กล่าวคือ สัญญาณรบกวนที่ผ่านภาคดิมอดูเลตออกไปได้จะเป็นสัญญาณรบกวนแบนด์วิดท์แคบซึ่งมีทั้งส่วนประกอบอินเฟส และควอดเรเจอร์เฟส ถ้าให้สัญญาณคลื่นพาห้สามารถเขียนได้เป็น $A_c \cos \omega_c t$ และสัญญาณรบกวนเขียนได้เป็น $n(t) = n_i(t) \cos \omega_c t - n_q(t) \sin \omega_c t$ ผลรวมของสองสัญญาณนี้เมื่อพิจารณาในรูปของขนาดและเฟสจะเขียนได้ดังนี้

$$x_R(t) = A_c \cos \omega_c t + n(t) = A(t) \cos[\omega_c t + \phi(t)] \quad (115)$$

โดยที่

$$A(t) = \sqrt{\{A_c + n_i(t)\}^2 + n_q^2(t)}$$

$$\phi(t) = \arctan \frac{n_q(t)}{A_c + n_i(t)}$$

เนื่องจาก n_i และ n_q เป็นตัวแปรแรนดัมที่เป็นอิสระจากกัน และมี PDF แบบเกาส์เซียนทั้งคู่ ดังนั้นในกรณีที่ $A_c = 0$ $A(t)$ จะเป็นเอ็นเวลโลปของสัญญาณรบกวน PDF ของสัญญาณรบกวนนี้จะมีการกระจายแบบเรย์เลห์ ซึ่งจะเขียนได้ในสมการ 116

$$p_{A_n}(A_n) = \frac{A_n}{\sigma^2} e^{-A_n^2/2\sigma^2} \quad A_n \geq 0 \quad (116)$$

ในกรณีที่ $A_c \gg \sigma$ $A(t)$ ก็จะประมาณค่าได้เป็น

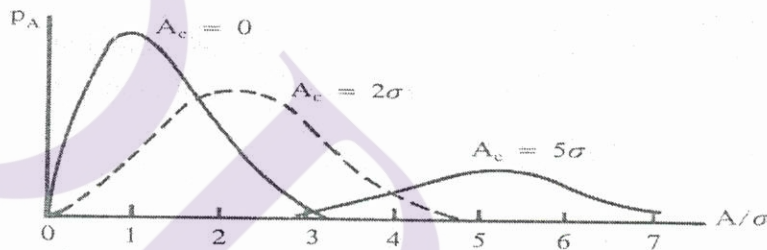
$$A(t) \approx A_c + n_i \quad (117)$$

ซึ่งก็หมายถึง A จะมี PDF กระจายอยู่ในรูปเดียวกัน n_i คืออยู่ในรูปเกาส์เซียน

ดังนั้น PDF ที่มีการกระจายแบบไรเซียน (Rician distribution) ในกรณีที่ $A_c \gg \sigma$ จะเขียนได้ดังนี้

$$p_A(A) \approx \sqrt{\frac{A}{2\pi A_c \sigma^2}} e^{-(A-A_c)^2/2\sigma^2} \quad (118)$$

และเนื่องจากในเงื่อนไขที่ $A_c \gg \sigma$ นี้ เพราะฉะนั้น $p_A(A)$ ในสมการ 118 ก็จะมีการกระจายอยู่ในรูปเกาส์เซียนโดยที่มีค่าแวกเรียนซ์เป็น σ^2 รูปที่ 2.55 แสดงผลการคำนวณ $p_A(A)$ ในกรณีที่ A_c มีค่าต่าง ๆ ซึ่งจะเห็นได้ว่ากรณีที่ $A_c = 0$, $p_A(A)$ จะเป็นสมการที่ 37 ซึ่งมีการกระจายแบบเรย์เลห์ และ $p_A(A)$ จะค่อย ๆ เปลี่ยนไปเข้าใกล้การกระจายแบบเกาส์เซียนเมื่อ A_c มีค่าสูงขึ้นเมื่อเทียบกับ σ



รูปที่ 2.55 PDF ของสัญญาณคลื่นพาห่ที่มีสัญญาณรบกวนผสมอยู่

2.3.15 การดีเทคสัญญาณไบนารีโดยวิธีเอนเวลโลปดีเทคชัน

(1) กรณีของสัญญาณไบนารีแบบ OOK

ในกรณี $(k+1)D$ จะเขียนได้ในรูปต่อไปนี้

$$x_c(t) = A_c a_k p(t-kD) \cos(\omega_c t + \theta) \quad (119)$$

โดยที่ a_k เป็น 1 หรือ 0 ตามสัญญาณดิจิทัลที่เป็น “1” หรือ “0” เมื่อทำการคำนวณพลังงานที่อยู่ในบิต “0” และบิต “1” จะได้ผลดังนี้

$$E_0 = 0 \quad (120)$$

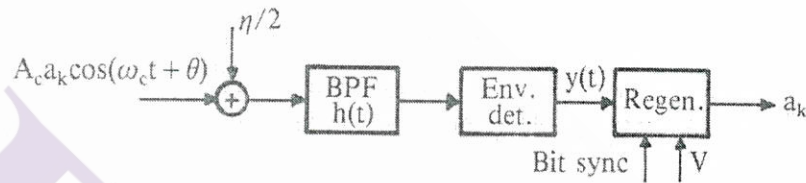
$$E_1 = \frac{A_c^2 D}{2} \quad (121)$$

และเมื่อทำการคำนวณค่าพลังงานเฉลี่ยต่อ 1 บิต ก็จะได้

$$E_b = \frac{E_1}{2} = \frac{A_c^2 D}{4} \quad (122)$$

รูปที่ 2.56 แสดงบล็อกไดอะแกรมของการตีเทคสัญญาณแบบเอ็นเวลโลปตีเทคชัน ส่วนที่เป็นฟิลเตอร์ผ่านแถบความถี่นี้จะพิจารณาในกรณีที่เป็นแมทซ์ฟิลเตอร์ซึ่งจะทำให้การตีเทคได้ผลดีที่สุดจากหลักการของแมทซ์ฟิลเตอร์ผลตอบสนองอิมพัลส์ของแมทซ์ฟิลเตอร์ในกรณีจะเขียนได้เป็น

$$h(t) = KA_c p(t) \cos \omega_c t \quad (123)$$



รูปที่ 2.56 บล็อกไดอะแกรมของการทำเอ็นเวลโลปตีเทคชัน

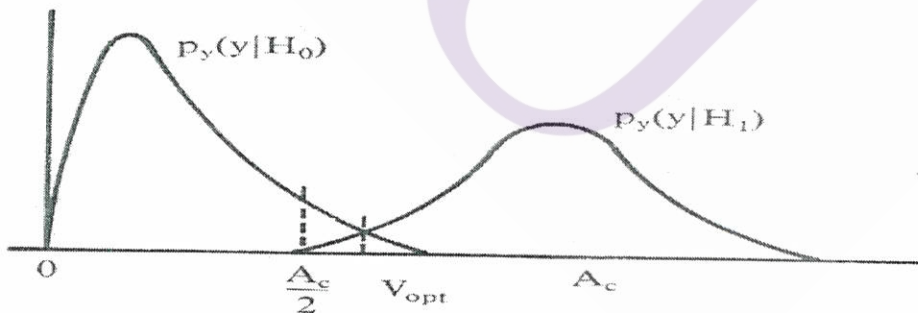
สัญญาณที่ผ่านออกจากแมทซ์ฟิลเตอร์ ดังนี้

$$y(t) = \frac{KA_c^2 D}{2} \left[1 - \left| \frac{t-D}{D} \right| \right] \cos(\omega_c t + \theta) \quad (124)$$

โดยที่ t อยู่ในช่วง 0 ถึง $2D$ ซึ่งตำแหน่งที่ค่าเอ็นเวลโลปสูงสุดก็คือ $t = D$ เพื่อให้สะดวกในการคิดจะให้ค่าคงที่ K ในสมการที่ 123 และสมการที่ 124 มีค่าเป็น $K = A_c/E_1$ ทำให้ได้อัตราส่วนระหว่าง A_c^2 กับ σ^2 ในรูปต่อไปนี้

$$\frac{A_c^2}{\sigma^2} = \frac{2E_1}{\eta} = \frac{4E_b}{\eta} \quad (125)$$

จากรูปที่ 2.57 จะเห็นได้ว่าระดับเทรชโฮลด์เหมาะสมที่สุดก็คือ จุดตัดระหว่าง $p_y(y|H_0)$ และ $p_y(y|H_1)$ ถ้าพิจารณาในกรณีที่ $A_c^2/\sigma^2 \gg 1$



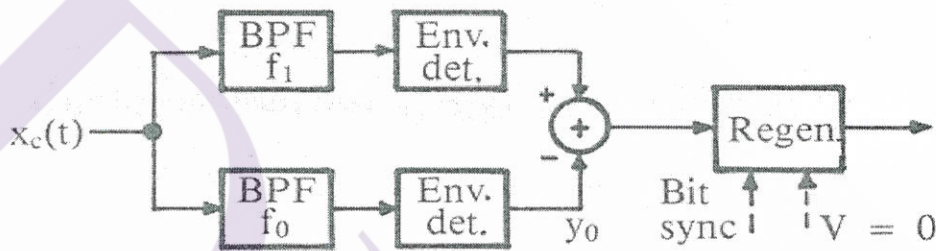
รูปที่ 2.57 PDF ของ $p_y(y|H_0)$ และ $p_y(y|H_1)$ ในกรณีของเอ็นเวลโลปตีเทคชัน

และเนื่องจาก $p_y(y|H_0)$ และ $p_y(y|H_1)$ มีรูปร่างไม่เหมือนกัน เพราะฉะนั้นพروبอบะบิลิตีของการผิดพลาดผลรวมก็จะเขียนได้ดังนี้

$$P_c = \frac{1}{2} e^{-E_b/2\eta}, \quad \frac{E_b}{\eta} \gg 1 \quad (126)$$

(2) กรณีของสัญญาณไบนารีแบบ FSK

ในกรณีของสัญญาณไบนารีแบบ FSK นั้น สรุปได้ว่าไม่สามารถใช้วิธีเอ็นเวลโลปดีเทคชันได้ เพราะขนาดของสัญญาณคลื่นพาห้ไม่เปลี่ยนแปลง อย่งไรก็ตามสัญญาณ FSK เป็นสัญญาณที่มีความถี่ของคลื่นพาห้ต่างกันอยู่ ดังนั้นเราจะสามารถใช้วิธีการทำเอ็นเวลโลปดีเทคชันสำหรับแต่ละความถี่ได้ ถ้าให้คลื่นพาห้ของสัญญาณ “1” และสัญญาณ “0” เป็น f_1 และ f_0 ตามลำดับ



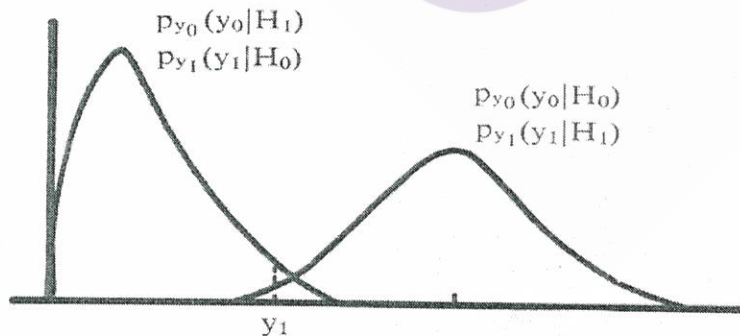
รูปที่ 2.58 บล็อกไดอะแกรมของการดีเทคสัญญาณ FSK แบบเอ็นเวลโลปดีเทคชัน

ฟิลเตอร์ในกรณีนี้จะต้องมีผลตอบสนองอิมพัลส์เป็นดังนี้

$$h_1(t) = KA_c p(t) \cos \omega_1 t \quad (127)$$

$$h_0(t) = KA_c p(t) \cos \omega_0 t$$

ในกรณีนี้เราจะได้พลังงานในบิต “1” และบิต “0” มีขนาดเท่ากันและเท่ากับพลังงานเฉลี่ยต่อบิตคือ $E_b = E_1 = E_0 = A_c^2 D/2$ และเมื่อให้ $K = A_c/E_b$ เราจะสามารถหาค่า A_c^2/σ^2 จะได้รูปที่ 2.58 เราจะสรุปได้ว่าระดับเทรซโซลด์ที่เหมาะสมนั้นก็คือ $V_{opt} = 0$ โดยไม่ขึ้นอยู่กับ A_c ในสภาพที่มีสัญญาณรบกวนปะปนอยู่นั้นเราสามารถหาขอบเขตบิติตีของการผิดพลาดโดย

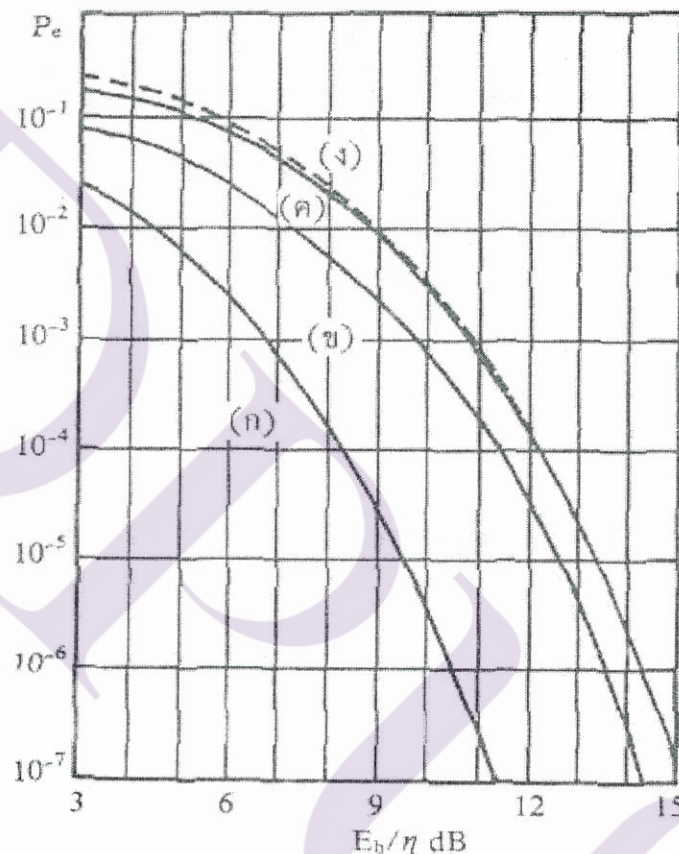


รูปที่ 2.59 PDF ของสัญญาณ y_0 และ y_1

ดังนั้น

$$P_e = \frac{1}{2} e^{-A_c^2/4\sigma^2} = \frac{1}{2} e^{-E_b/2\eta} \quad (128)$$

รูปที่ 2.60 แสดงผลการคำนวณ P_e ของสัญญาณไบนารีแบบต่าง ๆ เปรียบเทียบกันจะเห็นได้ว่าการดีเทคสัญญาณแบบเอนเวลโกลบดีเทคชันจะมีคุณสมบัติสู้แบบโคฮีเรนท์ดีเทคชันไม่ได้

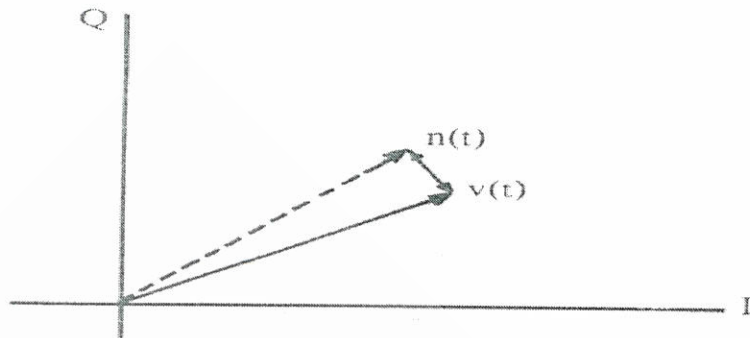


รูปที่ 2.60 พรอบประสิทธิภาพของการผิดพลาดของสัญญาณไบนารีที่มีการดีเทคแบบต่าง ๆ

(ก) coherent PSK (ข) coherent OOK (ค) noncoherent FSK (ง) noncoherent OOK

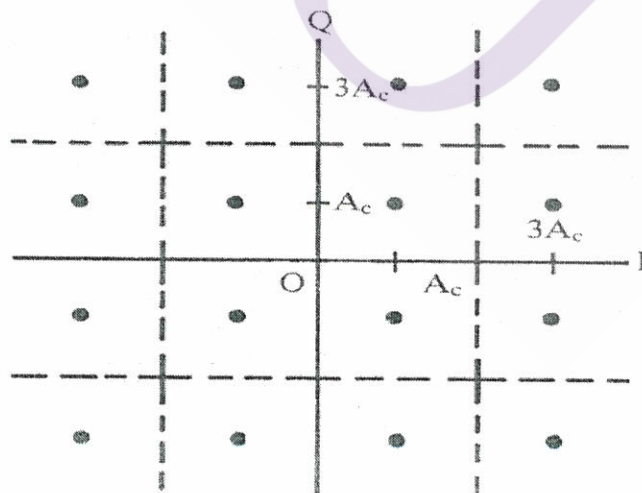
2.3.16 การดีเทคสัญญาณ M-QAM และสัญญาณ M-PSK

ในการพิจารณาผลกระทบของสัญญาณรบกวน ซึ่งเราจะใช้การแสดงตำแหน่งของสัญญาณด้วยเฟสเซอร์เวกเตอร์ (phasor vector) ในขณะเดียวกันก็จะพิจารณาว่าสัญญาณรบกวนก็จะเป็นเวกเตอร์ที่ไปบวกกับเฟสเซอร์เวกเตอร์ของสัญญาณ รูปที่ 2.61 แสดงภาพได้ดังนี้

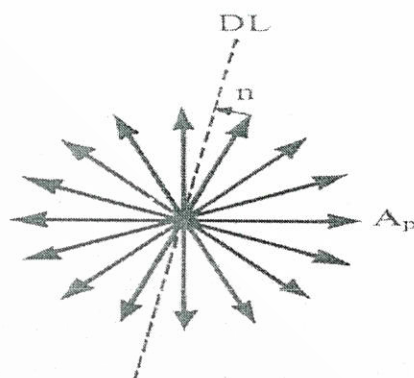


รูปที่ 2.61 เวกเตอร์ของสัญญาณและเวกเตอร์ของสัญญาณรบกวน

เราสามารถแบ่งอาณาเขต ระหว่างการให้สัญญาณแต่ละสัญญาณสามารถทนต่อขนาดของสัญญาณรบกวนได้เท่า ๆ กัน เพื่อให้พروبอบะบิลิตีของการผิดพลาดของสัญญาณต่ำที่สุด รูปที่ 2.62 แสดงการแบ่งอาณาเขตของสัญญาณ 16-QAM โดยใช้หลักการที่กล่าวมานี้ เมื่อสัญญาณที่รับมาได้ตกอยู่ในอาณาเขตของสัญลักษณ์ใด ระดับก็จะตัดสินใจให้เป็นสัญลักษณ์ที่เป็นเจ้าของอาณาเขตนั้น ตามรูปที่ 2.62 นี้สัญญาณแต่ละสัญลักษณ์จะทนต่อสัญญาณรบกวนที่มีขนาดเป็น A_c ได้ เมื่อสัญญาณรบกวนในจังหวะที่ทำการสุ่มตัวอย่างมีขนาดใหญ่กว่าค่า A_c ก็จะทำให้เกิดการผิดพลาดของสัญลักษณ์เกิดขึ้น จากการแบ่งอาณาเขตตามที่แสดงนี้จะทำให้พروبอบะบิลิตีของการผิดพลาดของสัญลักษณ์ที่อยู่ข้างเคียงกัน ซึ่งมีขนาดเท่ากันซึ่งเป็นการทำให้พروبอบะบิลิตีของการผิดพลาดของสัญลักษณ์โดยผลรวมมีขนาดต่ำที่สุด รูปที่ 2.63 เป็นรูปที่แสดงการแบ่งอาณาเขตของสัญลักษณ์ในระบบ 16-PSK ซึ่งสัญลักษณ์แต่ละสัญลักษณ์ก็จะทนต่อสัญญาณรบกวนขนาดเท่ากัน และเท่ากับ $A_p \times \pi/16$ และเมื่อแบ่งอาณาเขตของสัญลักษณ์ตามนี้แล้วพروبอบะบิลิตีของสัญลักษณ์โดยรวมก็จะมีค่าต่ำที่สุด



รูปที่ 2.62 การแบ่งอาณาเขตของสัญลักษณ์ในระบบ 16-QAM



รูปที่ 2.63 การแบ่งอาณาเขตของสัญลักษณ์ในระบบ 16-PSK

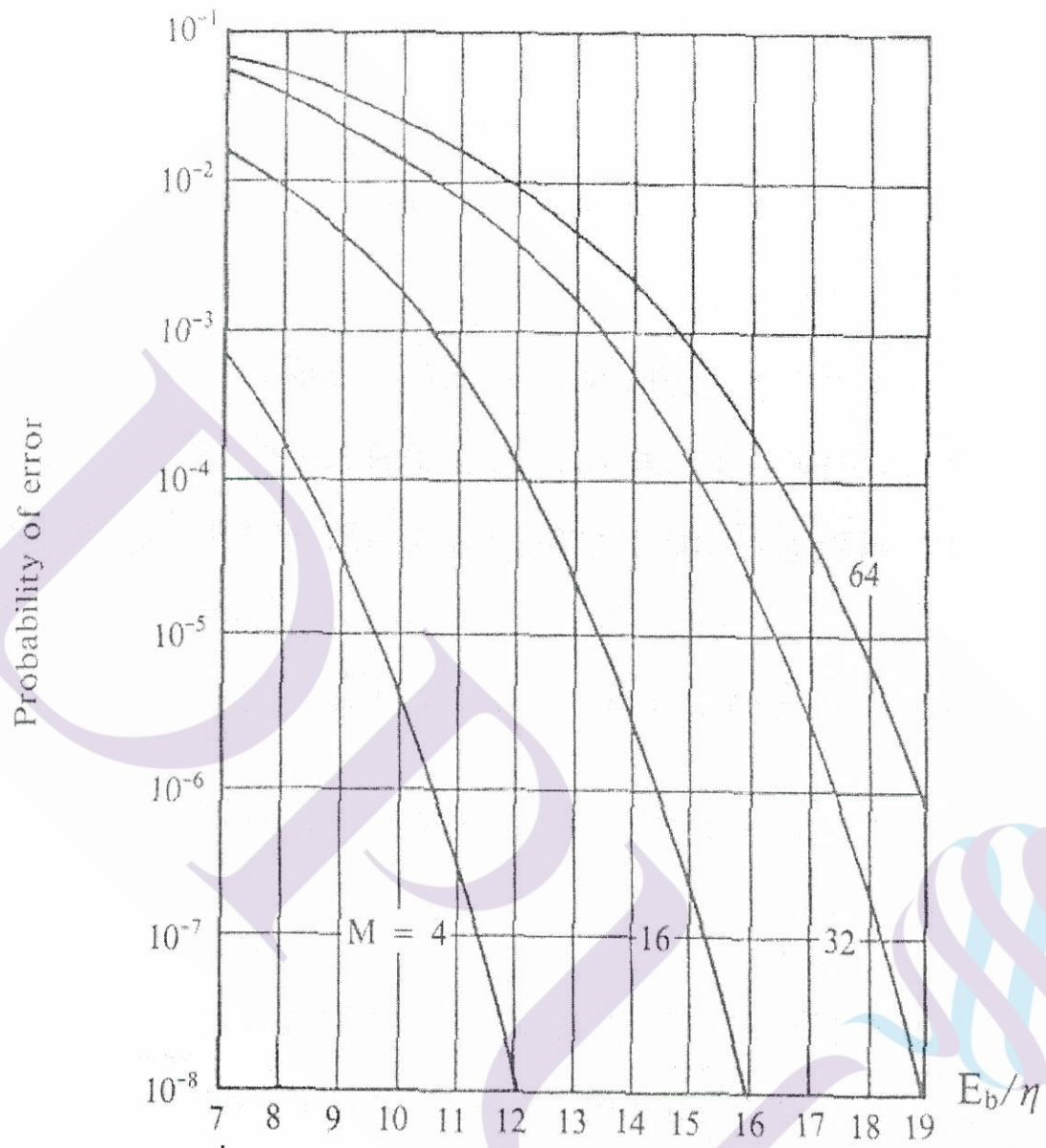
2.3.17 พหุคูณบิตของการผิดพลาดของสัญลักษณ์ในระบบ M-QAM และ ระบบ M-PSK

ในการพิจารณาพหุคูณบิตของการผิดพลาดของสัญญาณแบบ M-QAM และแบบ M-PSK ผลที่ได้สำหรับทั้งสองกรณีดังต่อไปนี้

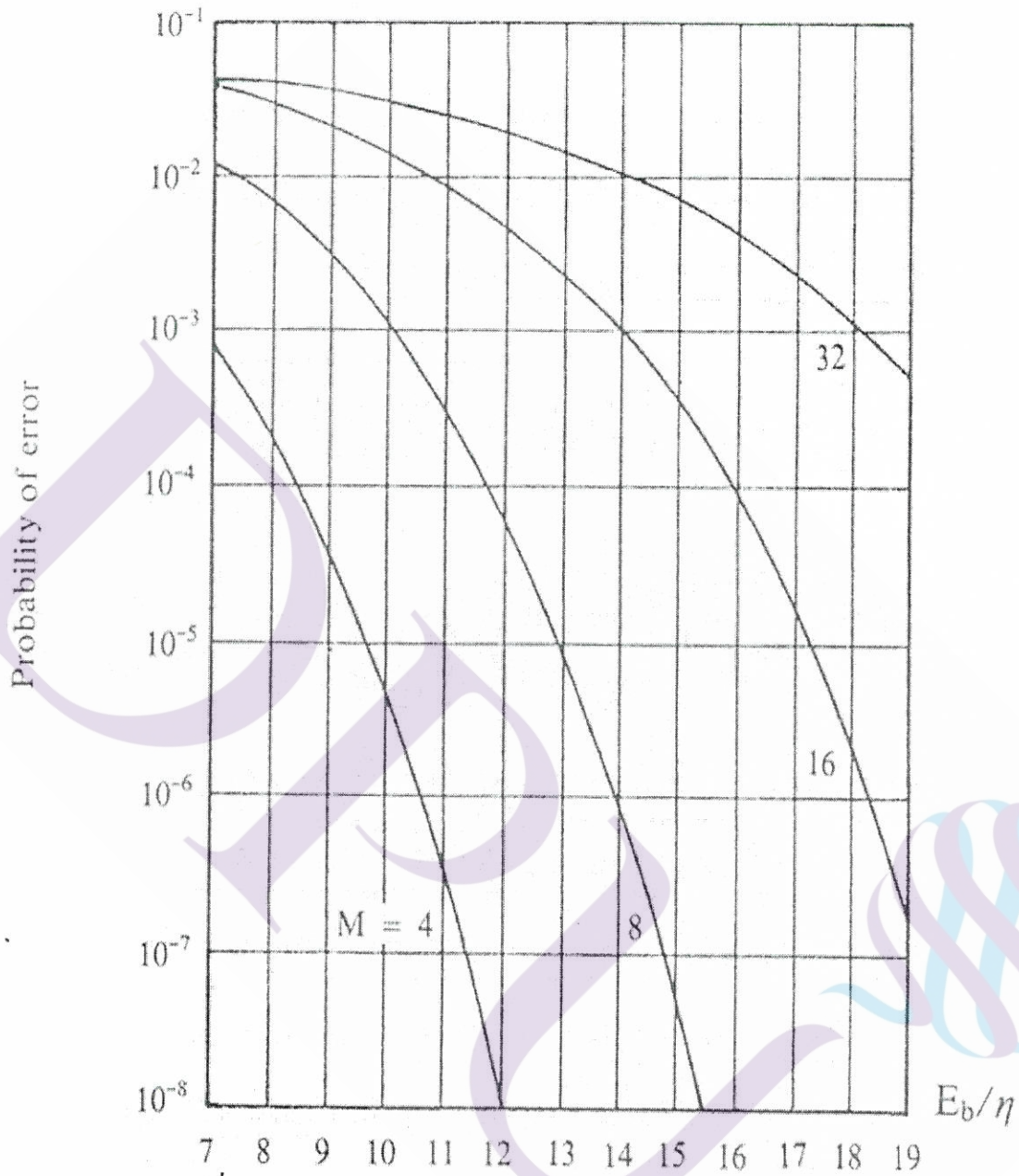
$$P_{eQ} \approx \frac{2}{\log_2 L} \left(1 - \frac{1}{L}\right) Q \left[\sqrt{\left(\frac{3 \log_2 L}{L^2 - 1}\right) \frac{2E_b}{\eta}} \right] \quad (129)$$

$$P_{eP} \approx 2Q \sqrt{\frac{2E_b \log_2 M}{\eta}} \sin \frac{\pi}{M} \quad (130)$$

โดยที่ P_{eQ} และ P_{eP} แสดงพหุคูณบิตของการผิดพลาดของ M-QAM และ M-PSK ตามลำดับ และ L ในสมการ 129 คือจำนวนระดับของสัญญาณ QAM รูปที่ 2.64 และรูปที่ 2.65 แสดงผลการคำนวณสมการ 129 และสมการ 130 เมื่อเปรียบเทียบรูปทั้งสองรูป จะเห็นได้ว่ากรณีของ 4-QAM และ 4-PSK จะเหมือนกันทุกประการ แต่เมื่อ $M > 4$ แบบ M-QAM จะมีคุณสมบัติดีกว่าแบบ M-PSK อย่างน่าสังเกต ในกรณีที่รู้พลังงานเฉลี่ยต่อ 1 สัญลักษณ์ ก็จะสามารถหา 1 บิตได้จาก $E_b = E_s / \log_2 M$ โดยที่ E_s เป็นพลังงานเฉลี่ยต่อ 1 สัญลักษณ์ ในกรณีที่รู้พลังงานสูงสุดของสัญลักษณ์ ก็จะต้องหาค่าพลังงานเฉลี่ยต่อ 1 สัญลักษณ์ออกมาก่อนจึงนำมาหา E_b ต่อไป



รูปที่ 2.64 พรอบบะบิลิตีของการผิดพลาดของสัญญาณ M-QAM



รูปที่ 2.65 พรอบบะบิลิตีของการผิดพลาดของสัญญาณ M-PSK

เมื่อลองอ่านค่าพรอบบะบิลิตีของการผิดพลาดจากรูปที่สองโดยให้ $E_b/\eta = 16$ dB ในกรณี
 ที่ $M = 16$ $P_{eQ} \approx 1.0 \times 10^{-8}$ ในขณะที่ $M = 8$ $P_{eP} = 8.0 \times 10^{-5}$ ซึ่งจะเห็นได้ว่าค่าต่างกัน
 อย่างน่าสังเกต ดังนั้นในกรณี M มีระดับสูงจึงมักนิยมใช้สัญญาณแบบ M-QAM มากกว่า

2.3.18 ความจุของช่องสัญญาณแบบต่อเนื่อง

เพื่อให้สามารถทำการแสดงการหาค่าความจุของช่องสัญญาณนั้นจะพิจารณาช่องสัญญาณแบบที่มีสัญญาณรบกวนแบบแอดดิทีฟไวท์เกาส์เซียน (additive white gaussian noise ย่อว่า AWGN) ซึ่งช่องสัญญาณแบบนี้จะมีคุณสมบัติดังนี้

1. สัญญาณกับสัญญาณรบกวนไม่มีคอร์รีเลชันกัน กล่าวคือถ้า $y(t) = x(t) + n(t)$ จะได้ $\overline{y^2} = \overline{x^2} + \overline{n^2}$
2. สัญญาณรบกวนเป็นแบบไวท์เกาส์เซียนซึ่งมีค่าเฉลี่ยของ $n(t)$ เป็นศูนย์ และกำลังของสัญญาณรบกวนเป็น $\overline{n^2} = N = \eta B$ โดย B เป็นแบนด์วิดท์ของช่องสัญญาณ
3. กำลังของสัญญาณอินพุตมีขนาดจำกัด $\overline{x^2} \leq S$
4. การผิดเพี้ยน และการสูญเสียกำลังของสัญญาณระหว่างทางจึงให้สามารถชดเชยได้จากวงจรทางภาค front

2.4 ทฤษฎีเบื้องต้นของระบบการสื่อสารแบบสเปกตรัม

2.4.1 ความจุช่องสัญญาณ (Channel Capacity, C)

ปกติช่องสัญญาณจะแปลงสัญญาณขาเข้าชุดหนึ่งให้เป็นสัญญาณขาออกชุดหนึ่ง โดยที่การเกิดสัญญาณขาออกแต่ละชุดนั้นสามารถแสดงได้ โดยอาศัยความรู้ด้านความน่าจะเป็นแบบมีเงื่อนไข ความจุช่องสัญญาณมีนิยามคือ เป็นความสามารถในการส่งผ่านสัญญาณ โดยทำให้ปริมาณข่าวสารร่วมเฉลี่ยหรือ $I(X; Y)$ สูงที่สุดโดยไม่มีผลกระทบเกิดขึ้น ดังนั้นจึงสามารถเขียนได้ในคณิตศาสตร์ในรูปต่อไปนี้

$$C = \max_{p(x)} I(X; Y) \quad (131)$$

โดยที่ $I(X; Y)$ จะแสดงความสัมพันธ์ระหว่าง X (ช่องสัญญาณขาเข้า) และ Y (ช่องสัญญาณขาออก) ดังสมการต่อไปนี้

$$I(X; Y) = \sum_{x \in X} \sum_{y \in Y} p(x)p(y/x) \log \left(\frac{p(x, y)}{p(x)p(y)} \right) \quad (132)$$

โดยที่ค่า x คือ ความน่าจะเป็น crossover ของช่องสัญญาณ X

y คือ ความน่าจะเป็น crossover ของช่องสัญญาณ Y

$p(x)$ คือ ความน่าจะเป็นของ x

$p(y)$ คือ ความน่าจะเป็นของ y

$p(y/x)$ คือ ความน่าจะเป็นของ y ใน x

$p(x, y)$ คือ จอยนต์ความน่าจะเป็นร่วมของ X และ Y

จะสามารถแสดงค่าความจุได้เป็น

$$C = 1 - H_b(\epsilon) \quad (133)$$

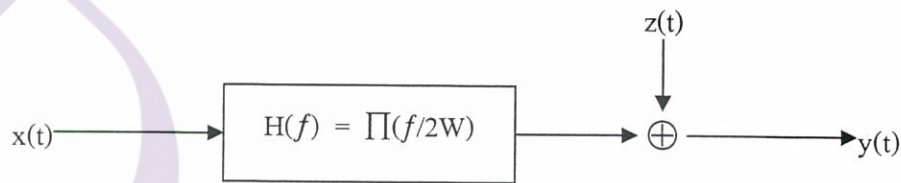
โดยที่ค่า ϵ คือ ความน่าจะเป็น crossover ของช่องสัญญาณ

ค่า $H_b(\bullet)$ คือ binary entropy function

ซึ่งค่า $H_b(\bullet)$ สามารถแสดงได้ดังนี้

$$H_b(x) = -x \log(x) - (1-x) \log(1-x) \quad (134)$$

ส่วนแบบจำลองของช่องสัญญาณสามารถเขียนได้ดังรูปที่ 2.1-1



รูปที่ 2.1-1 แบบจำลองของช่องสัญญาณ

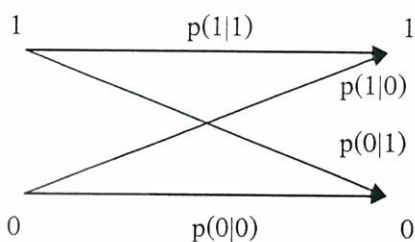
จากรูปที่ 2.1-1 เมื่อช่องสัญญาณมีช่วงลิมิตของความถี่เป็น $[-W, W]$ และมีสัญญาณรบกวนไวท์เกาส์เซียน ด้วยกำลังขาเข้า P ซึ่งสามารถแสดงค่า C ดังสมการต่อไปนี้

$$C = W \log \left(1 + \frac{P}{N_o W} \right) \quad \text{บิต/วินาที} \quad (135)$$

และเมื่อให้ค่าความแปรปรวนของสัญญาณรบกวนเป็น σ^2 จะได้ค่า

$$C = \frac{1}{2} \log \left(1 + \frac{P}{\sigma^2} \right) \quad (136)$$

จากรูปที่ 2.1-2 แสดงตัวอย่างช่องสัญญาณที่ส่งสัญญาณดิจิทัล “1” กับ “0” โดยในช่องสัญญาณนั้นจะมีสัญญาณรบกวนอยู่ ซึ่งสัญญาณรบกวนนั้นอาจจะทำให้สัญญาณขาเข้าที่เข้ามาถูกแปลงไปเป็นสัญญาณที่มีค่า “0” หรือ “1” เช่น สัญญาณขาเข้าที่เข้ามาเป็น “1” ช่องสัญญาณจะทำการแปลงให้เป็นสัญญาณเอาต์พุตที่มีค่าเป็น “1” ด้วยความน่าจะเป็นแบบมีเงื่อนไข $p(1|1)$ และสัญญาณขาออกที่มีค่าเป็น “0” เกิดขึ้นด้วยความน่าจะเป็นแบบมีเงื่อนไข $p(0|1)$



รูปที่ 2.1-2 ตัวอย่างช่องสัญญาณที่ส่งสัญญาณดิจิทัล “1” กับ “0”

ในทำนองเดียวกันถ้าสัญญาณจะเข้าเป็น “0” สัญญาณขาออกที่ได้จะเป็น “1” และ “0” ด้วยความน่าจะเป็น $p(1|0)$ และ $p(0|0)$ ตามลำดับทั้งนี้ ความน่าจะเป็น $p(0|1)$ และ $p(1|0)$ นั้นจะเป็นความน่าจะเป็นของความผิดพลาด และถ้า $p(0|1) = p(1|0)$ ช่องสัญญาณนั้นก็จะเป็น “ช่องสัญญาณแบบสมมาตร (symmetric channel)”

ในกรณีที่สัญญาณขาเข้าและสัญญาณขาออกเป็นสัญญาณแบบไม่ต่อเนื่อง ช่องสัญญาณนั้นก็จะถูกเรียกว่า “ช่องสัญญาณแบบไม่ต่อเนื่อง (discrete channel)” หากสัญญาณขาเข้าและสัญญาณขาออกเป็นสัญญาณแบบต่อเนื่อง ช่องสัญญาณนั้นก็จะถูกเรียกว่า “ช่องสัญญาณแบบต่อเนื่อง (continuous channel)” นอกจากนี้คุณสมบัติที่สำคัญอีกประการหนึ่งของช่องสัญญาณ คือ คุณสมบัติแบบมีหน่วยความจำ (with memory) และไม่มีหน่วยความจำ (memoryless) ซึ่งไม่ได้อธิบายไว้ในที่นี้

การหาค่าความน่าจะเป็นสามารถทำได้จากตัวอย่างดังต่อไปนี้ เช่น เมื่อเราทำการทอดลูกเต๋าจำนวนมา ๆ ครั้ง และทำการสังเกตโอกาสที่จะเกิดหน้า “1” ซึ่งในที่สุดจะลู่อเข้าหาตัวเลข $1/6$ ดังนั้นโอกาสที่จะเกิดหรือความน่าจะเป็นนั้น จึงเป็นการพิจารณาจากโอกาสที่จะเกิดเหตุการณ์นั้น ซึ่งจากการทดลองดูจำนวนมา ๆ ครั้ง กล่าวคือถ้าให้เหตุการณ์นั้นเป็น A และเมื่อทำการทดลองเป็นจำนวน N ครั้ง ปรากฏว่าเกิด A เป็นจำนวน N_A ครั้ง ความน่าจะเป็นที่จะเกิด A จะเขียนได้ดังนี้

$$P(A) = \lim_{N \rightarrow \infty} \frac{N_A}{N} \quad (137)$$

จากคำอธิบายที่กล่าวมาทั้งหมดนี้จะเห็นได้ว่าค่าความน่าจะเป็นจะมีค่าเป็นบวก และมีค่าอยู่ในช่วง 0 ถึง 1 เสมอ ซึ่ง 0 หมายถึง ไม่มีโอกาสเกิดเหตุการณ์นั้นเลย และ 1 หมายถึง จะเกิดเหตุการณ์นั้นเพียงอย่างเดียวเท่านั้น

การพิจารณาช่องสัญญาณที่มีการรบกวนแบบแอดดิทีฟไวท์เกาส์เซียน (Additive White Gaussian Noise ย่อว่า AWGN) ซึ่งช่องสัญญาณแบบนี้จะมีคุณสมบัติดังนี้

1. สัญญาณกับสัญญาณรบกวนไม่มีคอร์ริชัน (Correlation) กัน กล่าวคือถ้า

$$y(t) = x(t) + n(t) \text{ จะได้ } \overline{y^2} = \overline{x^2} + \overline{n^2}$$

2. สัญญาณรบกวนเป็นแบบไวท์เกาส์เซียนซึ่งมีค่าเฉลี่ยของ $n(t)$ เป็นศูนย์ และกำลังของสัญญาณรบกวนเป็น $\overline{n^2} = N = \eta B$ โดยที่ B เป็นแบนด์วิดท์ของช่องสัญญาณ
3. กำลังของสัญญาณอินพุตมีขนาดจำกัด $\overline{x^2} \leq S$
4. การผิดเพี้ยนและการสูญเสียกำลังของสัญญาณระหว่างทางสามารถชดเชยได้ โดยวงจรทางภาครับ

2.4.2 การเข้ารหัสช่องสัญญาณ (Channel Coding)

เนื่องจากการเข้ารหัสช่องสัญญาณเป็นการนำเอาคำรหัสที่ได้จากเครื่องเข้ารหัสแหล่งกำเนิดข่าวสารมาจัดทำเป็นสัญญาณชุดใหม่เพื่อส่งเข้าช่องสัญญาณต่อไป ในกรณีที่ต้องประกอบข่าวสารของแหล่งกำเนิดข่าวสารมีอยู่ M แบบและแต่ละแบบมีความน่าจะเป็นของการเกิดเท่า ๆ กัน คำรหัสที่ออกจากเครื่องเข้ารหัสข่าวสารก็จะมีขนาดยาวเท่ากันคือเท่ากับ $\log_2 M$ บิต และมีโอกาสเกิดขึ้นได้เท่า ๆ กัน ในกรณีเช่นนี้ ถ้าเราให้คำรหัสที่ผ่านออกจากเครื่องเข้ารหัสช่องสัญญาณมีความยาวเป็น N สัญลักษณ์อัตราข่าวสาร (R) ที่ส่งได้ต่อหนึ่งสัญลักษณ์

$$R = \frac{\log_2 M}{N} \quad \text{บิต/สัญลักษณ์} \quad (138)$$

ซึ่งในระบบเข้ารหัสและถอดรหัสช่องสัญญาณจะแบ่งกว้าง ๆ ได้เป็น 2 แบบด้วยกัน คือ การเข้ารหัสแบบบล็อก (block encoding) และการเข้ารหัสแบบลำดับ (sequential encoding) โดยที่การทำงานของระบบเข้ารหัสทั้งสองก็คือ ถ้าเครื่องเข้ารหัสช่องสัญญาณอ่านสัญญาณไบนารีเข้ามาเป็นจำนวน μ บิต ในช่วงระหว่างการส่งคำรหัสออกไปแต่ละชุด ถ้า $\mu = v$ ก็จะเป็นการเข้ารหัสแบบบล็อก และถ้าค่า $\mu < v$ ก็จะเป็นการเข้ารหัสแบบเป็นลำดับ ซึ่งการเข้ารหัสแบบบล็อกนั้นก็จะเป็นการอ่านและการเก็บจำนวนสัญญาณไบนารีให้ครบ 1 บล็อกก่อนจากนั้นจึงเข้ารหัสพร้อมกันแล้วส่งออกไป ส่วนสำหรับการเข้ารหัสแบบลำดับเป็นการอ่านเข้ารหัสไปพร้อม ๆ กัน

การเข้ารหัสข่าวสารนั้นจะมีจุดมุ่งหมายที่สำคัญ คือ ต้องการลดความซ้ำซ้อนของข่าวสารที่ถูกกำเนิดจากแหล่งกำเนิดข่าวสาร เพื่อให้มีประสิทธิภาพในการส่งข่าวสารสูงขึ้น หรือมีจุดมุ่งหมายที่เป็นรูปธรรมก็คือต้องการลดความยาวของตัวอักษรลงในการส่งข้อความเดียวกันเพราะฉะนั้นข้อความใหม่ที่ถูกสร้างขึ้นมาแทนข่าวสารเดิมนี้เราเรียกว่า “คำรหัส (code word)”

2.4.2.1 การเข้ารหัส (Coding)

เนื่องจากข่าวสารที่ทำการส่งผ่านระบบสื่อสารนั้น เมื่อทำการศึกษาคุณสมบัติเชิงสถิติให้ดีจะพบว่ามีความซ้ำซ้อนของข่าวสารอยู่มาก ดังนั้นการเข้ารหัสจึงเป็นสิ่งจำเป็น เพราะจะทำให้สามารถลดความซ้ำซ้อนของข่าวสารลงได้ ซึ่งหมายถึงจะสามารถส่งข่าวสารนั้นได้ด้วยปริมาณบิตที่น้อยลง และส่งได้เร็วขึ้นทางด้านภาครับก็จะต้องมีระบบถอดรหัสที่สอดคล้องกับทางภาคส่ง ซึ่งก็จะทำให้ได้ข่าวสารเดิมกลับคืนมา การเข้ารหัสนั้นจึงมีกระบวนการดังนี้

n จำนวนบิต
 0 → 00...00
 n จำนวนบิต
 1 → 11...11

จะได้ค่าความน่าจะเป็น

$$P_e = \sum_{k=(n+1)/2}^n \binom{n}{k} \epsilon^k (1-\epsilon)^{n-k} \quad (139)$$

โดยที่ค่า ϵ คือ ความน่าจะเป็น crossover ของช่องสัญญาณ

n คือ ความยาวของลำดับสัญญาณขาเข้าช่องสัญญาณแบบไบนารี

k คือ ความยาวลำดับสัญญาณขาออกต่อสัญญาณแบบไบนารี

ซึ่งค่าของ $(n+1)/2$ คือ ค่าความผิดพลาดที่เกิดขึ้น

ตัวอย่าง : เมื่อกำหนดค่า $n = 5$ กับ 9 และ $\epsilon = 0.001$

ที่ค่า $n = 5$ ค่าความน่าจะเป็นผิดพลาดแสดงได้ดังนี้

$$P_e = \sum_{k=3}^5 0.001^k (0.999)^{5-k} = 9.99 \times 10^{-10} \approx 10^{-9} \quad (140)$$

ที่ค่า $n = 9$ ค่าความน่าจะเป็นผิดพลาดแสดงได้ดังนี้

$$P_e = \sum_{k=5}^9 0.001^k (0.999)^{9-k} = 9.97 \times 10^{-16} \approx 10^{-15} \quad (141)$$

นั่นคือ ค่าความน่าจะเป็นเมื่อ $n = 5$ และ $\epsilon = 0.001$ คือ $9.99 \times 10^{-10} \approx 10^{-9}$

และ ค่าความน่าจะเป็นเมื่อ $n = 9$ และ $\epsilon = 0.001$ คือ $9.97 \times 10^{-16} \approx 10^{-15}$

จากสมการในเรื่องความจุช่องสัญญาณ ($C = 1 - H_b(\epsilon)$) เมื่อแทนค่า $\epsilon = 0.001$ จะได้ค่าความจุช่องสัญญาณ

$$C = 1 - H_b(\epsilon) = 1 - H_b(0.001) = 1 - 0.0114 = 0.9886 \text{ บิต/การส่งสัญญาณ}$$

2.4.2.2 การเข้ารหัสแบบบล็อกเชิงเส้น (Linear Block Code)

การเข้ารหัสแบบบล็อกเชิงเส้นเป็นการเข้ารหัสสัญลักษณ์ (codewords) โดยมีการรวมกันของ 2 สัญลักษณ์ เป็น 1 สัญลักษณ์ จากสับสเปซที่มีขนาด k ของสเปซที่มีขนาด n ซึ่งเมตริกซ์ G (Generator matrix) $= k \times n$ สามารถเขียนสมการเมตริกซ์ของการเข้ารหัสสัญลักษณ์ได้ดังนี้

$$c = uG \quad (142)$$

โดยที่ u คือ ข้อมูลเลขฐานสองที่เรียงตามลำดับความยาว k (ขาเข้าของ encoder)

ระยะทางที่น้อยที่สุด (d_{\min} : minimum distance) มีสมการดังนี้

$$d_{\min} = \min_{i \neq j} d_H(c_i, c_j) \quad (143)$$

d_H คือ ระยะทางของ Hamming

c_i คือ error-correcting capabilities ของเมตริกซ์แถว i

c_j คือ error-correcting capabilities ของเมตริกซ์คอลัมน์ j

น้ำหนักที่น้อยที่สุด (ω_{\min} : minimum weight) มีสมการดังนี้

$$\omega_{\min} = \min_{c_i \neq 0} \omega(c_i) \quad (144)$$

การเข้ารหัสแบบบล็อกเชิงเส้น (Linear block code) นั้นจะมีรูปแบบดังนี้

$$G = \begin{bmatrix} 1 & 0 & \dots & 0 & p_{1,1} & p_{1,2} & \dots & p_{1,n-k} \\ 0 & 1 & \dots & 0 & p_{2,1} & p_{2,2} & \dots & p_{2,n-k} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 1 & \dots & 0 & p_{k,1} & p_{k,2} & \dots & p_{k,n-k} \end{bmatrix} \quad (145)$$

ซึ่งสามารถเขียนค่า G ในรูปของค่า $G = [I_k | P]$

โดยที่ค่า I_k คือ ค่าเมตริกซ์เอกลักษณ์ขนาด $k \times k$

ค่า P คือ เมตริกซ์ขนาด $k \times (n-k)$

เมตริกซ์ตรวจสอบพาริตี (The parity-check matrix) สำหรับการเข้ารหัสทุก ๆ $(n-k) \times n$ คือ

$$cH^t = 0 \quad (146)$$

$$GH^t = 0 \quad (147)$$

โดยที่ H^t คือ ทรานโพสของเมตริกซ์ H

และถ้าค่า G เขียนเป็นรูปแบบ (systematic form) ใหม่จะได้ดังนี้ : $H = [P^t | I_k]$

2.4.2.3 สมรรถนะของการเข้ารหัสแบบบล็อกเชิงเส้น (Performance of Linear Block Codes)

ความผิดที่อาจเป็นไปได้นั้น เกิดจากการเข้ารหัสแบบบล็อกเชิงเส้นด้วยค่าระยะทางที่น้อยที่สุด (d_{\min}) ในการถอดรหัสแบบ hard – decision ซึ่งมีรูปแบบคือ

$$Pe \leq (M - 1)[4p(1 - p)]^{d_{\min}/2} \quad (148)$$

โดยที่ P คือค่าความน่าจะเป็นผิดพลาดสำหรับการตีความข้อมูล (error probability in demodulation)

M คือ เลขการเข้ารหัสสัญลักษณ์ (number codewords) โดยที่ $M = 2^k$

$$Pe \leq (M - 1)Q\left(\frac{d^E}{\sqrt{2N_0}}\right) \quad (149)$$

โดยที่ค่า N_0 คือ สเปกตรัมของความหนาแน่นกำลังสัญญาณรบกวนขาเข้าด้านหนึ่ง

d^E คือ ระยะทางยูคลิเดียนน้อยที่สุด (minimum Euclidean distance) ของรหัส ซึ่งสมการของ d^E แสดงได้ดังนี้

$$d^E = \begin{cases} \sqrt{d_{\min} E} & \text{(สัญญาณแบบอริโทโกนัล)} \\ \sqrt{2d_{\min} E} & \text{(สัญญาณแบบแอนติโพดอล)} \end{cases} \quad (150)$$

และ E คือ พลังงานของแต่ละส่วนประกอบของรหัสสัญลักษณ์

ดังนั้นเราจะได้อ่า

$$p_e \leq \begin{cases} (M - 1)Q\sqrt{\left(\frac{d_{\min} E}{2N_0}\right)} & \text{(สัญญาณแบบอริโทโกนัล)} \\ (M - 1)Q\sqrt{\left(\frac{d_{\min} E}{N_0}\right)} & \text{(สัญญาณแบบแอนติโพดอล)} \end{cases} \quad (151)$$

และ
$$E_b = \frac{nE}{k} = \frac{E}{R_c} \quad (152)$$

โดยที่ค่า $R_c = k/n$ คือ อัตราการเข้ารหัส

E_b คือ พลังงานของแต่ละส่วนประกอบของรหัสสัญลักษณ์ต่อ 1 บิต เราจะได้สมการของ p_e คือ

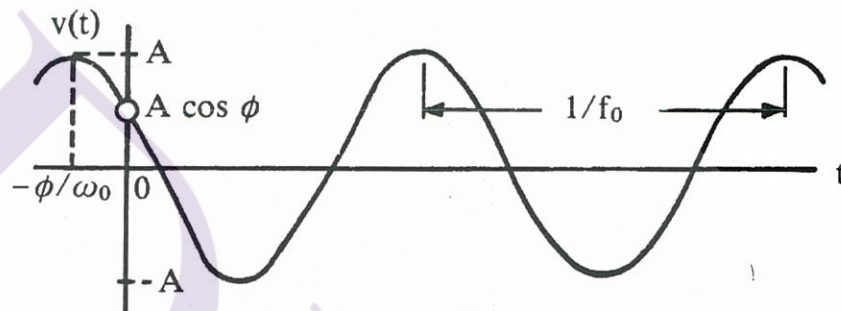
$$p_e \leq \begin{cases} (M - 1)Q\sqrt{\left(\frac{d_{\min} R_c E_b}{2N_0}\right)} & \text{(สัญญาณแบบอริโทโกนัล)} \\ (M - 1)Q\sqrt{\left(\frac{d_{\min} R_c E_b}{N_0}\right)} & \text{(สัญญาณแบบแอนติโพดอล)} \end{cases} \quad (153)$$

2.4.3 ระบบสื่อสารสเปกตรัมแผ่ (Spread Spectrum Communication Systems)

การแสดงสเปกตรัมของสัญญาณจากสัญญาณไซน์ซอซอด์ที่มีแอมพลิจูด (Amplitude) เท่ากับ A ความถี่เชิงมุมเท่ากับ ω_0 และเฟสเท่ากับ ϕ สามารถแสดงให้อยู่ในรูปคณิตศาสตร์ได้ดังนี้

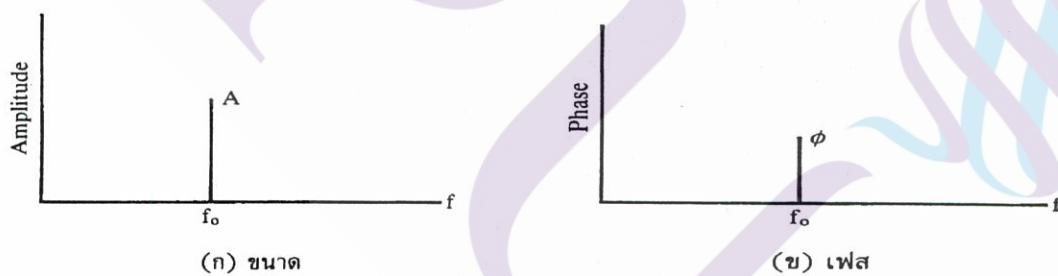
$$v(t) = A \cos(\omega_0 t + \phi) \quad (154)$$

และ สัญญาณสามารถแสดงได้ในรูปที่ 2.3-1



รูปที่ 2.3-1 สัญญาณไซน์ซอซอด์ความถี่เดียว

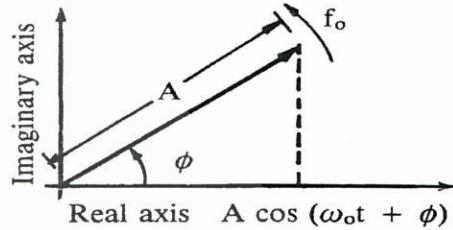
สัญญาณนี้เมื่อนำไปแสดงในรูปของสเปกตรัมจะได้ดังรูปที่ 2.3-2



รูปที่ 2.3-2 สเปกตรัมของสัญญาณไซน์ซอซอด์ความถี่เดียว (ก) ขนาด (ข) เฟส

นอกจากนี้สัญญาณไซน์ซอซอด์ความถี่เดียวยังสามารถแสดงในรูปของเฟสเซอร์ (Phasor) นั้นได้ดังแสดงในรูปที่ 2.3-3

$$A \cos(\omega_0 t + \phi) = \text{Re} [Ae^{j(\omega_0 t + \phi)}]$$

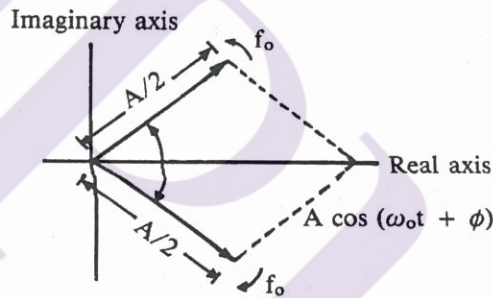


รูปที่ 2.3-3 เฟสเซอร์ของสัญญาณไซน์ซอซอด์ความถี่เดียว

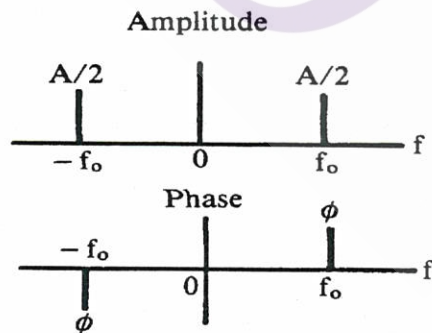
เมื่อแปลง $A \cos(\omega_0 t + \phi)$ ให้อยู่ในรูป

$$A \cos(\omega_0 t + \phi) = \frac{A}{2} e^{j\phi} e^{j\omega_0 t} + \frac{A}{2} e^{-j\phi} e^{-j\omega_0 t} \tag{155}$$

สามารถแสดงเฟสเซอร์และสเปกตรัมแบบความถี่สองด้านดังในรูปที่ 2.3-4 และ 2.3-5 ตามลำดับ



รูปที่ 2.3-4 เฟสเซอร์แบบความถี่สองด้านของสัญญาณไซน์ซอซอด์ความถี่เดียว



รูปที่ 2.3-5 สเปกตรัมแบบความถี่สองด้านของสัญญาณไซน์ซอซอด์ความถี่เดียว

2.4.3.1 สเปกตรัมกำลังของกระบวนการสุ่ม (Stochastic Process)

กำหนดให้ $X(t)$ เป็นตัวแปรสุ่มและ $x(t; \omega_i)$ เป็นฟังก์ชันตัวอย่างของ... การสุ่มนี้ ในการกำหนดความหนาแน่นสเปกตรัมกำลังสำหรับฟังก์ชันตัวอย่าง ทำโดยกำหนดให้

$$x_T(t; \omega_i) = x_{T_i}(t) = \begin{cases} x(t; \omega_i), & |t| < T/2 \\ 0, & \text{ที่อื่น ๆ} \end{cases} \quad (156)$$

จะได้ค่าสเปกตรัมกำลังตามค่าเฉลี่ยทั้งชุดในเชิงสถิติ (Ensemble Average) ดังนี้

$$S_X(f) \stackrel{\text{กำหนด}}{=} E \left[\lim_{T \rightarrow \infty} \frac{|X_T(f)|^2}{T} \right] = \lim_{T \rightarrow \infty} \frac{E[|X_T(f)|^2]}{T} \quad (157)$$

การมอดูเลต (Modulation) สัญญาณหมายถึงการนำสัญญาณไฟฟ้าที่ได้จากทรานสดิวเซอร์ทางด้านต้นทางผสมเข้าไปกับสัญญาณคลื่นพาห้ เพื่อฝากข่าวสารที่มีอยู่ในสัญญาณไฟฟ้าเข้าไปกับสัญญาณคลื่นพาห้ที่มีความถี่สูงกว่ามาก ๆ ประโยชน์ของการมอดูเลตนั้นมีอยู่หลายประการด้วยกัน ซึ่งประโยชน์ที่เด่นชัดมาก ๆ คือทำให้สามารถส่งสัญญาณหลาย ๆ สัญญาณพร้อม ๆ กันได้ โดยการมอดูเลตเข้าไปกับคลื่นพาห้ที่มีความถี่ต่างกัน

2.4.3.2 ระบบสเปกตรัมแผ่แบบ DS (Direct-Sequence Spread Spectrum Systems)

สัญญาณสเปกตรัมแผ่สามารถแสดงอยู่ในรูป

$$v(t) = \sum_{n=-\alpha}^{\alpha} a_n g_T(t - nT_B) \quad (158)$$

โดยที่ $\{a_n = \pm 1, -\alpha < n < \alpha\}$ และ มีค่า $g_T(t)$ เป็นช่วงเวลาของพัลส์ โดยจะได้ค่า $c(t)$ ดังนี้

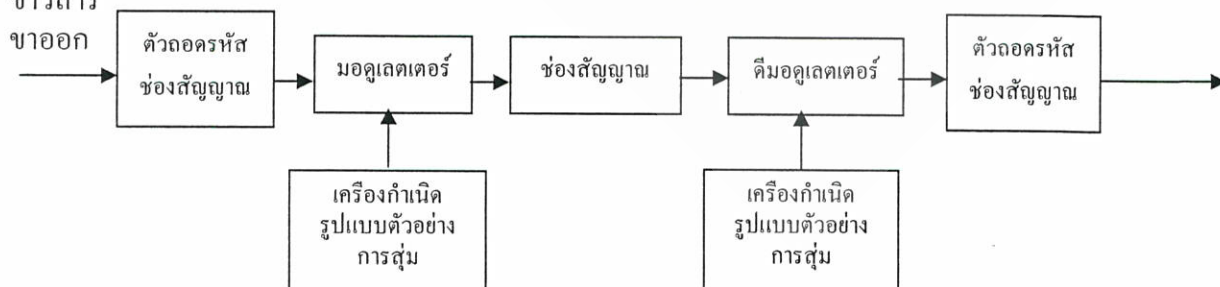
$$c(t) = \sum_{n=-\alpha}^{\alpha} c_n p(t - nT_c) \quad (159)$$

แหล่งกำเนิด

ข้อมูล

ข่าวสาร

ขาออก



รูปที่ 2.3-6 แบบจำลองระบบสื่อสารสเปกตรัมแผ่

ส่วนประกอบพื้นฐานสำคัญของการสื่อสารระบบสเปกตรัมแผ่ นั้น สามารถแสดงในรูปที่ 2.3-6 แบบจำลองระบบสื่อสารสเปกตรัมแผ่ จะประกอบไปด้วย ตัวถอดรหัสช่องสัญญาณ (Channel Encoder) 2 ตัว, ช่องสัญญาณ (Channel), มอดูเลเตอร์ (Modulator), ดีมอดูเลเตอร์ (Demodulator) โดยที่การมอดูเลเตอร์กับการดีมอดูเลเตอร์นั้น เป็นพื้นฐานของระบบสื่อสารดิจิทัลแบบคอนเวนชัน (conventional digital) และเครื่องกำเนิดรูปแบบตัวอย่างการสุ่ม (Pseudorandom pattern generator) 2 ตัว ซึ่งเครื่องกำเนิดรูปแบบตัวอย่างการสุ่มตัวที่หนึ่งนั้นจะใช้ร่วมกับการมอดูเลเตอร์ในการส่งสัญญาณ ส่วนเครื่องกำเนิดรูปแบบตัวอย่างการสุ่มตัวที่สองนั้นจะใช้ร่วมกับการดีมอดูเลเตอร์ในการรับสัญญาณ สิ่งที่ได้จากเครื่องกำเนิดรูปแบบตัวอย่างการสุ่มหรือสัญญาณรบกวน (Pseudonoise, PN) คือ การรับและส่งสัญญาณแผ่ในความถี่ที่มอดูเลเตอร์

2.4.3.2.1 การดีมอดูเลตสัญญาณ (Signal Demodulation)

สัญญาณดีมอดูเลตสามารถเขียนให้อยู่ในรูป

$$A_C v(t) c^2(t) \cos 2 \pi f_{ct} = A_C v(t) \cos 2 \pi f_{ct} \tag{160}$$

โดยที่ค่า

$$\text{sinc } c^2 t = 1 \text{ สำหรับทุกค่า } t$$

ผลกระทบของการดีสเปรด (despread) สำหรับการแทรกสอดแถบแคบจะได้สมการของสัญญาณรับได้ (received signal) คือ

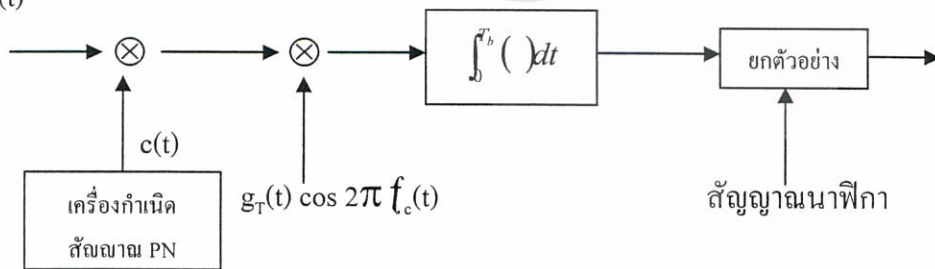
$$r(t) = A_C v(t) \alpha(t) \cos 2 \pi f_{ct} + i(t) \tag{161}$$

โดยที่ค่า $i(t)$ คือ ค่าการแทรกสอด

ดังนั้น $r(t) c(t) = A_C v(t) c(t) \cos 2 \pi f_{ct} + i(t) c(t) \tag{162}$

ภาครับสัญญาณ

ถอดรหัส $r(t)$



รูปที่ 2.3-7 การดีมอดูเลตของสัญญาณสเปกตรัมแผ่แบบ DS

รูปที่ 2.3-7 แสดงการคิมอดูเลตของสัญญาณสเปกตรัมแผ่แบบ DS นั้นจะมีการรับสัญญาณมาเป็นอันดับแรกเพื่อที่จะทำการ multiplied โดยใช้การกำเนิดแบบจำลองรูปคลื่น $c(t)$ โดยมีความต่อเนื่องเข้ากับสัญญาณการเข้ารหัส PN ที่เครื่องรับ

2.4.3.2.2 ความน่าจะเป็นของความผิดพลาด (Probability of Error)

ความน่าจะเป็นของความผิดพลาดที่เกิดขึ้นสามารถเขียนให้อยู่ในรูปสมการได้

$$P_b = Q \sqrt{\left(\frac{2E_b}{N_o} \right)} \quad (163)$$

โดยที่ความน่าจะเป็นสำหรับความผิดพลาดที่เกิดขึ้น สามารถหาได้จากสมการดังต่อไปนี้

$$P_b = Q \sqrt{\left(\frac{2E_b}{P_J/W} \right)} = Q \sqrt{\left(\frac{2E_b}{J_o} \right)} \quad (164)$$

โดยที่ค่า $N_o \ll P_J/W$

และ

$$P_b = Q \sqrt{\left(\frac{2E_b}{N_o + P_J/W} \right)} = Q \sqrt{\left(\frac{2E_b}{N_o + J_o} \right)} \quad (165)$$

2.4.3.2.3 มارجิ้นของการแจมมิง (The Jamming Margin)

สัญญาณของการแทรกสอดคือสัญญาณแจมมิง ซึ่งสามารถเขียนในรูป E_b/J_o

$$\frac{E_b}{J_o} = \frac{P_s T_b}{P_J/W} = \frac{P_s/R}{P_J/W} = \frac{W/R}{P_J/P_s} \quad (166)$$

ดังนั้นจะได้

$$10 \log \frac{P_J}{J_o} = 10 \log \frac{W}{R} - 10 \log \left(\frac{E_b}{J_o} \right) \quad (167)$$

$$\left(\frac{P_J}{P_s} \right)_{dB} = \left(\frac{W}{R} \right)_{dB} - \left(\frac{E_b}{J_o} \right)_{dB} \quad (168)$$

ซึ่งค่าอัตราส่วน $(P_J/P_s)_{dB}$ เรียกว่า มارجิ้นของการแจมมิง

2.4.3.2.4 สมรรถนะสัญญาณสเปกตรัมแผ่

ด้วยวิธีการเข้ารหัสแบบไบนารีเชิงเส้น (แบบบล็อก หรือ แบบคอนวอลูชัน) โดยมีค่า SNR ที่ขาออกของการถอดรหัสแบบ soft - decision ซึ่งเป็นการเพิ่มขึ้นของอัตราการใช้ โดยที่สามารถหาเกณฑ์การเข้ารหัสได้จากสมการ

$$\text{อัตราการใช้} = R_c d_{\min}^H \quad (169)$$

โดยที่ R_c คือ อัตราการเข้ารหัส

และ d_{\min}^H คือ ระยะทางการเข้ารหัสที่น้อยที่สุดของแฮมมิง

$$\text{จากสมการ} \quad \left(\frac{P_J}{P_S}\right)_{dB} = \left(\frac{W}{R}\right)_{dB} - \left(\frac{E_b}{J_o}\right)_{dB} \quad (170)$$

$$\text{จะได้} \quad \left(\frac{P_J}{P_S}\right)_{dB} = \left(\frac{W}{R}\right)_{dB} + (CG)_{dB} - \left(\frac{E_b}{J_o}\right)_{dB} \quad (171)$$

2.4.3.2.5 การประยุกต์สำหรับสัญญาณสเปกตรัมแผ่แบบ DS

การประยุกต์สำหรับสัญญาณสเปกตรัมแผ่แบบ DS นั้นสามารถแบ่งออกได้เป็น

- การส่งผ่านสัญญาณในระดับสัญญาณต่ำ (Low-Delectability Signal Transmission)
- Code Division Multiple Access หรือ CDMA

ระบบ CDMA สามารถเขียนให้อยู่ในรูป

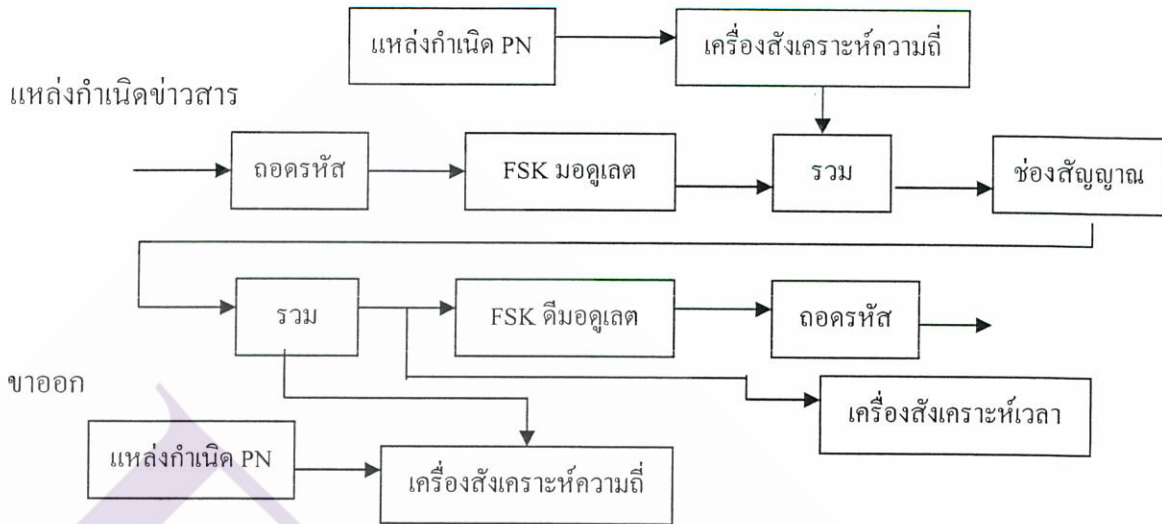
$$\frac{P_S}{P_N} = \frac{P_S}{(N_u - 1)P_S} = \frac{1}{N_u - 1} \quad (172)$$

โดยที่ P_S คือ ค่ากำลังเฉลี่ยของสัญญาณ

P_N คือ ค่ากำลังเฉลี่ยของการรบกวน

2.4.3.3 สเปกตรัมแผ่แบบการกระโดดความถี่ (Frequency – Hopped Spread Spectrum)

รูปที่ 2.3-8 เป็นการแสดงบล็อกไดอะแกรมการส่งและรับสำหรับระบบสื่อสารแผ่ FH โดยเมื่อแหล่งกำเนิดข่าวสาร (Information sequence) เข้ามาจะมีการถอดรหัส (Encoder) ส่งต่อไปยังการมอดูเลต FSK (FSK modulator) เพื่อที่จะทำการรวมกับเครื่องสังเคราะห์ความถี่ (Frequency synthesizer) ที่เกิดจากแหล่งกำเนิด PN (PN sequence generator) ส่งต่อไปยังช่องสัญญาณแล้วทำการสังเคราะห์ความถี่ที่เกิดจากแหล่งกำเนิด PN ออก ซึ่งผลที่ได้จะทำการแยกออกเป็นการดีมอดูเลต FSK (FSK demodulator) เพื่อทำการถอดรหัส (Decoder) กับ เครื่องสังเคราะห์เวลา (Time-synthesizer)



รูปที่ 2.3-8 ระบบสเปกตรัมแผ่แบบ FH

ความผิดพลาดที่เป็นไปได้สำหรับสัญญาณ FH

$$P_b = \frac{1}{2} \exp^{-E_b/2N_o} \tag{173}$$

โดยที่มีค่า $E_b = P_s T_b = P_s / R$ (174)

โดยที่ P_s คือ กำลังเฉลี่ยของสัญญาณ

R คือ อัตราบิต

เมื่อค่า $J_o \gg N_o$ ค่า SNR จะสามารถคำนวณได้จากสมการ

$$\frac{E_b}{J_o} = \frac{W/R}{P_j/P_s} \tag{175}$$

โดยที่ W/R คือ อัตราการขยายกระบวนการ (the processing gain)

P_j/P_s คือ มารัจเงินของการแจ่มมิ่งสำหรับสัญญาณสเปกตรัมแผ่แบบ FH

ดังนั้นค่าความน่าจะเป็นผิดพลาด (The probability of error) ที่เป็นไปได้สามารถเขียนในรูป

$$\frac{1}{2} \exp(-\alpha \rho_b / 2) \quad ; \quad \rho_b \equiv E_b / J_o \tag{176}$$

ซึ่งค่าของ $P_2(\alpha)$ มีค่าดังนี้

$$\begin{aligned} P_2(\alpha) &= (\alpha/2) \exp(-\alpha \rho_b / 2) \\ &= (\alpha/2) \exp\left(\frac{-\alpha W/R}{2P_j/P_s}\right) \end{aligned} \tag{177}$$

โดยที่ $P_2(\alpha)$ คือ ค่าความน่าจะเป็นที่มากที่สุด

การใช้สัญญาณไคเวอร์ซิติ (Diversity) เพื่อเอาชนะการแทรกสอดบางช่วงของแถบความถี่
ค่าของตัวแปรที่ใช้ในการตัดสินใจที่จะนำมาประกอบกันคือ

$$x = \omega_1 r_{11} + \omega_2 r_{12} \quad (178)$$

$$y = \omega_2 r_{21} + \omega_2 r_{22} \quad (179)$$

โดยที่ r_{11} และ r_{21} คือ ค่า สัญญาณขาออกของสัญญาณช่วงแรกที่สื่อสาร

r_{12} และ r_{22} คือ ค่า สัญญาณขาออกของสัญญาณช่วงที่สองที่สื่อสาร

โดยที่มีค่าความผิดพลาดที่เป็นไปได้ในการแทรกการรบกวนคือ

$$P_2(2) = \frac{K_2}{\rho_b^D} , p_b > 2 \quad (180)$$

โดยที่มีค่า K_2 เป็นค่าคงที่ และ ค่า $p_b = E_b/J_o$

$$P_2(D) = \frac{K_D}{\rho_b^D} , p_b > 2 \quad (181)$$

โดยที่มีค่า K_D เป็นค่าคงที่ และ ค่า $p_b = E_b/J_o$

หลักการและทฤษฎีเบื้องต้นดังกล่าวมาแล้วในข้างต้นนั้น จะเป็นส่วนสำคัญส่วนหนึ่งที่จะ
สามารถช่วยให้ออกแบบและสร้างโปรแกรมแสดงผล GUI (Graphical User Interface) เพื่อ
แสดงผลลัพธ์ โดยใช้ซอฟต์แวร์ MATLAB และงานวิจัยนี้ ประสบความสำเร็จได้ เพราะในการ
ออกแบบและสร้างโปรแกรมแสดงผล GUI ให้สามารถแสดงผลลัพธ์ได้นั้น จะต้องมีความรู้และ
ความเข้าใจในหลักการและทฤษฎีเบื้องต้นของเนื้อหา นั้น ดังนั้นในบทที่ 3 จะเป็นเนื้อหาที่เกี่ยวข้อง
กับขั้นตอนต่าง ๆ ในการออกแบบและสร้างโปรแกรมแสดงผล GUI เพื่อนำมาใช้ในระบบงาน
การศึกษาและวิจัยด้านวิศวกรรมสื่อสารหรือโทรคมนาคม

บทที่ 3

การสร้างและออกแบบจำลองของโปรแกรม GUI (Graphical User Interface)

3.1 กล่าวนำ

แบบจำลอง (Model) คือสิ่งที่ถูกสร้างขึ้นมาเพื่อที่จะช่วยทำให้เกิดความเข้าใจลักษณะการทำงานของระบบ และเพื่อที่จะสามารถอธิบายพฤติกรรมที่เกิดขึ้นตามธรรมชาติที่ได้จากการจัดทำ การเลียนแบบลักษณะของพฤติกรรมที่เกิดขึ้นในระบบ

แบบจำลองของระบบการสื่อสารแบบแอนะล็อก, ดิจิทัล และสเปกตรัม ในที่นี้แสดงด้วยค่า โปรแกรม GUI (Graphical User Interface) ซึ่งมีความสามารถแสดงผลการเปลี่ยนแปลงลักษณะของ พฤติกรรมที่เกิดขึ้นได้จากการเปลี่ยนค่าของตัวแปรต่าง ๆ จากผู้ใช้งานในแบบจำลองของ GUI

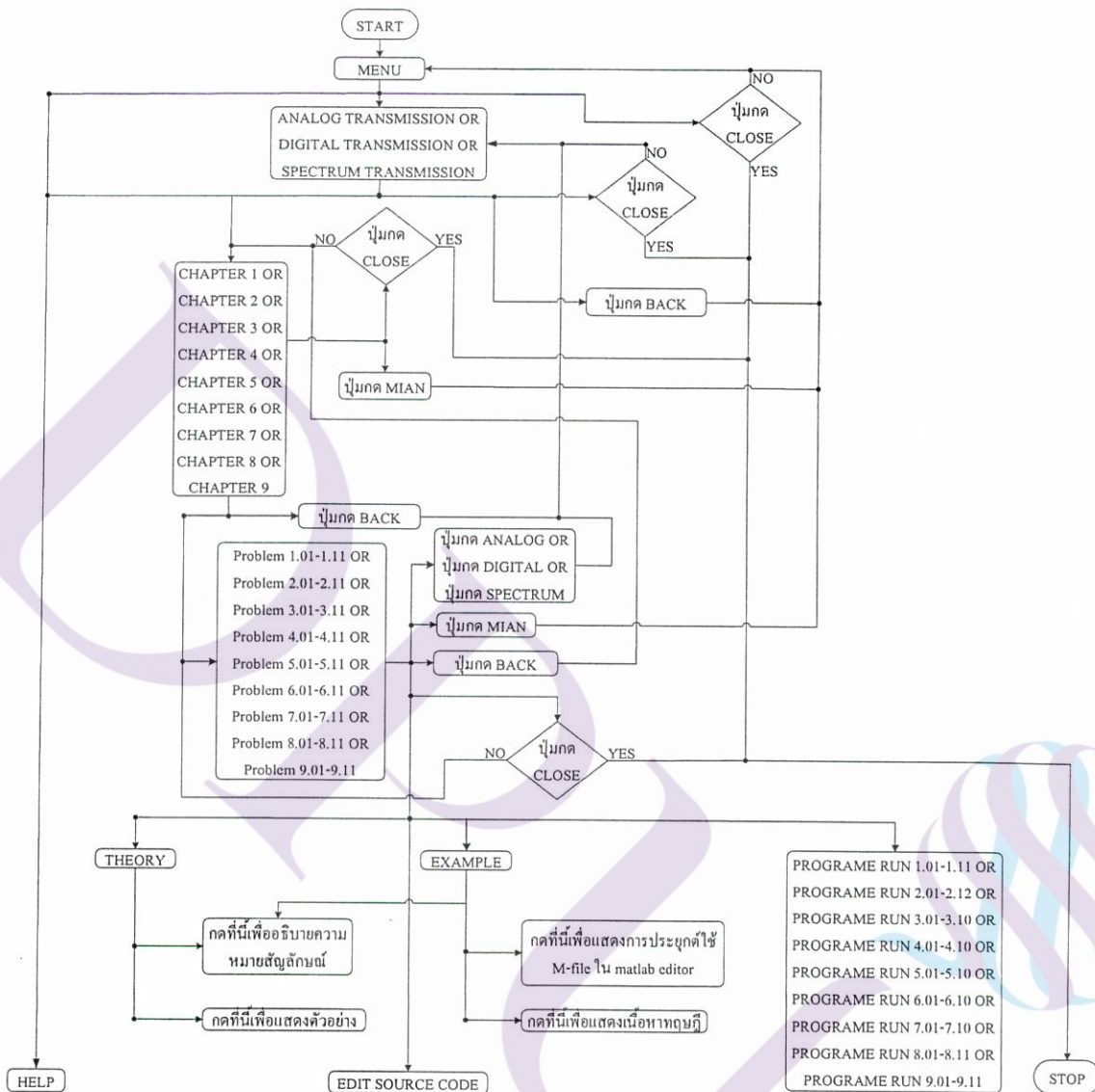
3.2 การออกแบบและสร้างโปรแกรมแสดงผล GUI โดยใช้ซอฟต์แวร์ MATLAB

เริ่มต้นจากการศึกษาตัวอย่างโปรแกรม MATLAB ที่มีอยู่แล้ว โดยศึกษาได้จากหนังสือ CONTEMPORARY COMMUNICATION SYSTEMS USING MATLAB เป็นหลัก ซึ่งมีเนื้อหา เกี่ยวกับระบบการสื่อสารแอนะล็อก, ดิจิทัล และสเปกตรัม จากนั้นนำตัวอย่างโปรแกรมต่าง ๆ บน พื้นฐานซอฟต์แวร์ MATLAB ที่มีอยู่แล้วนั้น มาทำการออกแบบสร้างแบบจำลองโปรแกรม แสดงผลโดยใช้โปรแกรม GUI

ในการออกแบบสร้างแบบจำลองโปรแกรมแสดงผล GUI นั้นมีเหตุผลดังต่อไปนี้

1. ให้เกิดความง่ายต่อการทำความเข้าใจ ในการศึกษาทฤษฎีเบื้องต้นสำหรับผู้สนใจ เกี่ยวกับวิศวกรรมการสื่อสารในเรื่อง สัญญาณแอนะล็อก, ดิจิทัล และสเปกตรัม
2. มีการแยกหัวข้อของเนื้อหาออกเป็นส่วน ๆ เพื่อง่ายต่อการศึกษาทำความเข้าใจและ ค้นคว้า
3. สามารถทำการเปลี่ยนค่าพารามิเตอร์ เพื่อทำให้เกิดผลลัพธ์ที่เปลี่ยนไป
4. ดูผลลัพธ์ในลักษณะของกราฟที่เกิดขึ้นของแต่ละโปรแกรม
5. ให้ผู้ที่มีความสนใจศึกษาโปรแกรม MATLAB และ GUI ทำการศึกษา

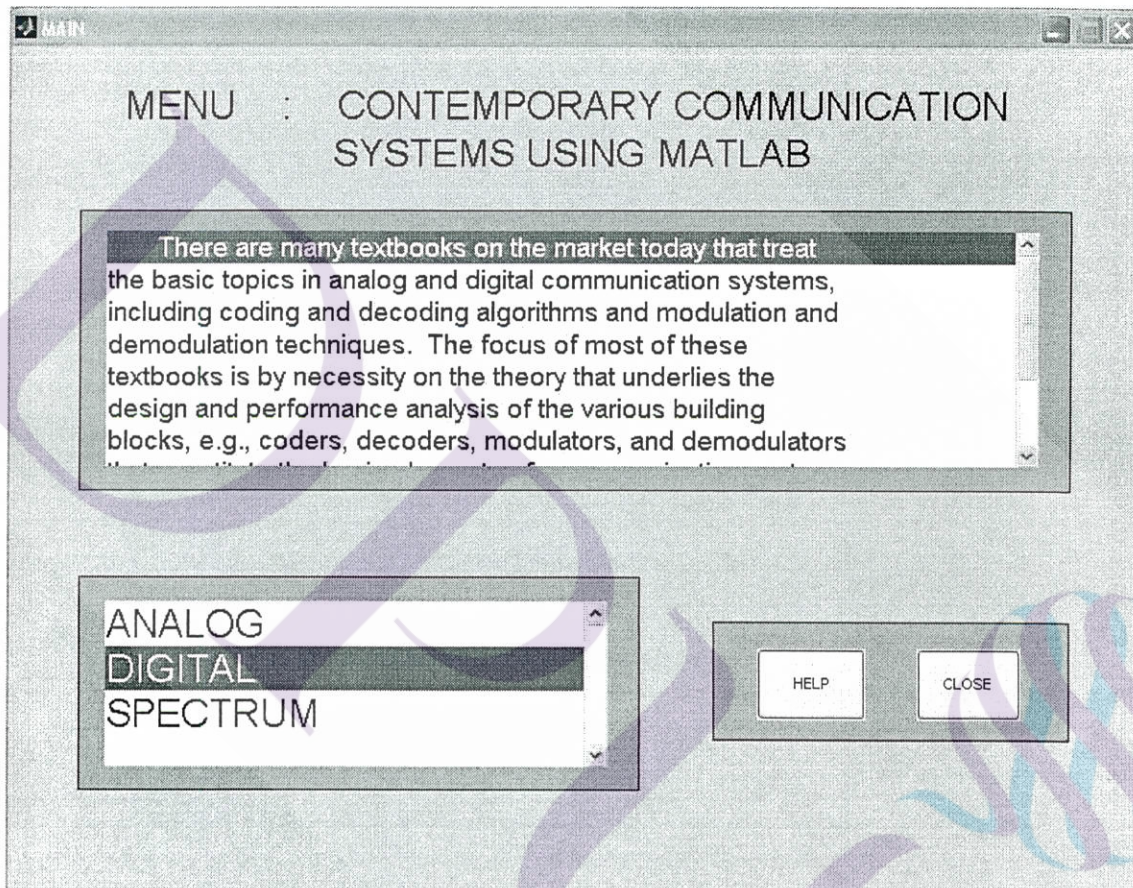
จากแบบจำลองโปรแกรม GUI นี้สามารถแสดงการเขียนแผนผังแบบจำลอง ได้ดังรูปที่ 3.1



รูปที่ 3.1 แผนผังสำหรับแบบจำลอง SRAPCES โดยใช้โปรแกรม GUI

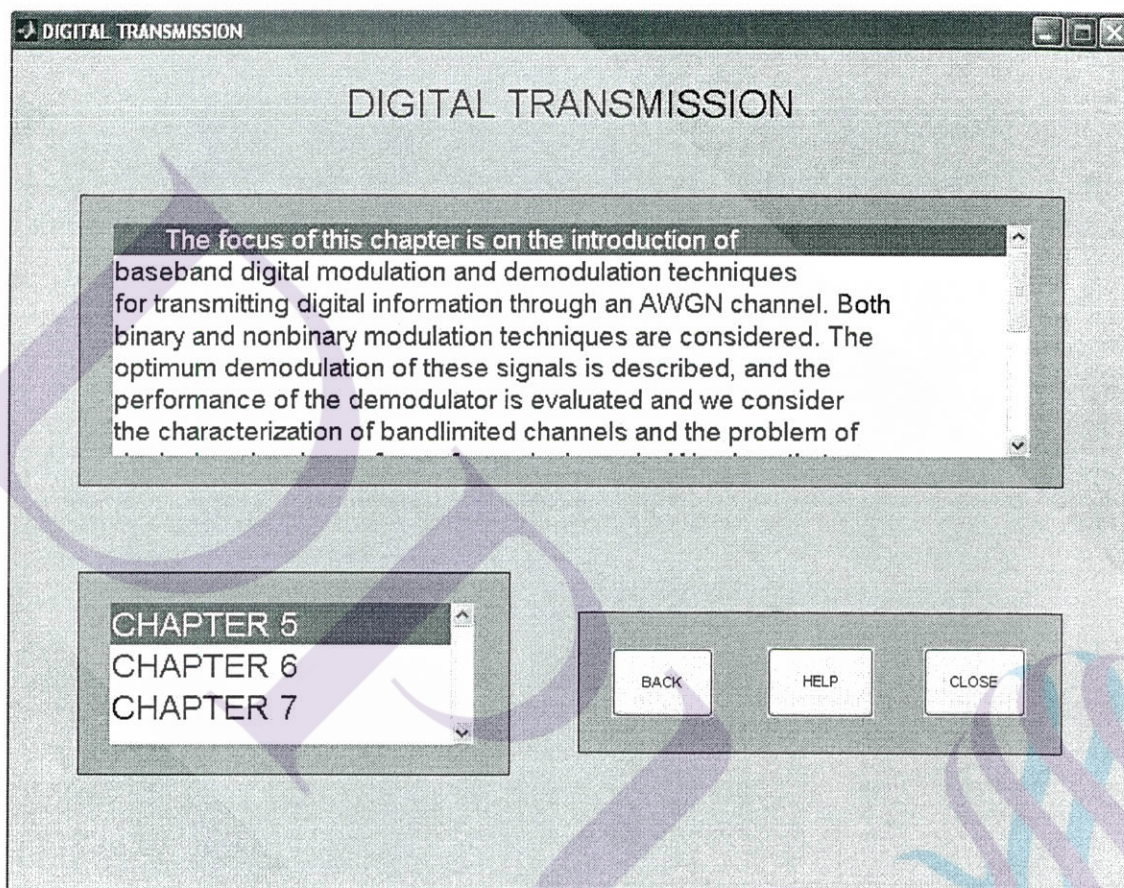
จากรูปที่ 3.1 แผนผังสำหรับแบบจำลองโปรแกรม GUI นี้เราสามารถทำการออกแบบและสร้างแบบจำลองในการแสดงผล ได้ดังต่อไปนี้

1. การสร้างรูปภาพแสดงผล MENU



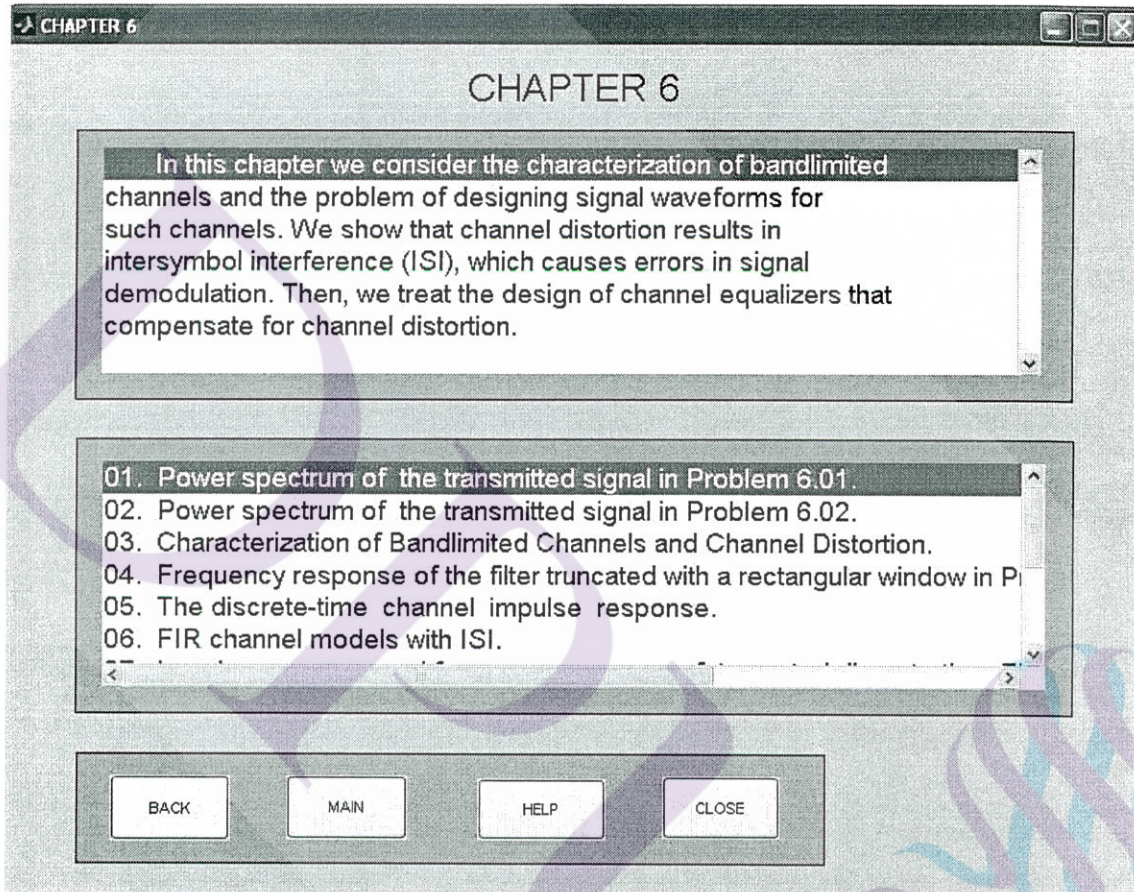
รูปที่ 3.2 แบบจำลองหน้าจอแสดงผล MENU

2. ทำการยกตัวอย่างการสร้างรูปหน้าแสดงผล DIGITAL TRANSMISSION จากทั้งหมดมี 3 หน้าแสดงผล คือ ANALOG TRANSMISSION, DIGITAL TRANSMISSION และ SPECTRUM TRANSMISSION



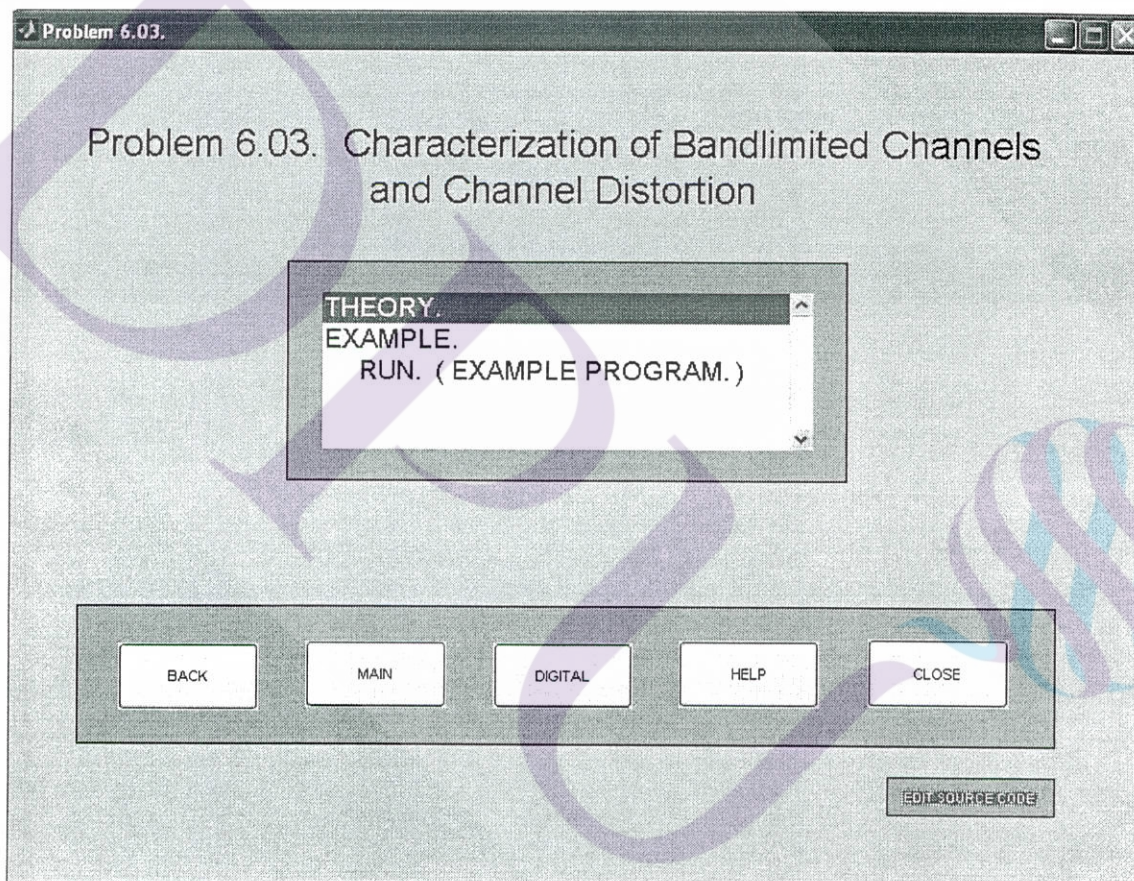
รูปที่ 3.3 แบบจำลองหน้าแสดงผล DIGITAL TRANSMISSION

3. ทำการยกตัวอย่างการสร้างรูปหน้าแสดงผล CHAPTER 6 จากทั้งหมดมี 9 หน้าแสดงผล คือ CHAPTER 1, CHAPTER 2, CHAPTER 3, CHAPTER 4, CHAPTER 5, CHAPTER 6, CHAPTER 7, CHAPTER 8 และ CHAPTER 9



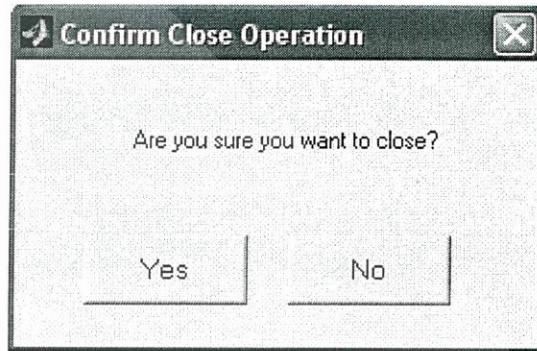
รูปที่ 3.4 แบบจำลองหน้าแสดงผล CHAPTER 6

4. ทำการยกตัวอย่างการสร้างรูปหน้าแสดงผล Problem 6.03 จากทั้งหมดมี 77 หน้าแสดงผล คือ Problem 1 แบ่งเป็น Problem 1.01 - Problem 1.08, Problem 2 แบ่งเป็น Problem 2.01 - Problem 2.09, Problem 3 แบ่งเป็น Problem 3.01 - Problem 3.09, Problem 4 แบ่งเป็น Problem 4.01 - Problem 4.02, Problem 5 แบ่งเป็น Problem 5.01 - Problem 5.11, Problem 6 แบ่งเป็น Problem 6.01 - Problem 6.12, Problem 7 แบ่งเป็น Problem 7.01 - Problem 7.10, Problem 8 แบ่งเป็น Problem 8.01 - Problem 8.12, Problem 9 แบ่งเป็น Problem 9.01 - Problem 9.04



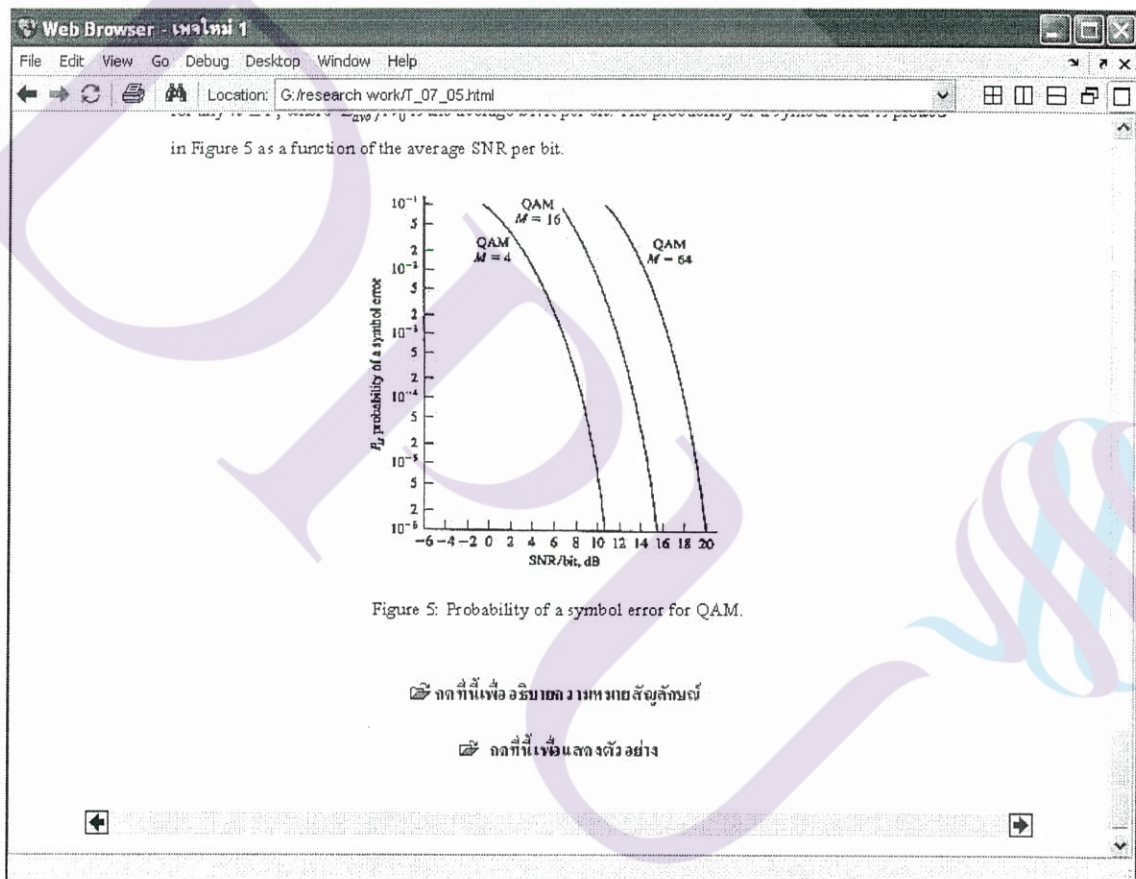
รูปที่ 3.5 แบบจำลองหน้าแสดงผล Problem 6.03

5. รูปหน้าแสดงผล Confirm Close Operation



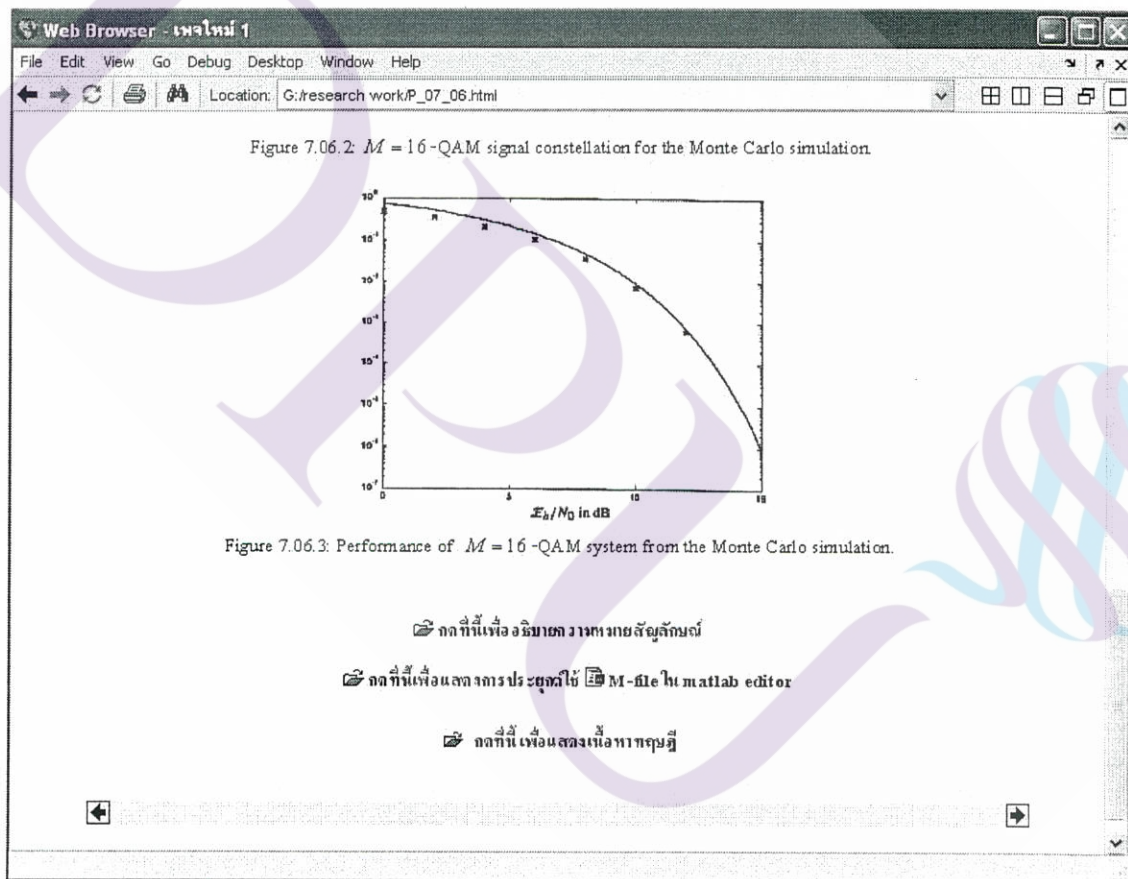
รูปที่ 3.6 การสร้างแบบจำลองหน้าแสดงผล Confirm Close Operation

6. ทำการยกตัวอย่างการสร้างรูปหน้าแสดงผลของ THEORY จากทั้งหมดมี 77 หน้าแสดงผล คือ THEORY 1 แบ่งเป็น THEORY 1.01 - THEORY 1.08, THEORY 2 แบ่งเป็น THEORY 2.01 - THEORY 2.09, THEORY 3 แบ่งเป็น THEORY 3.01 - THEORY 3.09, THEORY 4 แบ่งเป็น THEORY 4.01 - THEORY 4.02, THEORY 5 แบ่งเป็น THEORY 5.01 - THEORY 5.11, THEORY 6 แบ่งเป็น THEORY 6.01 - THEORY 6.12, THEORY 7 แบ่งเป็น THEORY 7.01 - THEORY 7.10, THEORY 8 แบ่งเป็น THEORY 8.01 - THEORY 8.12, THEORY 9 แบ่งเป็น THEORY 9.01 - THEORY 9.04



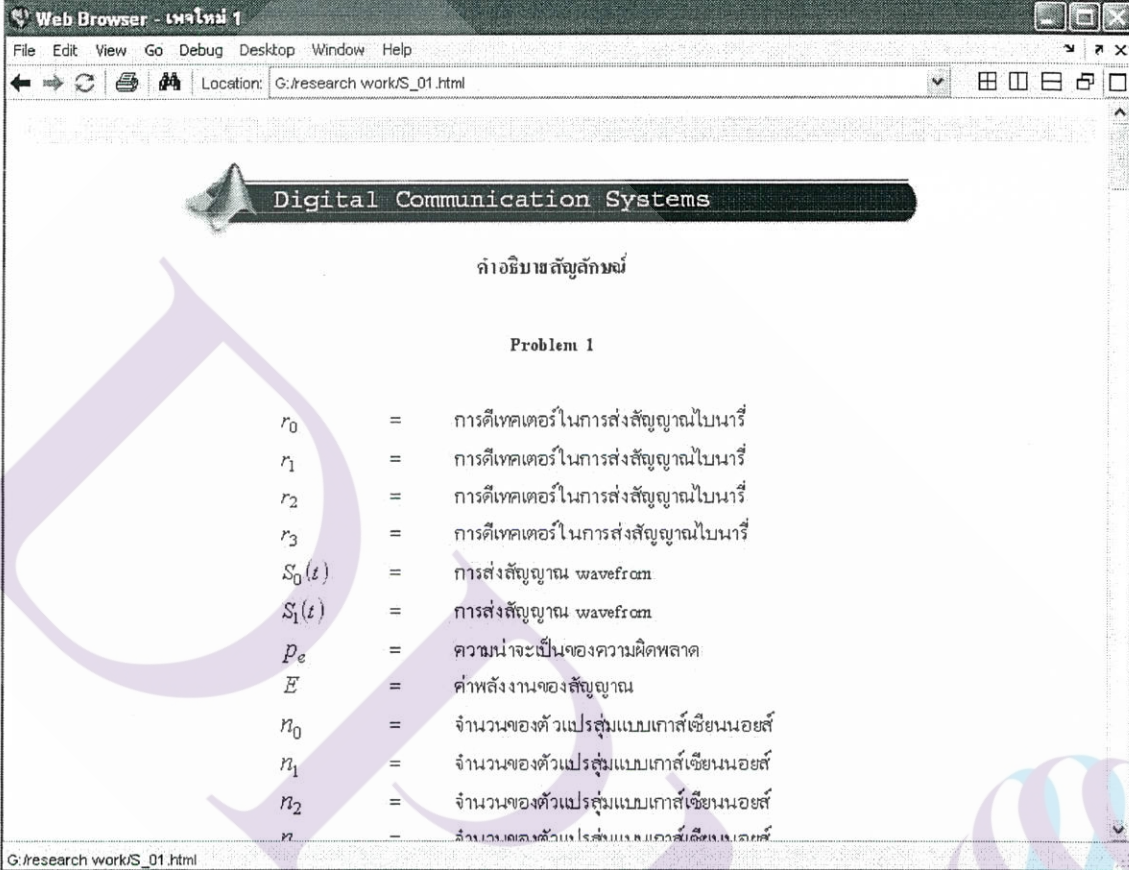
รูปที่ 3.7 แบบจำลองหน้าแสดงผล THEORY

7. ทำการยกตัวอย่างการสร้างรูปหน้าแสดงผล EXAMPLE จากทั้งหมดมี 77 หน้าแสดงผล คือ EXAMPLE 1 แบ่งเป็น EXAMPLE 1.01 - EXAMPLE 1.08, EXAMPLE 2 แบ่งเป็น EXAMPLE 2.01 - EXAMPLE 2.09, EXAMPLE 3 แบ่งเป็น EXAMPLE 3.01 - EXAMPLE 3.09, EXAMPLE 4 แบ่งเป็น EXAMPLE 4.01 - EXAMPLE 4.02, EXAMPLE 5 แบ่งเป็น EXAMPLE 5.01 - EXAMPLE 5.11, EXAMPLE 6 แบ่งเป็น EXAMPLE 6.01 - EXAMPLE 6.12, EXAMPLE 7 แบ่งเป็น EXAMPLE 7.01 - EXAMPLE 7.10, EXAMPLE 8 แบ่งเป็น EXAMPLE 8.01 - EXAMPLE 8.12, EXAMPLE 9 แบ่งเป็น EXAMPLE 9.01 - EXAMPLE 9.04



รูปที่ 3.8 แบบจำลองหน้าแสดงผล EXAMPLE

8. การสร้างรูปหน้าแสดงผล คำอธิบายความหมายสัญลักษณ์



The screenshot shows a web browser window with the title 'Web Browser - เหนงใหม่ 1'. The address bar shows the location 'G:/research work/S_01.html'. The page content is as follows:

Digital Communication Systems

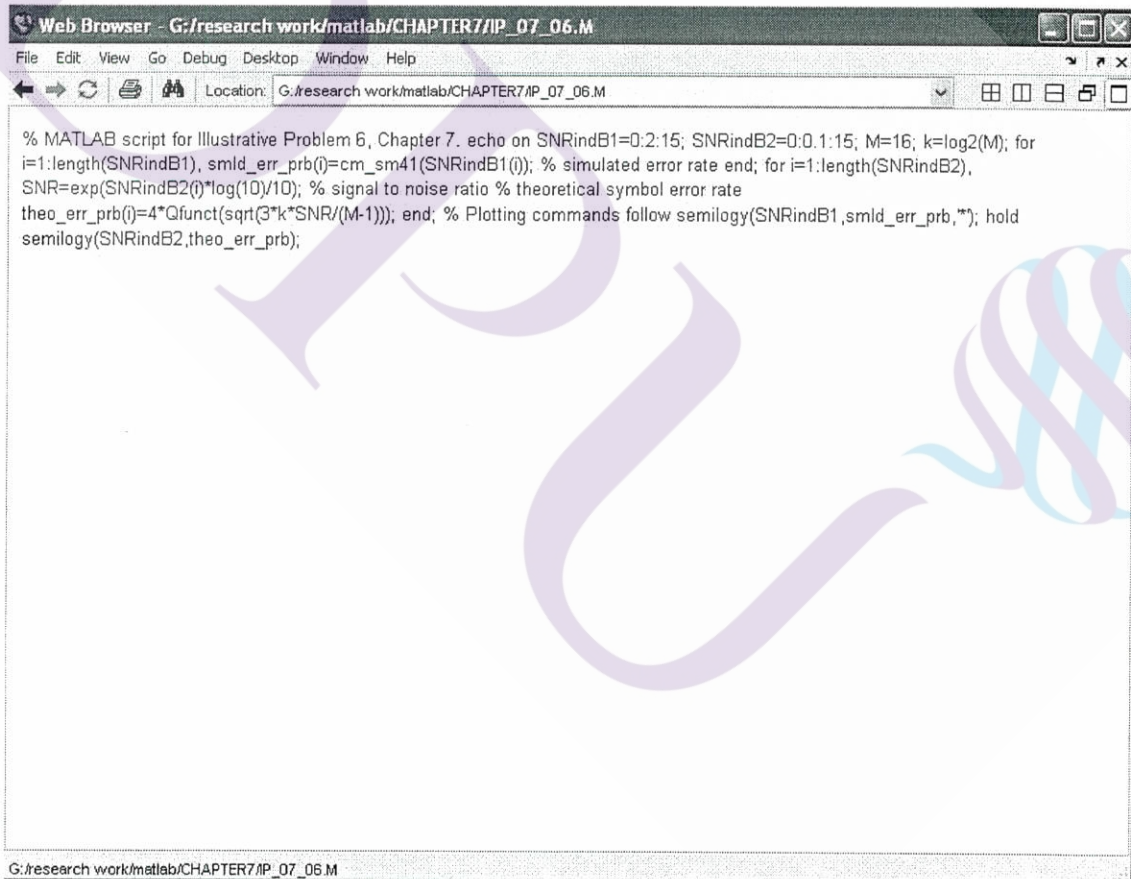
คำอธิบายสัญลักษณ์

Problem 1

r_0	=	การตีเทคเตอร์ในการส่งสัญญาณไบนารี
r_1	=	การตีเทคเตอร์ในการส่งสัญญาณไบนารี
r_2	=	การตีเทคเตอร์ในการส่งสัญญาณไบนารี
r_3	=	การตีเทคเตอร์ในการส่งสัญญาณไบนารี
$S_0(t)$	=	การส่งสัญญาณ waveform
$S_1(t)$	=	การส่งสัญญาณ waveform
p_e	=	ความน่าจะเป็นของความผิดพลาด
E	=	ค่าพลังงานของสัญญาณ
n_0	=	จำนวนของตัวแปรสุ่มแบบเกาส์เซียนนอยส์
n_1	=	จำนวนของตัวแปรสุ่มแบบเกาส์เซียนนอยส์
n_2	=	จำนวนของตัวแปรสุ่มแบบเกาส์เซียนนอยส์
n	=	จำนวนของตัวแปรสุ่มแบบเกาส์เซียนนอยส์

รูปที่ 3.9 รูปหน้าแสดงผล คำอธิบายความหมายสัญลักษณ์

9. ทำการยกตัวอย่างการสร้างรูปหน้าแสดงผลการประยุกต์ใช้ M-file ใน matlab editor จากทั้งหมดมี 77 หน้าแสดงผล คือ การประยุกต์ใช้ 1 แบ่งเป็น การประยุกต์ใช้ 1.01 – การประยุกต์ใช้ 1.08, การประยุกต์ใช้ 2 แบ่งเป็น การประยุกต์ใช้ 2.01 - การประยุกต์ใช้ 2.09, การประยุกต์ใช้ 3 แบ่งเป็น การประยุกต์ใช้ 3.01 – การประยุกต์ใช้ 3.09, การประยุกต์ใช้ 4 แบ่งเป็น การประยุกต์ใช้ 4.01 – การประยุกต์ใช้ 4.02, การประยุกต์ใช้ 5 แบ่งเป็น การประยุกต์ใช้ 5.01 – การประยุกต์ใช้ 5.11, การประยุกต์ใช้ 6 แบ่งเป็น การประยุกต์ใช้ 6.01 – การประยุกต์ใช้ 6.12, การประยุกต์ใช้ 7 แบ่งเป็น การประยุกต์ใช้ 7.01 – การประยุกต์ใช้ 7.10, การประยุกต์ใช้ 8 แบ่งเป็น การประยุกต์ใช้ 8.01 – การประยุกต์ใช้ 8.12, การประยุกต์ใช้ 9 แบ่งเป็น การประยุกต์ใช้ 9.01 – การประยุกต์ใช้ 9.04



```

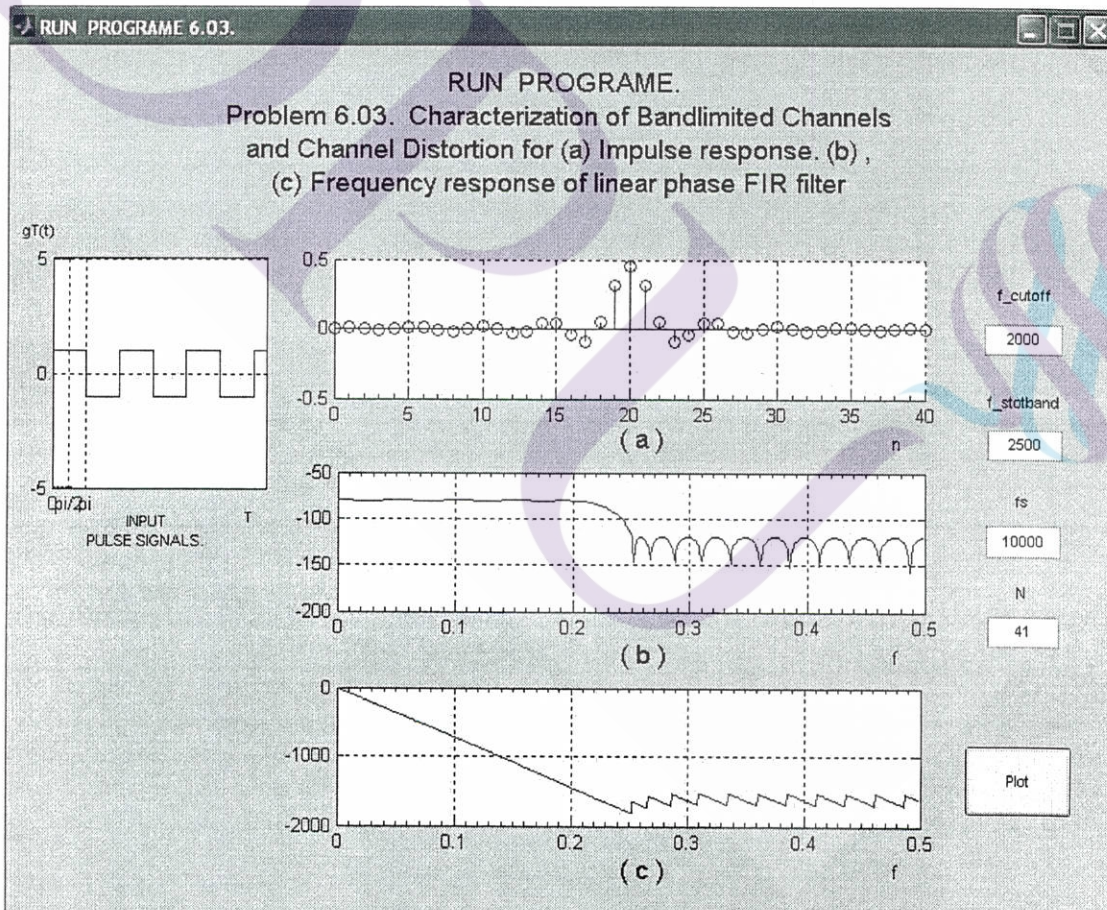
Web Browser - G:/research work/matlab/CHAPTER7/IP_07_06.M
File Edit View Go Debug Desktop Window Help
Location: G:/research work/matlab/CHAPTER7/IP_07_06.M

% MATLAB script for Illustrative Problem 6, Chapter 7. echo on SNRindB1=0:2:15; SNRindB2=0:0.1:15; M=16; k=log2(M); for
i=1:length(SNRindB1), smld_err_prb(i)=cm_sm41(SNRindB1(i)); % simulated error rate end; for i=1:length(SNRindB2),
SNR=exp(SNRindB2(i)*log(10)/10); % signal to noise ratio % theoretical symbol error rate
theo_err_prb(i)=4*Qfunc(sqrt(3*k*SNR/(M-1))); end; % Plotting commands follow semilogy(SNRindB1,smld_err_prb,**); hold
semilogy(SNRindB2,theo_err_prb);

```

รูปที่ 3.10 รูปหน้าแสดงผลการประยุกต์ใช้ M-file ใน matlab editor

10. ทำการยกตัวอย่างการสร้างรูปหน้าแสดงผล RUN PROGRAME จากทั้งหมดมี 77 หน้า
 แสดงผล คือ RUN PROGRAME 1 แบ่งเป็น RUN PROGRAME 1.01 - RUN
 PROGRAME 1.08, RUN PROGRAME 2 แบ่งเป็น RUN PROGRAME 2.01 - RUN
 PROGRAME 2.09, RUN PROGRAME 3 แบ่งเป็น RUN PROGRAME 3.01 - RUN
 PROGRAME 3.09, RUN PROGRAME 4 แบ่งเป็น RUN PROGRAME 4.01 - RUN
 PROGRAME 4.02, RUN PROGRAME 5 แบ่งเป็น RUN PROGRAME 5.01 - RUN
 PROGRAME 5.11, RUN PROGRAME 6 แบ่งเป็น RUN PROGRAME 6.01 - RUN
 PROGRAME 6.12, RUN PROGRAME 7 แบ่งเป็น RUN PROGRAME 7.01 - RUN
 PROGRAME 7.10, RUN PROGRAME 8 แบ่งเป็น RUN PROGRAME 8.01 - RUN
 PROGRAME 8.12, RUN PROGRAME 9 แบ่งเป็น RUN PROGRAME 9.01 - RUN
 PROGRAME 9.04



รูปที่ 3.11 แบบจำลองหน้าแสดงผล RUN PROGRAME

3.3 โปรแกรมที่ใช้ในการสร้างหน้าแสดงผล GUI

ในการออกแบบและสร้างโปรแกรมแสดงผล GUI (Graphical User Interface) ของเรื่องแอนะล็อก ดิจิทัล และสเปกตรัม จากหัวข้อที่ 3.2 นั้น จะต้องทำการออกแบบและเขียนโปรแกรมในการสร้าง GUI หลัก ๆ ตามหัวข้อต่อไปนี้

3.3.1 โปรแกรมในการสร้างหน้าแสดงผล MENU

3.3.2 โปรแกรมในการสร้างหน้าแสดงผล DIGITAL TRANSMISSION

3.3.3 โปรแกรมในการสร้างหน้าแสดงผล CHAPTER

3.3.4 โปรแกรมในการสร้างหน้าแสดงผล PROBLEM

3.3.5 โปรแกรมในการสร้างหน้าแสดงผล M-file ใน matlab editor ก่อนสร้างเป็น GUI

3.3.6 โปรแกรมในการสร้างหน้าแสดงผล RUN PROGRAME

3.3.7 โปรแกรมในการสร้างหน้าแสดงผล CLOSE

หมายเหตุ เนื้อหาของโปรแกรมในหัวข้อ 3.3 นั้นทำการแสดงอยู่ในส่วนของภาคผนวก

3.4 ข้อมูลต่างๆ ที่ใช้ในการสร้างโปรแกรมแสดงผล GUI ประกอบด้วยข้อมูลดังต่อไปนี้

- 1) ข้อมูล m-file ของตัวอย่างโปรแกรม matlab ก่อนที่จะสร้างเป็นโปรแกรมแสดงผลเก็บไว้ใน Folder ชื่อ matlab มี 9 CHAPTER ซึ่งประกอบไปด้วย
 - 1.1) Folder ชื่อ CHAPTER 1 มีข้อมูล m-file ชื่อ IP_01_03 ถึง IP_01_08, ANALYTIC, DIS_SPCT, ENV_PHAS, ENV-PHAS, FFTSEQ, FSERIES, FSERIEST, LAMBDA, LAMBDAAB, LOWEQ, NORMAL, POWER, QUADCOMP, RECT, RECTAB, SIGNUM, SINC, SINC2
 - 1.2) Folder ชื่อ CHAPTER 2 มีข้อมูล m-file ชื่อ IP_02_02 ถึง IP_02_10, GAUS_MAR, GNGAUSS, MULTI_GP, RX_EST, VARIANCE
 - 1.3) Folder ชื่อ CHAPTER 3 มีข้อมูล m-file ชื่อ AM, AM_DEM, AM_MOD, ANALYTIC, DSB1, DSB2, DSB3, DSB_DEM, DSB_MOD, ENV_PHAS, FFTSEQ, FM1, FM2, LOWEQ, LSSB, LSSB_DEM, LSSB_MOD, POWER, USSB_MOD
 - 1.4) Folder ชื่อ CHAPTER 4 มีข้อมูล m-file ชื่อ IP_04_04 , IP_04_09 , IP_04_10, 2ENTROPY, CENTROID, ENTROPY, ENTROPY2, EXPECTED, HUFFMAN, IL4_8FUN, INVMULAW, LLOYDMAX, MSE_DIST,

MULA_PCM, MULAW, NORMAL, SIGNUM, U_PCM, UQ_DIST, UQ_MDPNT, VARIANCE

1.5) Folder ชื่อ CHAPTER 5 มีข้อมูล m-file ชื่อ IP_05_03 ถึง IP_05_11, GNGAUSS, BDT_INT, BDT_INT2, QFUNCT, SMLDP510, SMLDPE 54 ถึง SMLDPE 59

1.6) Folder ชื่อ CHAPTER 6 มีข้อมูล m-file ชื่อ IP_06_01 ถึง IP_06_12, GNGAUSS, XRC

1.7) Folder ชื่อ CHAPTER 7 มีข้อมูล m-file ชื่อ IP_07_01 ถึง IP_07_10, CM_DPSKE, QFUNCT, GNGAUSS, CM_F34, CM_SM32, CM_SM34, CM_SM41, CM_SM52

1.8) Folder ชื่อ CHAPTER 8 มีข้อมูล m-file ชื่อ IP_08_01 ถึง IP_08_12, BIN2DECI, CAPEXMPL, CNV_ENCD, DECI2BIN, ENTROPY, ENTROPY2, IL3_8FUN, METRIC, NXT_STAT, P_E_HD_A, P_E_HD_O, P_E_SD_A, P_E_SD_O, Q, VITERB~1, VITERBI

1.9) Folder ชื่อ CHAPTER 9 มีข้อมูล m-file ชื่อ IP_09_04 ถึง IP_09_07, GNGAUSS, SS42, SS_MLSRS, SS_PE42, SS_PE94, SS_PE96, SS_PE97, SS_TBL31

หมายเหตุ หน้าแสดงผลข้อมูล m-file นี้ มีการเชื่อมโยงกับหน้าแสดงผลข้อมูล Microsoft FrontPage

2) ข้อมูล m-file และ fig-file ของหน้าแสดงผลโปรแกรม GUI มี file ดังต่อไปนี้

2.1) ข้อมูล m-file มี file ชื่อ BDT_INT, BDT_INT2, chapter_01 ถึง chapter_09, CM_DPSKE, CM_F34, CM_SM32, CM_SM34, CM_SM41, CM_SM52, digital_transmission, GNGAUSS, gui_ip_01_01 ถึง gui_ip_01_08, gui_ip_02_01 ถึง gui_ip_02_09, gui_ip_03_01 ถึง gui_ip_03_09, gui_ip_04_01 ถึง gui_ip_04_02, gui_ip_05_01 ถึง gui_ip_05_11, gui_ip_06_01 ถึง gui_ip_06_12, gui_ip_07_01 ถึง gui_ip_07_10, gui_ip_08_01 ถึง gui_ip_08_12, gui_ip_09_01 ถึง gui_ip_09_04, menu, modaldlg, problem_05_01 ถึง problem_05_11, problem_06_01 ถึง problem_06_12, problem_07_01 ถึง problem_07_10, QFUNCT, SMLDP510, SMLDPE54 ถึง SMLDPE59, ANALYTIC,

DIS_SPCT, ENV_PHAS, ENV-PHAS, FFTSEQ, FSERIES, FSERIEST, LAMBDA, LAMBDAAB, LOWEQ, NORMAL, POWER, QUADCOMP, RECT, RECTAB, SIGNUM, SINC, SINC2, GAUS_MAR, GNGAUSS, MULTI_GP, RX_EST, VARIANCE, AM, AM_DEM, AM_MOD, ANALYTIC, DSB1, DSB2, DSB3, DSB_DEM, DSB_MOD, ENV_PHAS, FFTSEQ, FM1, FM2, LOWEQ, LSSB, LSSB_DEM, LSSB_MOD, POWER, USSB_MOD, 2ENTROPY, CENTROID, ENTROPY, ENTROPY2, EXPECTED, HUFFMAN, IL4_8FUN, INVMULAW, LLOYDMAX, MSE_DIST, MULA_PCM, MULAW, NORMAL, SIGNUM, U_PCM, UQ_DIST, UQ_MDPNT, VARIANCE, BIN2DECI, CAPEXMPL, CNV_ENCD, DECI2BIN, ENTROPY, ENTROPY2, IL3_8FUN, METRIC, NXT_STAT, P_E_HD_A, P_E_HD_O, P_E_SD_A, P_E_SD_O, Q, VITERB~1, VITERBI, GNGAUSS, SS42, SS_MLSRS, SS_PE42, SS_PE94, SS_PE96, SS_PE97, SS_TBL31

2.2) ข้อมูล fig-file มี file ชื่อ CHAPTER_01 ถึง CHAPTER_03, DIGITAL_TRANSMISSION, GUI_IP_01_01 ถึง GUI_IP_01_08, GUI_IP_02_01 ถึง GUI_IP_02_09, GUI_IP_03_01 ถึง GUI_IP_03_09, GUI_IP_04_01 ถึง GUI_IP_04_02, GUI_IP_05_01 ถึง GUI_IP_05_11, GUI_IP_06_01 ถึง GUI_IP_06_12, GUI_IP_07_01 ถึง GUI_IP_07_10, GUI_IP_08_01 ถึง GUI_IP_08_12, GUI_IP_09_01 ถึง GUI_IP_09_04, MENU, PROBLEM_01_01 ถึง PROBLEM_01_08, PROBLEM_02_01 ถึง PROBLEM_02_09, PROBLEM_03_01 ถึง PROBLEM_03_09, PROBLEM_04_01 ถึง PROBLEM_04_02, PROBLEM_05_01 ถึง PROBLEM_05_11, PROBLEM_06_01 ถึง PROBLEM_06_12, PROBLEM_07_01 ถึง PROBLEM_07_10, PROBLEM_08_01 ถึง PROBLEM_08_12, PROBLEM_09_01 ถึง PROBLEM_09_04

หมายเหตุ ข้อมูลของโปรแกรมแสดงผลนี้ มีการเชื่อมโยงกับหน้าแสดงผลข้อมูล Microsoft FrontPage, มีการเชื่อมโยงกับข้อมูลหน้าแสดงผลโปรแกรมด้วยตัวเอง และใช้ในการแสดงผลของลักษณะเส้นกราฟ

- 3) ข้อมูลของโปรแกรม Microsoft FrontPage มี file ชื่อ P_01_01 ถึง P_01_08, T_01_01 ถึง T_01_08, P_02_01 ถึง P_02_09, T_02_01 ถึง T_02_09, P_03_01 ถึง P_03_09, T_03_01 ถึง T_03_09, P_04_01 ถึง P_04_02, T_04_01 ถึง T_04_02, P_05_01 ถึง P_05_11, T_05_01 ถึง T_05_11, P_06_01 ถึง P_06_12, T_06_01 ถึง T_06_12, P_07_01 ถึง P_07_10, T_07_01 ถึง T_07_10, P_08_01 ถึง P_08_12, T_08_01 ถึง T_08_12, P_09_01 ถึง P_09_04, T_09_01 ถึง T_09_04 และ S_01

หมายเหตุ ข้อมูลของหน้าแสดงผลโปรแกรม Microsoft FrontPage มีการเชื่อมโยงกับข้อมูลหน้าแสดงผลโปรแกรม GUI, มีการเชื่อมโยงกับข้อมูลหน้าแสดงผล m-file ของตัวอย่างโปรแกรม matlab ก่อนที่จะสร้างเป็นโปรแกรมแสดงผล, มีการเชื่อมโยงกับข้อมูลหน้าแสดงผล คำอธิบายความหมายสัญลักษณ์, มีการเชื่อมโยงกับข้อมูลหน้าแสดงผลโปรแกรม Microsoft FrontPage ด้วยตัวเอง

- 4) ข้อมูลของรูปภาพเก็บไว้ใน Folder ชื่อ Picture

หมายเหตุ ข้อมูลของรูปภาพมีการเชื่อมโยงกับข้อมูลหน้าแสดงผลโปรแกรม Microsoft FrontPage

บทที่ 4

ผลของการจำลองแบบ

คู่มือการใช้งาน

โปรแกรมช่วยศึกษาโปรแกรมช่วยศึกษาและวิจัยสำหรับระบบวิศวกรรมการสื่อสาร

4.1 กล่าวนำ

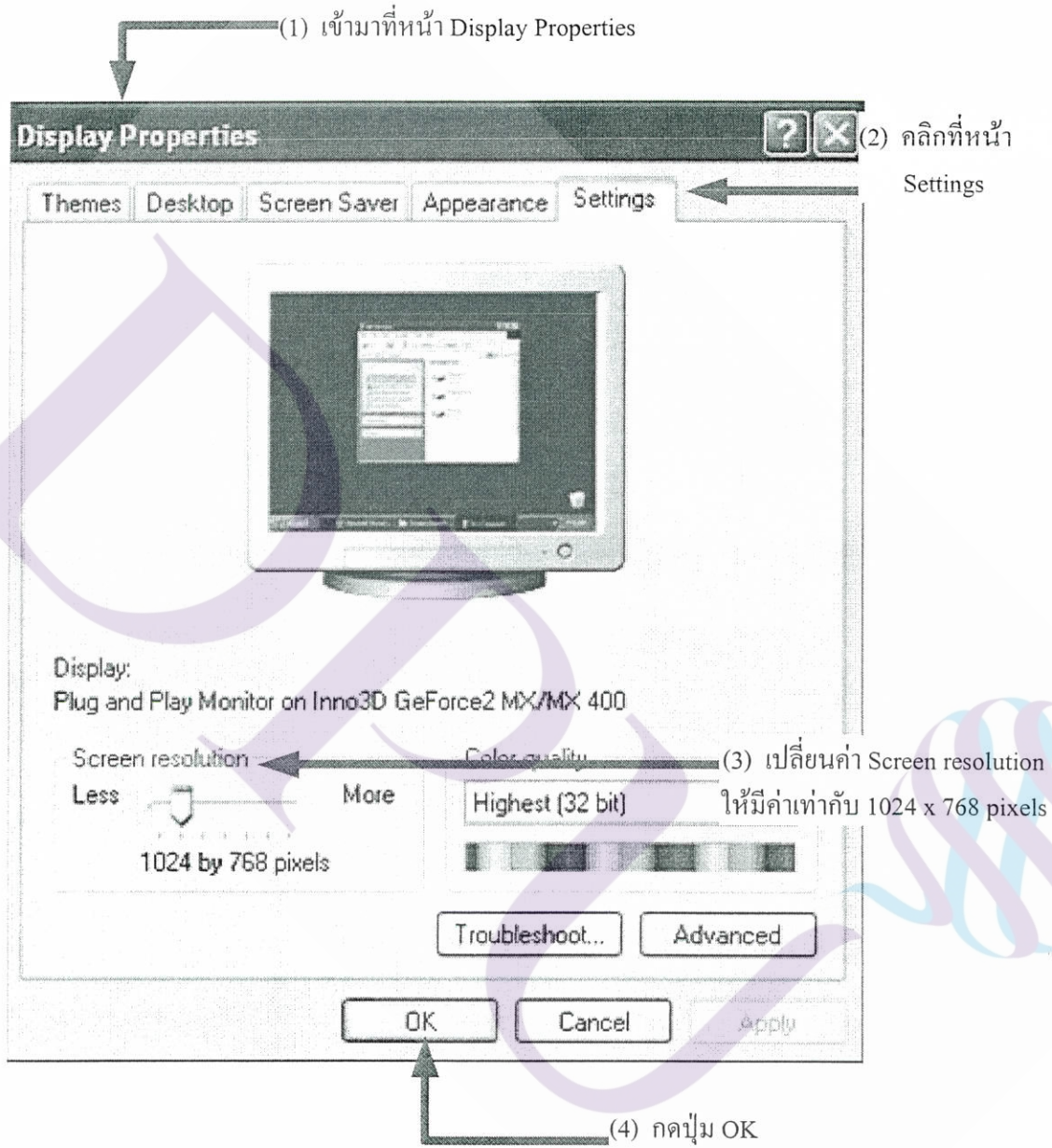
ในบทที่ 4 นี้ เป็นการกล่าวถึงผลที่ได้จากการสร้างแบบจำลอง ตลอดจนวิธีการใช้งานของแบบจำลองโปรแกรมช่วยศึกษาโปรแกรมช่วยศึกษาและวิจัยสำหรับระบบวิศวกรรมการสื่อสารด้านแอนะล็อก ดิจิทัล และสเปกตรัม ที่ทำการออกแบบและสร้างให้ใช้งานแบบ GUI โดยจะอธิบายการทำงานของหน้าต่างแสดงผลแบบจำลอง ตั้งแต่การเริ่มต้นการเข้าสู่โปรแกรม จนถึงจุดสิ้นสุดการทำงานของโปรแกรม คือ จะได้รับกราฟผลลัพธ์จากการ RUN โปรแกรม เช่น อธิบายการเชื่อมโยงของแต่ละหน้าต่าง การปุ่มกดต่าง ๆ ของหน้าต่าง GUI เป็นต้น

4.2 วิธีการใช้งานแบบจำลองโปรแกรม GUI

สิ่งที่สำคัญมาก ในการเปิดโปรแกรม SRAPCES นี้ คือ ต้องใช้โปรแกรม MATLAB 7.0 ขึ้นไปเท่านั้น จึงจะสามารถใช้งานได้

วิธีการใช้งาน โปรแกรม GUI ที่จะนำเสนอต่อไปนี้เป็นแค่บางส่วนเท่านั้น ซึ่งการใช้โปรแกรม GUI มีวิธีการดังต่อไปนี้

1. ตั้งค่าหน้า Display Properties ดังต่อไปนี้

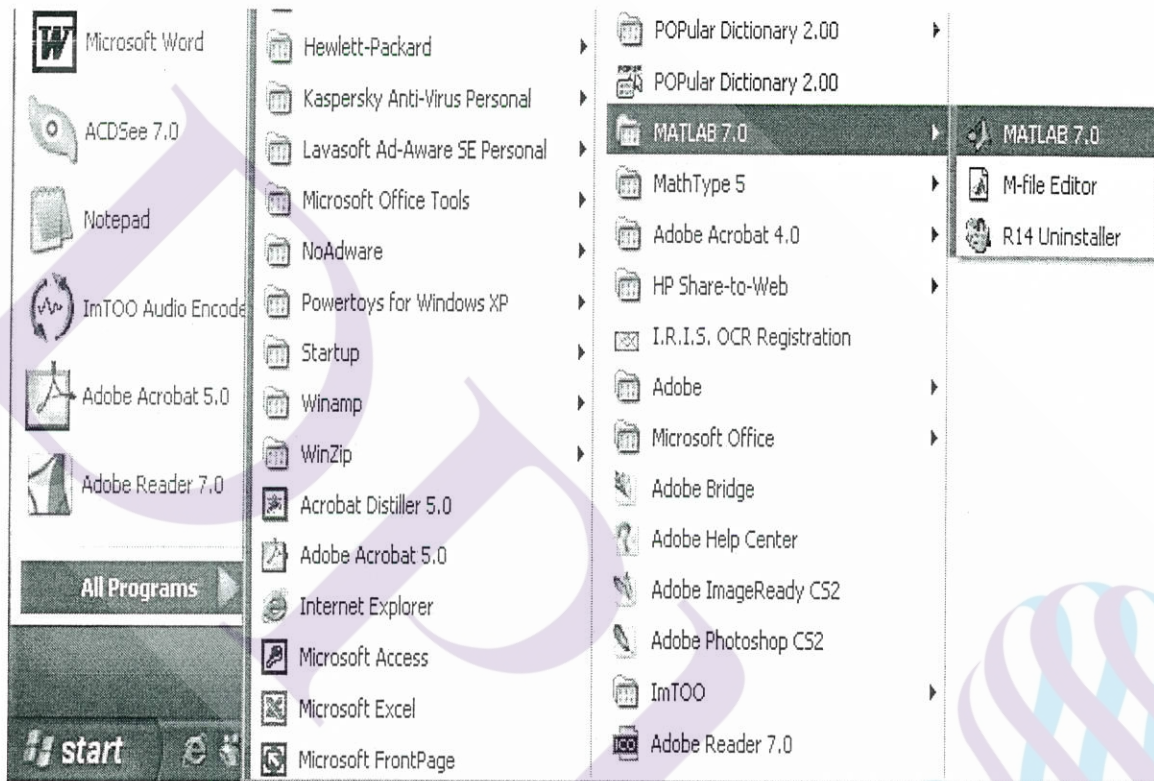


รูปที่ 4.1 การตั้งค่าหน้า Display Properties

2. คลิกที่ปุ่ม



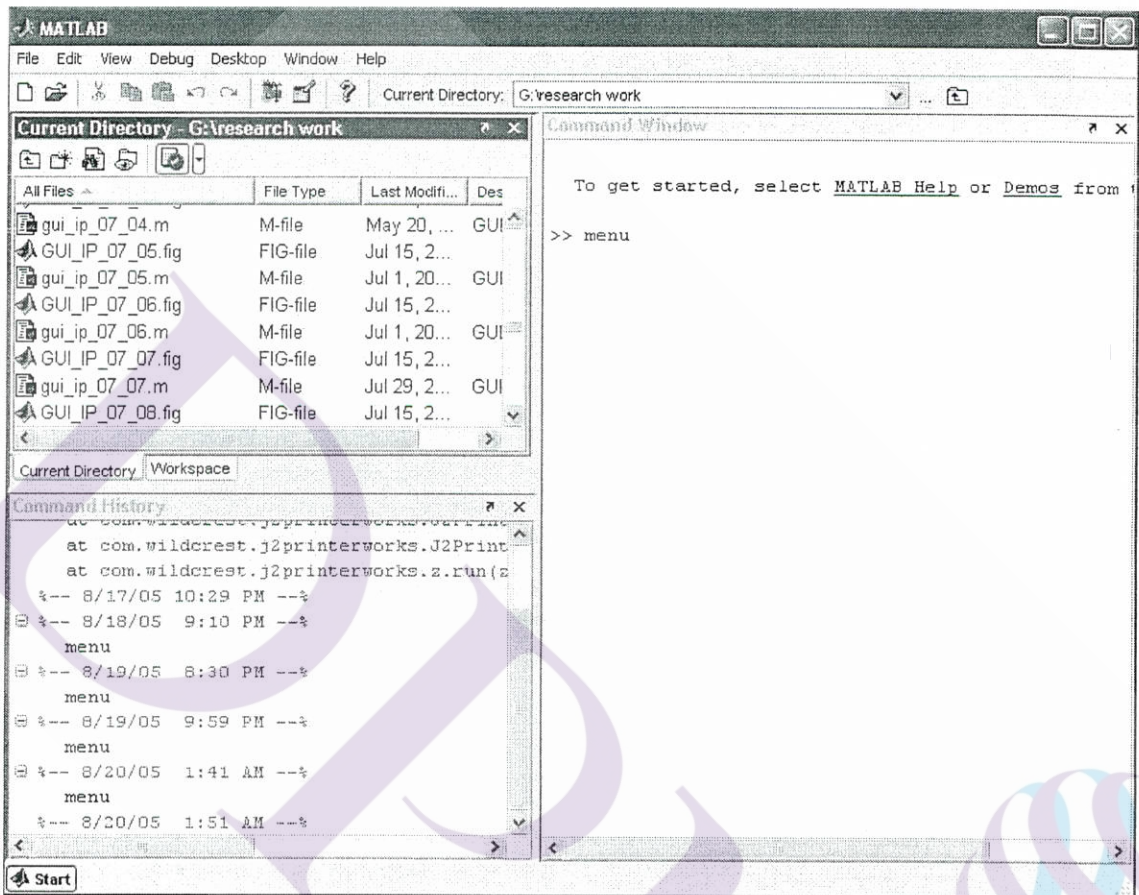
3. เลื่อนเมาส์ไปที่ All Programs ซึ่งจะมีเมนูย่อยของโปรแกรมต่าง ๆ นั้นปรากฏขึ้น ให้เลื่อนเมาส์ไปที่โปรแกรม MATLAB 7.0 จะมีเมนูย่อยของโปรแกรม MATLAB 7.0 ปรากฏขึ้น และต่อจากนั้นให้คลิกที่ MATLAB 7.0 ดังรูปที่ 4.2



รูปที่ 4.2 การเปิดโปรแกรม MATLAB 7.0

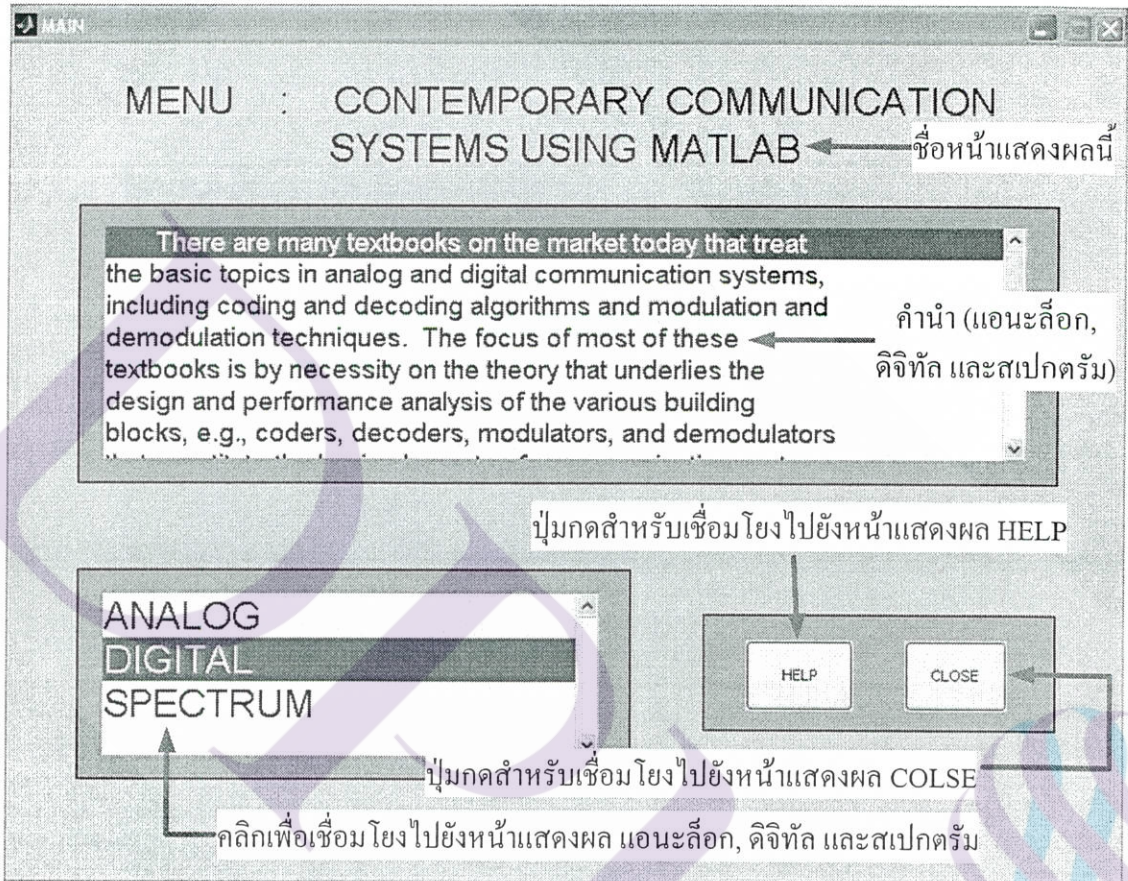
หมายเหตุ สำหรับหัวข้อ 3 นี้เป็นวิธีการใช้งานสำหรับ Windows XP

4. เปิดไควร์ข้อมูลที่อยู่ใน Current Directory ดังรูปที่ 4.3



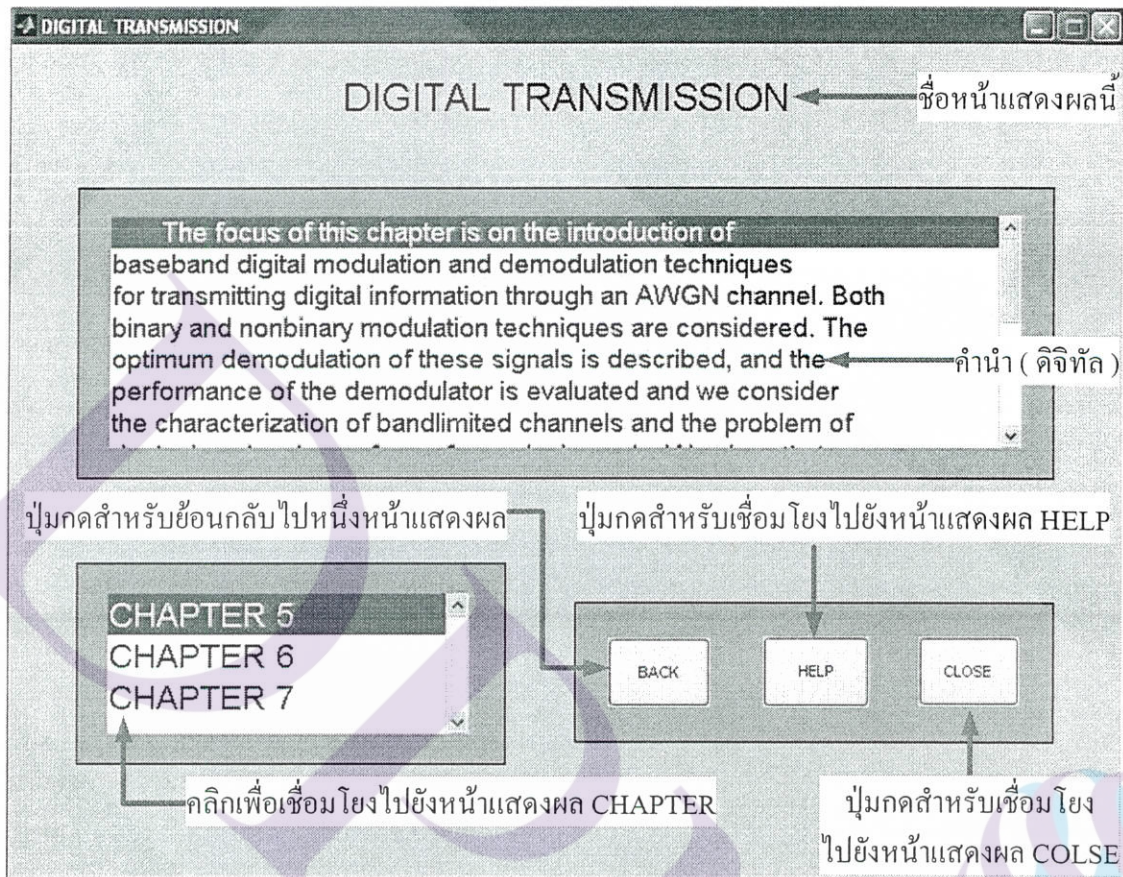
รูปที่ 4.3 การเปิดไควร์ข้อมูลที่อยู่ใน Current Directory และการเปิดหน้าต่าง menu

5. พิมพ์คำว่า menu ลงใน Command Window ดังรูปที่ 4.3 แล้วกด Enter จะปรากฏหน้าต่างแสดงผลดังรูปที่ 4.4



รูปที่ 4.4 การเชื่อมโยงของหน้าต่างแสดงผล MENU

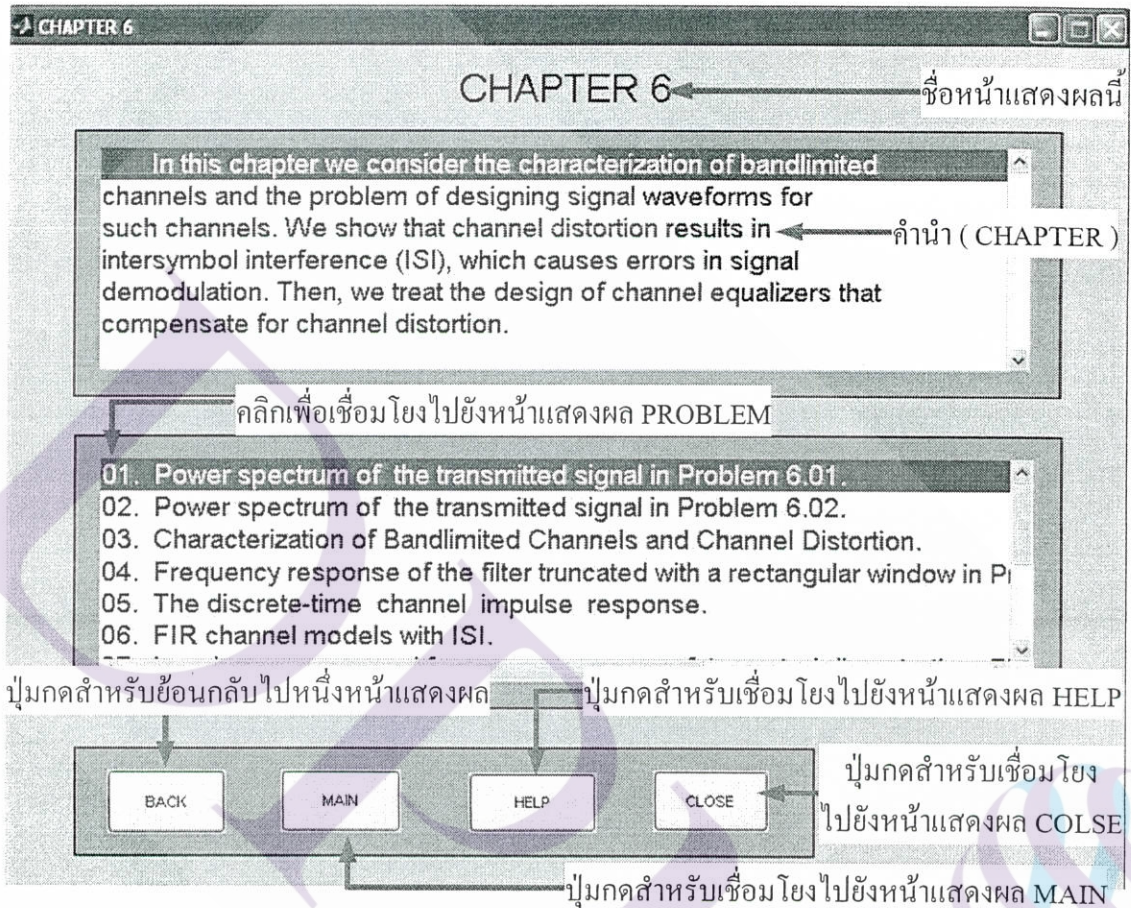
6. ยกตัวอย่างการคลิกที่ DIGITAL ซึ่งจะปรากฏหน้าแสดงผลดังรูปที่ 4.5



รูปที่ 4.5 การเชื่อมโยงของหน้าแสดงผล DIGITAL TRANSMISSION

หมายเหตุ ส่วนหน้าแสดงผล ANALOG TRANSMISSION และ SPECTRUM TRANSMISSION นั้น จะมีวิธีการใช้งานเหมือนกับตัวอย่างของหน้าแสดงผล DIGITAL TRANSMISSION ที่นำเสนอมา

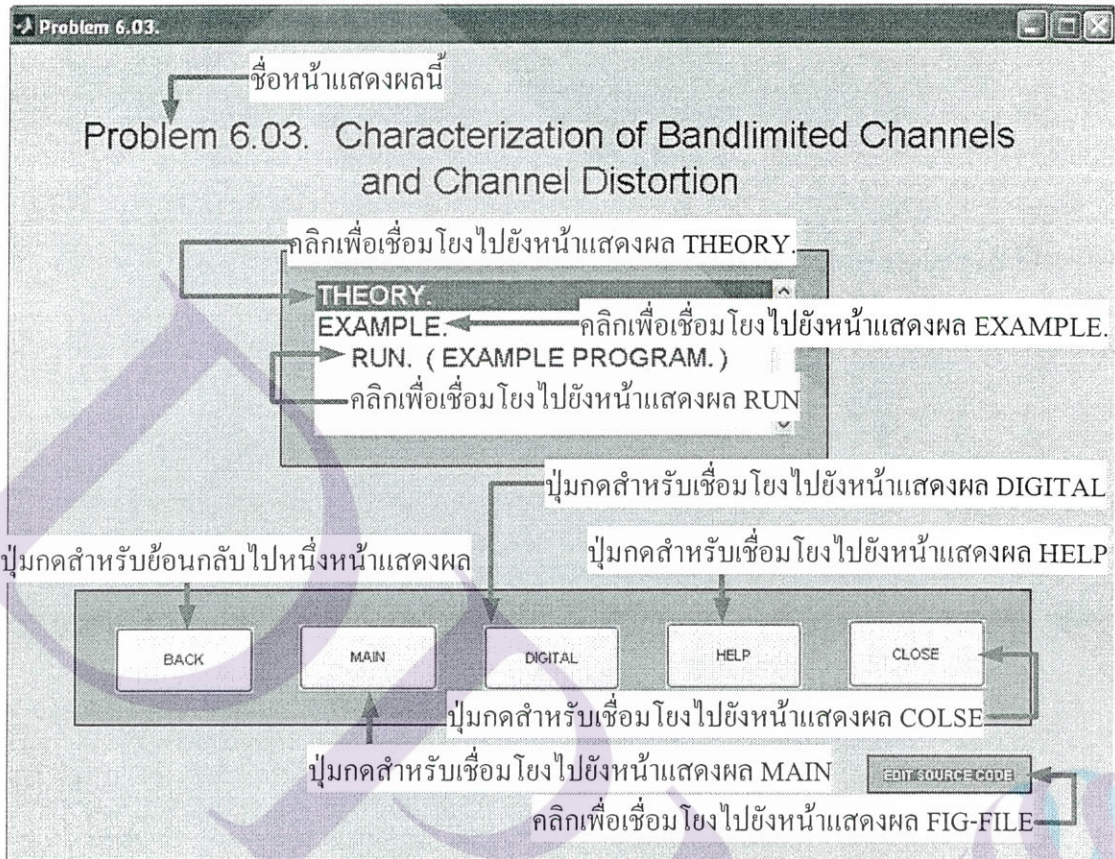
7. ยกตัวอย่างการคลิกที่ CHAPTER 6 ซึ่งจะปรากฏหน้าแสดงผลดังรูปที่ 4.6



รูปที่ 4.6 การเชื่อมโยงของหน้าแสดงผล CHAPTER 6

หมายเหตุ ส่วนหน้าแสดงผล CHAPTER อื่น ๆ นั้น จะมีวิธีการใช้งานเหมือนกับตัวอย่างของหน้าแสดงผล CHAPTER 6 ที่นำเสนอมา

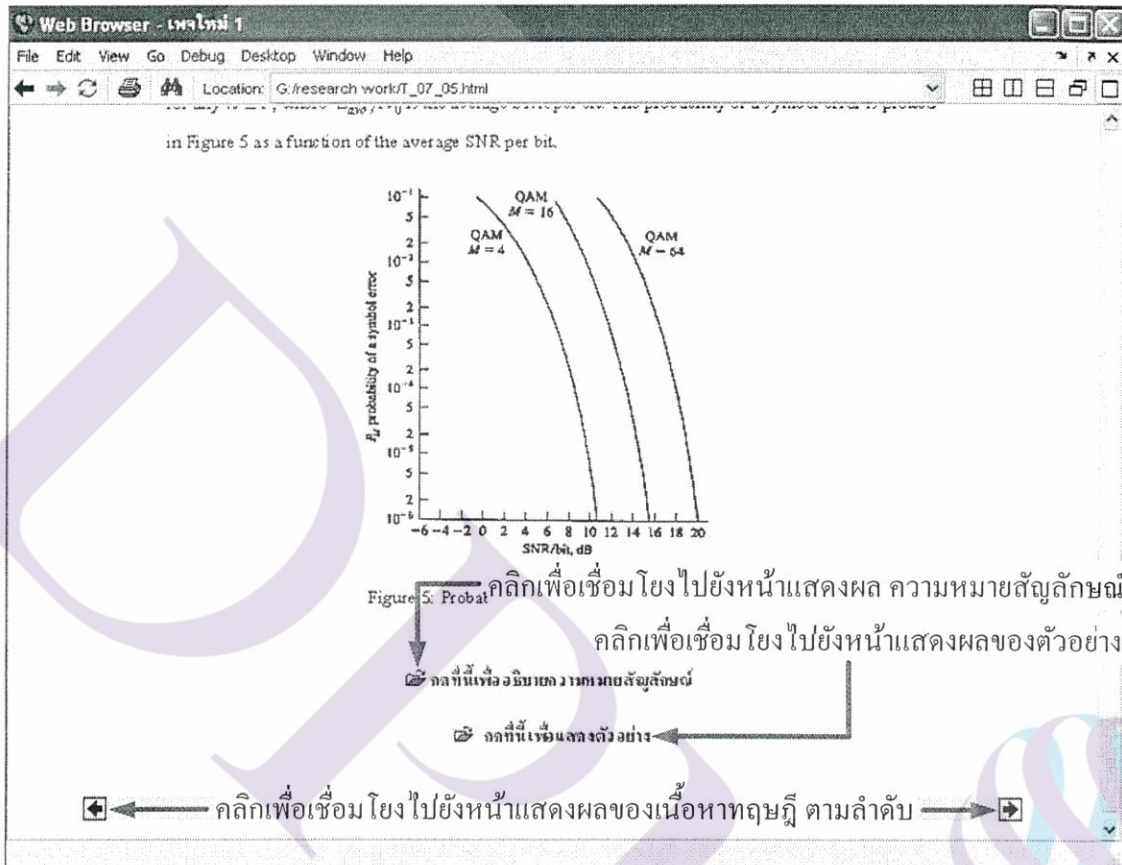
8. ยกตัวอย่างการคลิกหัวข้อที่ 3 ซึ่งจะปรากฏหน้าแสดงผลดังรูปที่ 4.7



รูปที่ 4.7 การเชื่อมโยงของหน้าแสดงผล Problem 6.03

หมายเหตุ ส่วนหน้าแสดงผล Problem อื่น ๆ นั้น จะมีวิธีการใช้งานเหมือนกับตัวอย่างของหน้าแสดงผล Problem 6.03 ที่นำเสนอมา

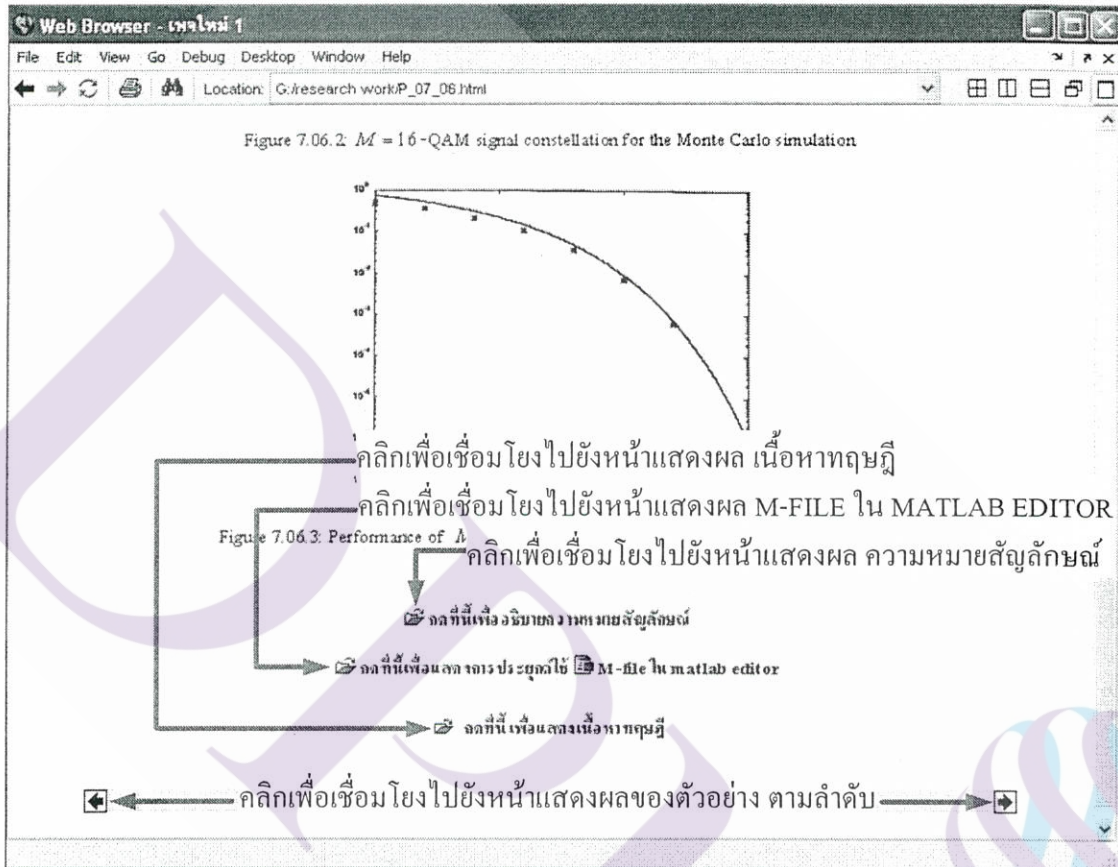
9. จากรูปที่ 8 การเชื่อมโยงของหน้าแสดงผล Problem 6.03 เมื่อเราทำการยกตัวอย่างการคลิกที่ THEORY ของ Problem 6.03 จะปรากฏหน้าแสดงผลดังรูปที่ 4.8



รูปที่ 4.8 การเชื่อมโยงของหน้าแสดงผล THEORY

หมายเหตุ ส่วนหน้าแสดงผล THEORY อื่น ๆ นั้น จะมีวิธีการใช้งานเหมือนกับตัวอย่างของหน้าแสดงผล THEORY จากรูป 4.8 ที่นำเสนอมา

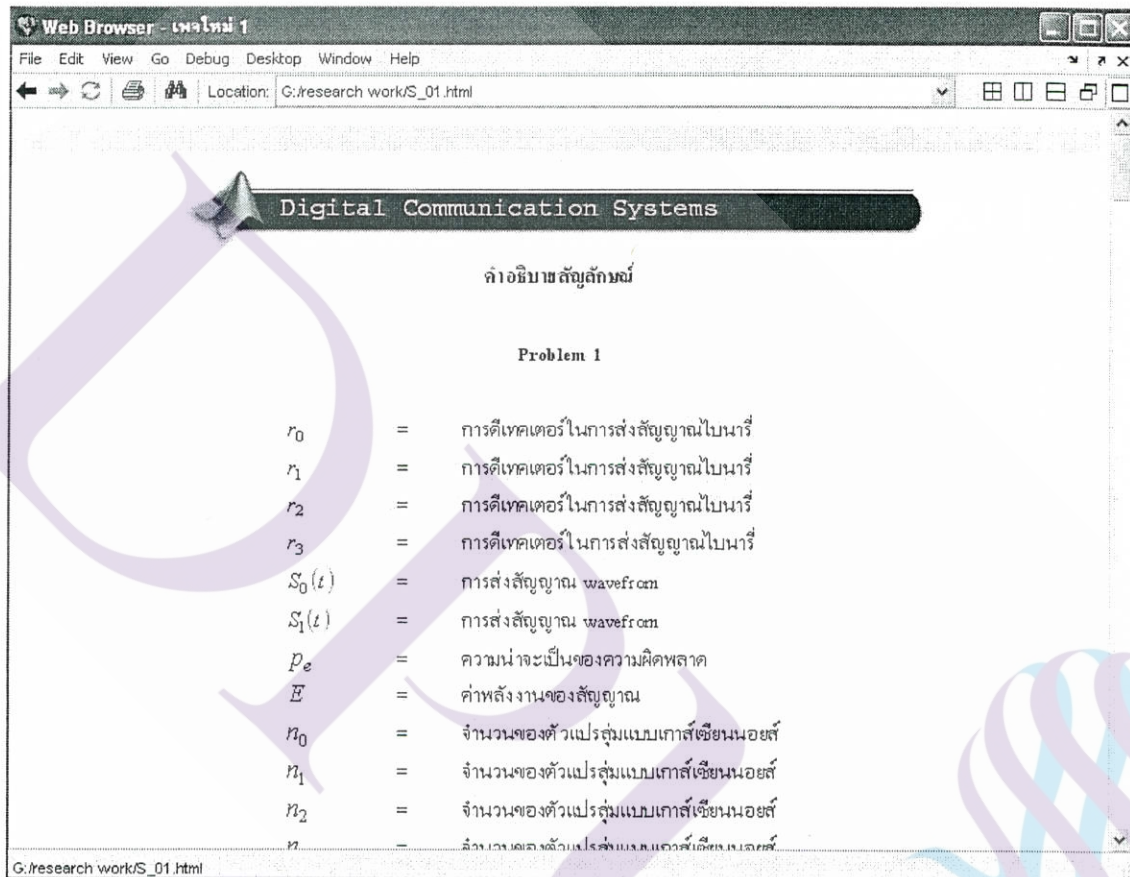
10. จากรูปที่ 8 การเชื่อมโยงของหน้าแสดงผล Problem 6.03 เมื่อเราทำการยกตัวอย่างการคลิกที่ EXAMPLE ของ Problem 6.03 จะปรากฏหน้าแสดงผลดังรูปที่ 4.9



รูปที่ 4.9 การเชื่อมโยงของหน้าแสดงผล EXAMPLE

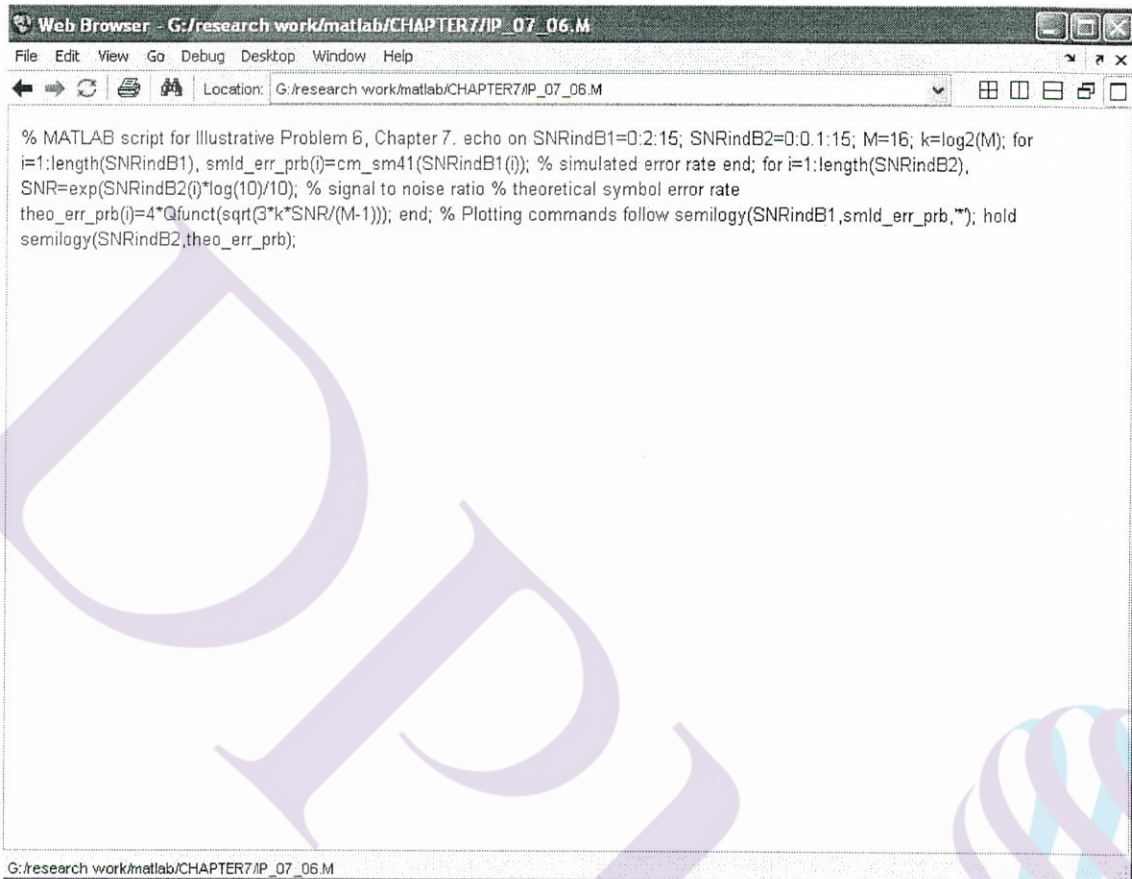
หมายเหตุ ส่วนหน้าแสดงผล EXAMPLE อื่น ๆ นั้น จะมีวิธีการใช้งานเหมือนกับตัวอย่างของหน้าแสดงผล EXAMPLE จากรูป 4.9 ที่นำเสนอมา

11. จากรูปที่ 9 การเชื่อมโยงของหน้าแสดงผล THEORY และ รูปที่ 4.9 การเชื่อมโยงของหน้าแสดงผล EXAMPLE เมื่อเราทำการกดตัวอย่างการคลิก “กดที่นี่เพื่ออธิบายความหมายสัญลักษณ์” จะปรากฏหน้าแสดงผลดังรูปที่ 4.10



รูปที่ 4.10 การเชื่อมโยงของหน้าแสดงผล คำอธิบายความหมายสัญลักษณ์

12. จากรูปที่ 4.9 การเชื่อมโยงของหน้าแสดงผล EXAMPLE เมื่อเราทำการยกตัวอย่างการคลิก “กดที่นี่เพื่อแสดงการประยุกต์ใช้ M-file ใน matlab editor” จะปรากฏหน้าแสดงผลดังรูปที่ 4.11



```

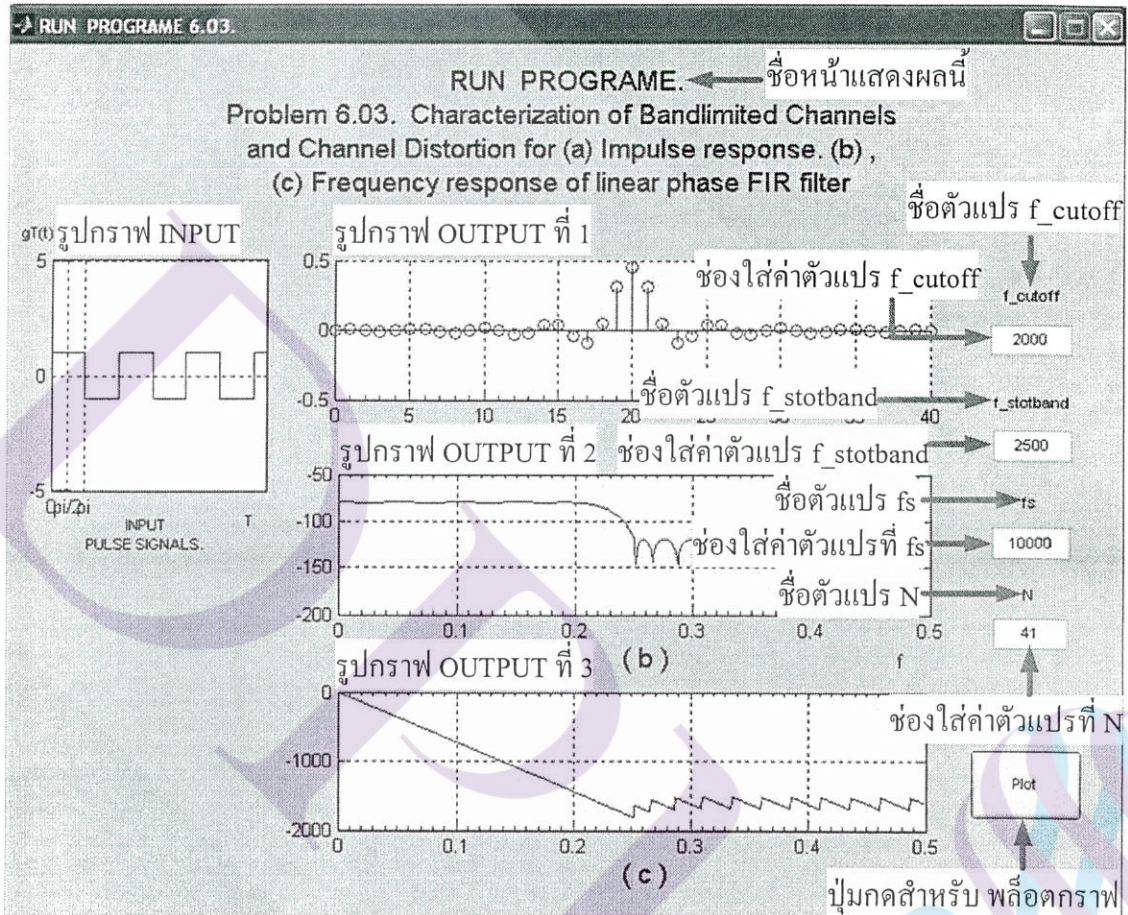
% MATLAB script for Illustrative Problem 6, Chapter 7. echo on SNRindB1=0:2:15; SNRindB2=0:0.1:15; M=16; k=log2(M); for
i=1:length(SNRindB1), smld_err_prb(i)=cm_sm41(SNRindB1(i)); % simulated error rate end; for i=1:length(SNRindB2),
SNR=exp(SNRindB2(i)*log(10)/10); % signal to noise ratio % theoretical symbol error rate
theo_err_prb(i)=4*Qfunct(sqrt(3*k*SNR/(M-1))); end; % Plotting commands follow semilogy(SNRindB1,smld_err_prb,'); hold
semilogy(SNRindB2,theo_err_prb);

```

รูปที่ 4.11 การเชื่อมโยงของหน้าแสดงผล การประยุกต์ใช้ M-file ใน matlab editor

หมายเหตุ ส่วนหน้าแสดงผล ประยุกต์ใช้ M-file ใน matlab editor อื่น ๆ นั้น จะมีวิธีการใช้งาน เหมือนกับตัวอย่างของหน้าแสดงผล ประยุกต์ใช้ M-file ใน matlab editor จากรูป 4.11 ที่นำเสนอมา

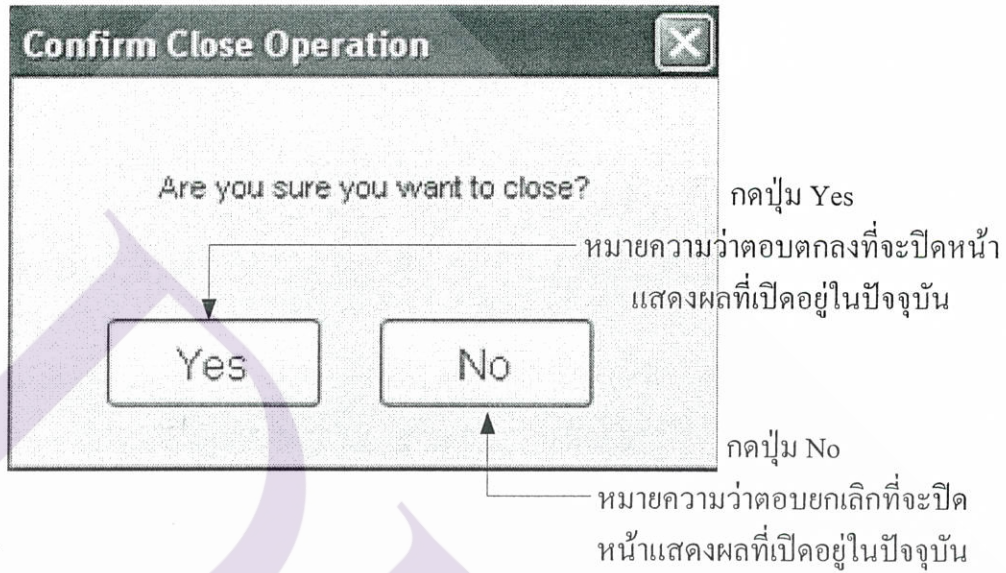
13. จากรูปที่ 4.7 การเชื่อมโยงของหน้าต่างแสดงผล Problem 6.03 เมื่อเราทำการยกตัวอย่างการคลิกที่ RUN. (EXAMPLE PROGRAM) ของ Problem 6.03 จะปรากฏหน้าต่างแสดงผลดังรูปที่ 4.12



รูปที่ 4.12 การเชื่อมโยงของหน้าต่างแสดงผล RUN PROGRAM

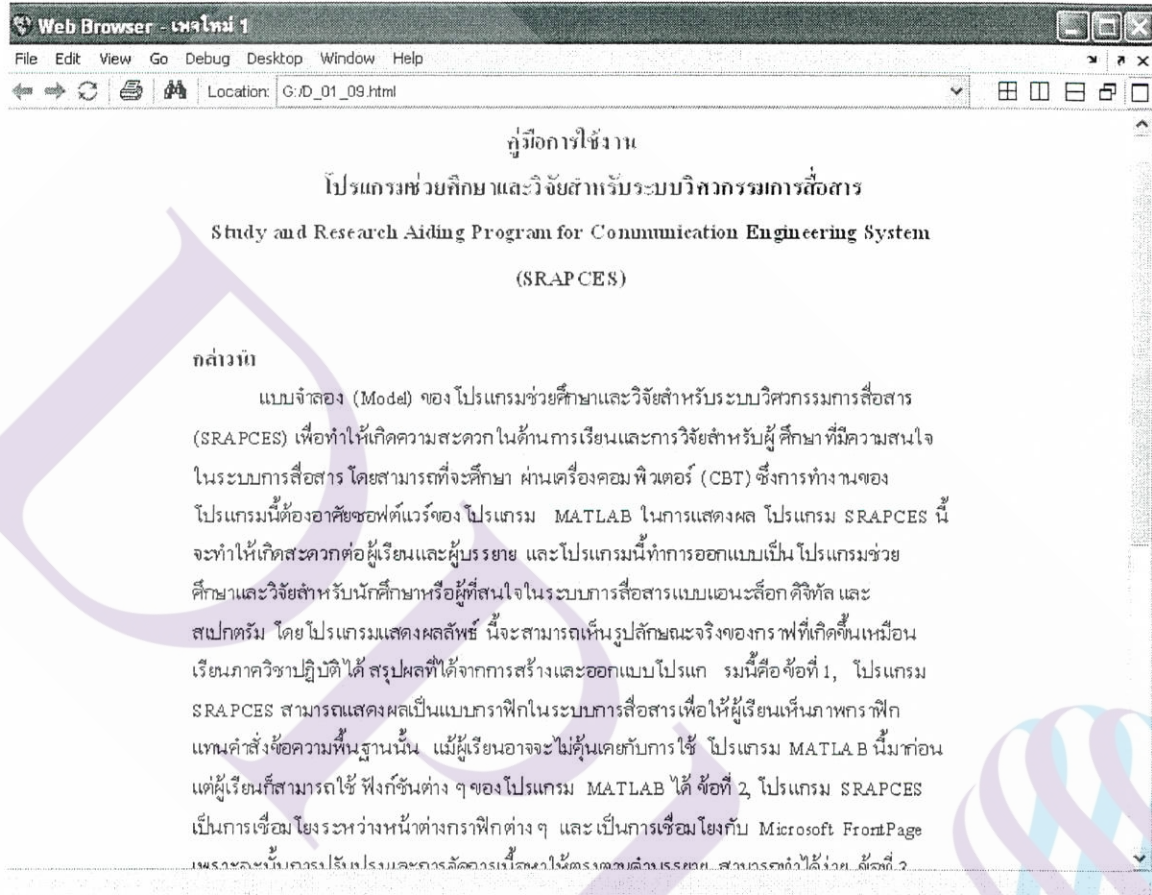
หมายเหตุ ส่วนหน้าต่างแสดงผล RUN PROGRAM อื่น ๆ นั้น จะมีวิธีการใช้งานเหมือนกับตัวอย่างของหน้าต่างแสดงผล RUN PROGRAM 6.03 ที่นำเสนอมา

14. จากรูปหน้าแสดงผลที่มีปุ่มกด CLOSE นั้น เมื่อทำการกดปุ่ม CLOSE จะปรากฏหน้าต่างแสดงผลดังรูปที่ 4.13



รูปที่ 4.13 การเชื่อมโยงของหน้าต่างแสดงผล CLOSE

15. จากรูปหน้าแสดงผลที่มีปุ่มกด HELP นั้น เมื่อทำการกดปุ่ม HELP จะปรากฏหน้าแสดงผล ดังรูปที่ 4.14



รูปที่ 4.14 การเชื่อมโยงของหน้าแสดงผล HELP

บทที่ 5

สรุปผลการจำลองแบบและข้อเสนอแนะ

5.1 สรุปผลการจำลองแบบ

งานวิจัยโปรแกรม SRAPCES ได้ทำการสร้างและออกแบบให้แสดงผลคุณลักษณะของกราฟ โดยใช้ GUI (Graphical User Interfaces) ของโปรแกรม MATLAB โดยทำการอ้างอิงมาจากหนังสือ ชื่อ CONTEMPORARY COMMUNICATION SYSTEMS USING MATLAB และในหนังสือมี ข้อมูลของ m-file ที่สามารถนำมาสร้างเป็น โปรแกรมแสดงผลได้

จากโปรแกรม SRAPCES นี้สามารถทำการทดสอบความถูกต้องของแบบจำลองที่ออกแบบมานั้น โดยทำการเปรียบเทียบผลลัพธ์ของเส้นกราฟที่ได้จากแบบจำลอง GUI (Graphical User Interfaces) ที่ออกแบบไว้กับตัวอย่างผลลัพธ์ของเส้นกราฟที่ได้จากการ RUN โปรแกรม MATLAB ที่มี อยู่ในหนังสือ CONTEMPORARY COMMUNICATION SYSTEMS USING MATLAB โดยผลของการเปรียบเทียบพบว่าผลลัพธ์ของเส้นกราฟที่ได้จากการออกแบบจำลอง GUI กับตัวอย่างผลลัพธ์ของเส้นกราฟในหนังสือ ซึ่งผลลัพธ์เส้นกราฟทั้งสองนั้นจะมีลักษณะเส้นกราฟที่เหมือนกัน เมื่อค่าพารามิเตอร์ของตัวแปรมีค่าเท่ากัน เพราะฉะนั้นจึงเป็นข้อพิสูจน์ได้ว่าการสร้างแบบจำลอง โดยใช้โปรแกรม GUI ในการแสดงผลมีความถูกต้องตามทฤษฎี

ผลที่ได้จากการสร้างและออกแบบโปรแกรม SRAPCES คือสามารถแสดงผลคุณลักษณะของเส้นกราฟได้, มีการเชื่อมโยงระหว่างหน้าแสดงผลของโปรแกรม GUI ด้วยกันเอง, หน้าแสดงผลโปรแกรม GUI มีการเชื่อมโยงร่วมกับหน้าแสดงผลโปรแกรม Microsoft FrontPage, หน้าแสดงผลโปรแกรม GUI มีการเชื่อมโยงร่วมกับ m-file ในส่วนก่อนที่จะนำมาสร้างเป็นโปรแกรมแสดงผล

โปรแกรม SRAPCES นี้ทำการสร้างขึ้นมาเพื่อที่จะนำมาใช้ในระบบงานการศึกษาและวิจัย ด้านวิศวกรรมสื่อสารหรือโทรคมนาคม ซึ่งโปรแกรม GUI ที่ทำการออกแบบและสร้างขึ้นมา นี้สามารถที่จะนำไปใช้งานได้จริง เพื่อให้เกิดความรู้ความเข้าใจในบทเรียนมากขึ้น และโปรแกรม GUI นี้ยังสามารถทำการเปรียบเทียบโดยการคำนวณหาผลลัพธ์ที่ได้จากรูปกราฟจากสูตรคำนวณที่เกิดขึ้นได้ นอกจากนี้ยังสามารถทำการเปลี่ยนแปลงค่าพารามิเตอร์ต่าง ๆ ในแต่ละโปรแกรมของโปรแกรม GUI เพื่อให้เกิดผลลัพธ์ที่แตกต่างกันออกไปได้

5.2 ข้อเสนอแนะ

สำหรับข้อเสนอแนะในการสร้างแบบจำลองโดยใช้โปรแกรม GUI เพื่อแสดงผลลัพธ์นั้น ต้องอาศัยความเข้าใจในตัวแบบจำลอง และระบบที่ทำการสร้าง ไม่ว่าจะสร้างแบบจำลองหรือโปรแกรมจะทำได้โดยใช้วิธีใดก็ตาม ต้องอาศัยทฤษฎีมาใช้ในการตรวจสอบแบบจำลองของผลที่ได้จากการจำลองแบบด้วยกันทั้งสิ้น หากผลที่ได้จากการจำลองแบบไม่เป็นไปตามทฤษฎีให้ทำการแก้ไขเปลี่ยนแปลงหรือตัดแปลงแบบจำลองใหม่ ในงานวิจัยเล่มนี้เป็นการสร้างแบบจำลองพร้อมกับมีส่วน GUI สำหรับด้านวิศวกรรมการสื่อสารหรือโทรคมนาคมในส่วนของแอนะล็อก ดิจิทัล และสเปกตรัม และแนวคิดในการสร้างแบบจำลองของงานวิจัยนี้จะส่งผลทำให้เกิดความเข้าใจในการที่จะสร้างแบบจำลองที่มีความซับซ้อนมากกว่านี้ การทำให้เกิดแนวคิดดังกล่าวจะเป็นแรงบันดาลใจให้เกิดตัวงานเช่นเดียวกับงานวิจัยเล่มนี้ นอกจากนี้ควรศึกษาเพิ่มเติมในส่วนของทฤษฎีระบบการสื่อสารแบบแอนะล็อก ดิจิทัล และสเปกตรัม

ส่วนที่ยังไม่สามารถสร้างและออกแบบเป็นโปรแกรมแสดงผลโดยใช้ GUI ได้ นั้นมีดังต่อไปนี้

บทที่ 1 มีหัวข้อ UNDER CONSTRUCTION มีดังนี้

1.02. A periodic signal.

1.03. The magnitude and the phase spectra.

บทที่ 2 มีหัวข้อ UNDER CONSTRUCTION มีดังนี้

2.04. The autocorrelation function.

บทที่ 3 มีหัวข้อ UNDER CONSTRUCTION มีดังนี้

3.03. Amplitude Modulation (AM).

3.07. Envelope detection (AM).

3.08. Frequency modulation 1 (FM1).

3.09. Frequency modulation 2 (FM2).

บทที่ 8 มีหัวข้อ UNDER CONSTRUCTION มีดังนี้

8.08. Linear block codes.

บทที่ 9 มีหัวข้อ UNDER CONSTRUCTION มีดังนี้

9.04. The maximum value of the cross correlation.

ส่วนที่สามารถแสดงผลในทางกราฟิกได้ แต่ไม่สามารถเปลี่ยนแปลงค่าอินพุตได้ เพราะยังไม่มีอินพุต นั้นมีดังต่อไปนี้

บทที่ 7 มีหัวข้อที่ยังไม่มีอินพุต มีดังนี้

7.04. Implement a differential encoder for the case of $M = 8$ DPSK.

บทที่ 8 มีหัวข้อที่ยังไม่มีอินพุต มีดังนี้

8.09. Hamming codes.

ในส่วนที่ยังไม่สามารถสร้างและออกแบบได้นี้ จะทำการพัฒนาให้ดีขึ้นและสมบูรณ์แบบในเฟสต่อไป

ขณะนี้โปรแกรม SRAPCES จะทำงานในการแสดงผลได้ต้องรันต้นซอฟต์แวร์ MATLAB ซึ่งในเฟสต่อไปจะทำให้เป็นโปรแกรมสำเร็จรูป (Execute File) โดยไม่ต้องติดตั้งโปรแกรม MATLAB สิ่งที่สำคัญมากโปรแกรม SRAPCES จะทำงานในการแสดงผลได้ต้องรันต้นซอฟต์แวร์ MATLAB เวอร์ชัน 7.0 ขึ้นไปเท่านั้น

บรรณานุกรม

- [1] บัณฑิต โรจน์อารยานนท์, “หลักการไฟฟ้าสื่อสาร”, พิมพ์ครั้งที่ 7, กรุงเทพฯ: สพจ, 2541.
- [2] ผู้ช่วยศาสตราจารย์ ดร.บงการ หอมนาน, “หลักการสื่อสาร”, พิมพ์ครั้งที่ 1, กรุงเทพฯ: มหาวิทยาลัยธุรกิจบัณฑิต, 2545.
- [3] Proakis, J. ; and Salehi, M., “Contemporary Communication Systems Using Matlab PWS”, Boston, Publishing Company, 1998.





ตารางที่ ก.1 แสดงข้อเสนองานวิจัย

PROPOSAL OF A PROJECT	
Project definition :	งานวิจัยวิศวกรรมเรื่อง SRAPCES (Study and Research Aiding Program for Communication Engineering System)
Project objectives :	เป็นการเขียนและการพัฒนาระบบโปรแกรมคอมพิวเตอร์เบื้องต้น โดยการใช้โปรแกรม MATLAB เพื่อนำมาใช้ในระบบงานการวิจัยในด้านวิศวกรรมการสื่อสาร
Strategy for achieving objectives :	การศึกษาโปรแกรม MATLAB ควบคู่กับหนังสือ CONTEMPORARY COMMUNICATION SYSTEM USING MATLAB เป็นหลัก ซึ่งในหนังสือจะมีตัวอย่างรูปโปรแกรม MATLAB อยู่แล้ว ดังนั้นจึงทำการออกแบบ โปรแกรม SRAPCES โดยใช้ GUI (Graphical User Interface) เพื่อนำมาสร้างเป็นโปรแกรมการประมวลผลในซอฟต์แวร์ MATLAB
Plan of action :	<ol style="list-style-type: none"> 1. ศึกษาตัวอย่างโปรแกรม MATLAB เกี่ยวกับเรื่องระบบการสื่อสารแอนะล็อก, ดิจิทัล และสเปกตรัม โดยอ้างอิงหนังสือ CONTEMPORARY COMMUNICATION SYSTEM USING MATLAB 2. ศึกษาและเรียนรู้เกี่ยวกับเรื่องระบบการสื่อสารแอนะล็อก, ดิจิทัล และสเปกตรัม 3. วิเคราะห์คุณสมบัติของสัญญาณ โดยใช้โปรแกรม MATLAB ช่วยในการวิเคราะห์แก้ปัญหา 4. ออกแบบจำลองระบบสัญญาณให้เป็น โปรแกรมแสดงผล โดยใช้โปรแกรม MATLAB ในการทำ GUI (Graphic User Interface) 5. ทำการทดสอบ โปรแกรมแสดงผลที่ทำการออกแบบขึ้นมา 6. ทำการประเมินและสรุปผลการสร้าง โปรแกรมแสดงผล 7. สรุปผลที่ได้จากผลการทดสอบ 8. เขียนและพิมพ์งานวิจัย
Reporting :	รายงานความคืบหน้าในการปฏิบัติงานวิจัยและประชุมสรุปงานวิจัยทุก 2 สัปดาห์
Budget :	งบประมาณในการทำงานวิจัยวิศวกรรมชิ้นนี้ประมาณ 218,000 บาท

Evaluation :	<ol style="list-style-type: none">1. ตรวจสอบโปรแกรมต่าง ๆ บนพื้นฐานซอฟต์แวร์ MATLAB ตามหนังสือ CONTEMPORARY COMMUNICATION SYSTEM USING MATLAB2. ตรวจสอบโปรแกรม GUI (Graphical User Interface)3. ทำการตรวจสอบโปรแกรมทั้งหมดอีกครั้งจากทั้ง 3 ส่วน คือแอนะล็อก, ดิจิทัล และสเปกตรัม
---------------------	---



CODE ที่ใช้ในการสร้างโปรแกรม SRAPCES โดย GUI ของ MATLAB

1. อ้างอิงจากหัวข้อ 3.3.1 คือ CODE ในการสร้างหน้าต่างแสดงผล MENU


```
function varargout = MENU(varargin)
% MENU Application M-file for MENU.fig
% FIG = MENU launch MENU GUI.
% MENU('callback_name', ...) invoke the named callback.
% Last Modified by GUIDE v2.0 01-Mar-2004 23:00:39
if nargin == 0 % LAUNCH GUI
fig = openfig(mfilename,'reuse');
% Generate a structure of handles to pass to callbacks, and store it.
handles = guihandles(fig);
guidata(fig, handles);
if nargin > 0
varargout{1} = fig;
end
elseif ischar(varargin{1}) % INVOKE NAMED SUBFUNCTION OR CALLBACK
try
if (nargout)
[varargout{1:nargout}] = feval(varargin{:}); % FEVAL switchyard
else
feval(varargin{:}); % FEVAL switchyard
end
catch
disp(lasterr);
end
end
%| ABOUT CALLBACKS:
%| GUIDE automatically appends subfunction prototypes to this file, and
%| sets objects' callback properties to call them through the FEVAL
```



```

%| switchyard above. This comment describes that mechanism.
%|
%| Each callback subfunction declaration has the following form:
%| <SUBFUNCTION_NAME>(H, EVENTDATA, HANDLES, VARARGIN)
%|
%| The subfunction name is composed using the object's Tag and the
%| callback type separated by '_', e.g. 'slider2_Callback',
%| 'figure1_CloseRequestFcn', 'axis1_ButtondownFcn'.
%|
%| H is the callback object's handle (obtained using GCBO).
%|
%| EVENTDATA is empty, but reserved for future use.
%|
%| HANDLES is a structure containing handles of components in GUI using
%| tags as fieldnames, e.g. handles.figure1, handles.slider2. This
%| structure is created at GUI startup using GUIHANDLES and stored in
%| the figure's application data using GUIDATA. A copy of the structure
%| is passed to each callback. You can store additional information in
%| this structure at GUI startup, and you can change the structure
%| during callbacks. Call guidata(h, handles) after changing your
%| copy to replace the stored original so that subsequent callbacks see
%| the updates. Type "help guihandles" and "help guidata" for more
%| information.
%|
%| VARARGIN contains any extra arguments you have passed to the
%| callback. Specify the extra arguments by editing the callback
%| property in the inspector. By default, GUIDE sets the property to:
%| <MFILENAME>(<SUBFUNCTION_NAME>', gcbo, [], guidata(gcbo))
%| Add any extra arguments after the last argument, before the final
%| closing parenthesis.

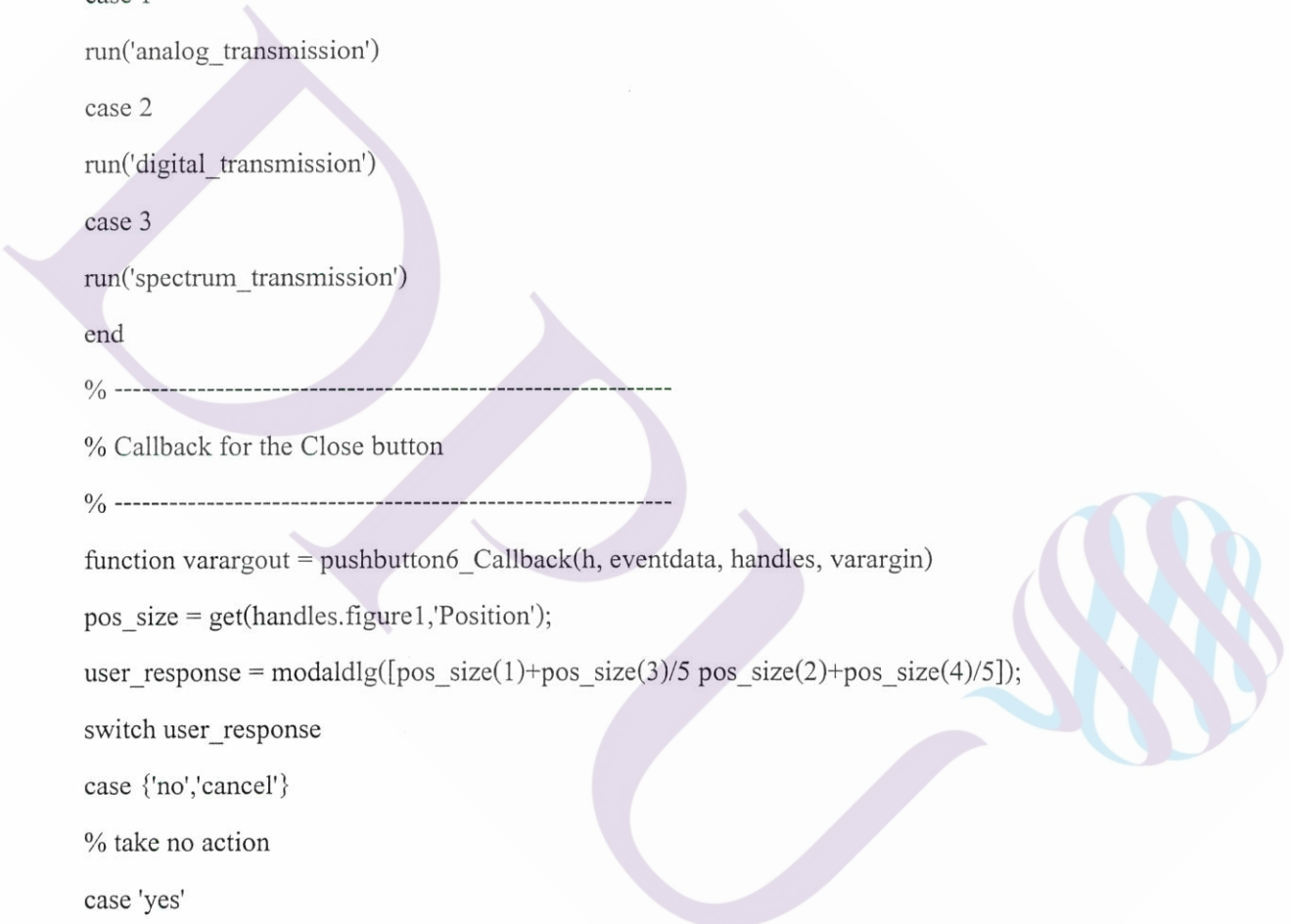
```



```

% -----
% Callback for the switch
% -----
function varargout = listBox9_Callback(h, eventdata, handles, varargin)
val = get(h,'Value');
switch val
case 1
run('analog_transmission')
case 2
run('digital_transmission')
case 3
run('spectrum_transmission')
end
% -----
% Callback for the Close button
% -----
function varargout = pushbutton6_Callback(h, eventdata, handles, varargin)
pos_size = get(handles.figure1,'Position');
user_response = modalDlg([pos_size(1)+pos_size(3)/5 pos_size(2)+pos_size(4)/5]);
switch user_response
case {'no','cancel'}
% take no action
case 'yes'
% Prepare to close GUI application window
%
%
%
delete(handles.figure1)
end
% -----

```



```

% Callback for the GUI figure CloseRequestFcn property
% -----
function varargout = figure1_CloseRequestFcn(h, eventdata, handles, varargin)
pushbutton1_Callback(h,eventdata,handles)
% -----
% Callback for the Help push button
% -----
function varargout = pushbutton7_Callback(h, eventdata, handles, varargin)
% Callback of the uicontrol handles.HelpButton.
err = javachk('mwt', 'The Help browser');
if (~isempty(err))
error('The Help browser is not supported on this platform. Try helpdesk or helpwin instead.');
```

end

```

try
    % Launch the Help Browser
    com.mathworks.ide.help.HelpBrowser.invoke;
catch
    % Failed. Bail
    error('Failed to open Help Browser');
```

end

2. อ้างอิงจากหัวข้อ 3.3.2 คือ CODE ในการสร้างหน้าต่างแสดงผล DIGITAL TRANSMISSION

```


function varargout = DIGITAL_TRANSMISSION(varargin)
% DIGITAL_TRANSMISSION Application M-file for DIGITAL_TRANSMISSION.fig
% FIG = DIGITAL_TRANSMISSION launch DIGITAL_TRANSMISSION GUI.
% DIGITAL_TRANSMISSION('callback_name', ...) invoke the named callback.
% Last Modified by GUIDE v2.0 01-Mar-2004 23:16:41
if nargin == 0 % LAUNCH GUI
fig = openfig(mfilename,'reuse');
```

% Generate a structure of handles to pass to callbacks, and store it.


```

handles = guihandles(fig);
guidata(fig, handles);
if nargout > 0
varargout{1} = fig;
end
elseif ischar(varargin{1}) % INVOKE NAMED SUBFUNCTION OR CALLBACK
try
if (nargout)
[varargout{1:nargout}] = feval(varargin{:}); % FEVAL switchyard
else
feval(varargin{:}); % FEVAL switchyard
end
catch
disp(lasterr);
end
end
%| ABOUT CALLBACKS:
%| GUIDE automatically appends subfunction prototypes to this file, and
%| sets objects' callback properties to call them through the FEVAL
%| switchyard above. This comment describes that mechanism.
%|
%| Each callback subfunction declaration has the following form:
%| <SUBFUNCTION_NAME>(H, EVENTDATA, HANDLES, VARARGIN)
%|
%| The subfunction name is composed using the object's Tag and the
%| callback type separated by '_', e.g. 'slider2_Callback',
%| 'figure1_CloseRequestFcn', 'axis1_ButtontdownFcn'.
%|
%| H is the callback object's handle (obtained using GCBO).
%|

```



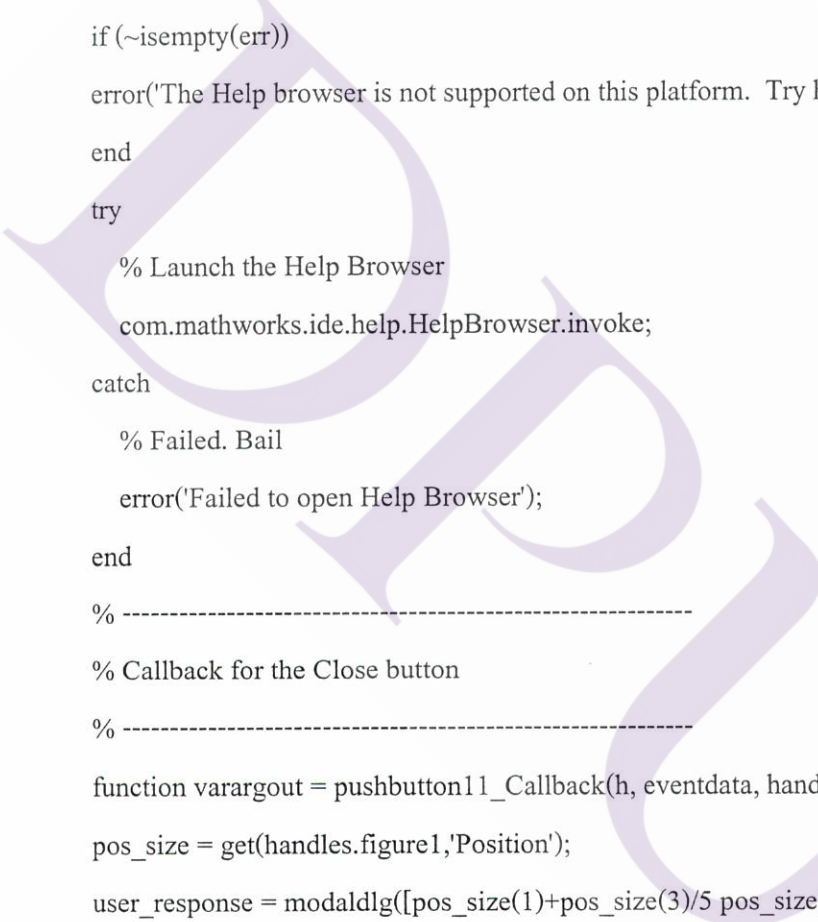

```

%| EVENTDATA is empty, but reserved for future use.
%|
%| HANDLES is a structure containing handles of components in GUI using
%| tags as fieldnames, e.g. handles.figure1, handles.slider2. This
%| structure is created at GUI startup using GUIHANDLES and stored in
%| the figure's application data using GUIDATA. A copy of the structure
%| is passed to each callback. You can store additional information in
%| this structure at GUI startup, and you can change the structure
%| during callbacks. Call guidata(h, handles) after changing your
%| copy to replace the stored original so that subsequent callbacks see
%| the updates. Type "help guihandles" and "help guidata" for more
%| information.
%|
%| VARARGIN contains any extra arguments you have passed to the
%| callback. Specify the extra arguments by editing the callback
%| property in the inspector. By default, GUIDE sets the property to:
%| <MFILENAME>(<SUBFUNCTION_NAME>', gcbo, [], guidata(gcbo))
%| Add any extra arguments after the last argument, before the final
%| closing parenthesis.
% -----
% Callback for the switch
% -----
function varargout = listbox6_Callback(h, eventdata, handles, varargin)
val = get(h,'Value');
switch val
case 1
run('chapter_05')
case 2
run('chapter_06')
case 3

```

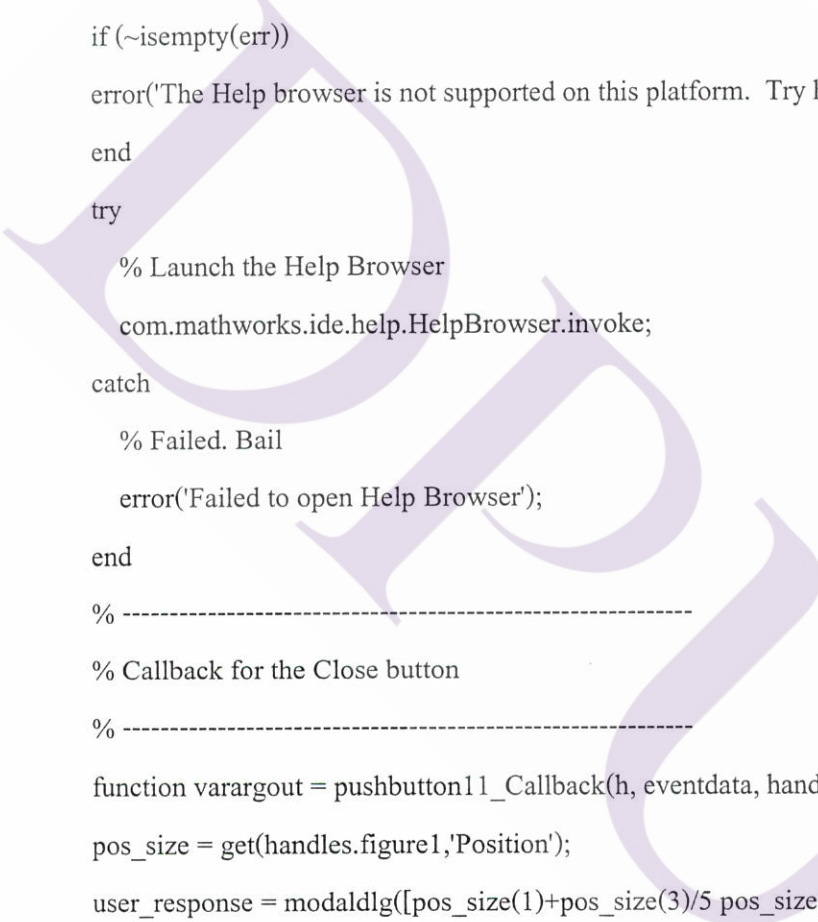
```

run('chapter_07')
end
% -----
% Callback for the Help push button
% -----
function varargout = pushbutton12_Callback(h, eventdata, handles, varargin)
err = javachk('mwt', 'The Help browser');
if (~isempty(err))
error('The Help browser is not supported on this platform. Try helpdesk or helpwin instead.');
```

```

end
try
% Launch the Help Browser
com.mathworks.ide.help.HelpBrowser.invoke;
catch
% Failed. Bail
error('Failed to open Help Browser');
```



```

end
% -----
% Callback for the Close button
% -----
function varargout = pushbutton11_Callback(h, eventdata, handles, varargin)
pos_size = get(handles.figure1, 'Position');
user_response = modaldlg([pos_size(1)+pos_size(3)/5 pos_size(2)+pos_size(4)/5]);
switch user_response
case {'no','cancel'}
% take no action
case 'yes'
% Prepare to close GUI application window
%
%
```



```

%
delete(handles.figure1)

end

% -----
% Callback for the GUI figure CloseRequestFcn property
% -----
function varargout = figure1_CloseRequestFcn(h, eventdata, handles, varargin)
% -----
function varargout = pushbutton15_Callback(h, eventdata, handles, varargin)
run('menu')

```

3. อ้างอิงจากหัวข้อ 3.3.3 คือ CODE โปรแกรมในการสร้างหน้าต่างแสดงผล CHAPTER

```

function varargout = CHAPTER_06(varargin)
% CHAPTER_06 Application M-file for CHAPTER_06.fig
% FIG = CHAPTER_06 launch CHAPTER_06 GUI.
% CHAPTER_06('callback_name', ...) invoke the named callback.
% Last Modified by GUIDE v2.0 02-Mar-2004 02:06:05


if nargin == 0 % LAUNCH GUI
    fig = openfig(mfilename,'reuse');
    % Generate a structure of handles to pass to callbacks, and store it.
    handles = guihandles(fig);
    guidata(fig, handles);
    if nargin > 0
        varargout{1} = fig;
    end
elseif ischar(varargin{1}) % INVOKE NAMED SUBFUNCTION OR CALLBACK
    try
        if (nargout)
            [varargout{1:nargout}] = feval(varargin{:}); % FEVAL switchyard
        else

```

```

        feval(varargin{:}); % FEVAL switchyard
    end
catch
    disp(lasterr);
end
end
%| ABOUT CALLBACKS:
%| GUIDE automatically appends subfunction prototypes to this file, and
%| sets objects' callback properties to call them through the FEVAL
%| switchyard above. This comment describes that mechanism.
%|
%| Each callback subfunction declaration has the following form:
%| <SUBFUNCTION_NAME>(H, EVENTDATA, HANDLES, VARARGIN)
%|
%| The subfunction name is composed using the object's Tag and the
%| callback type separated by '_', e.g. 'slider2_Callback',
%| 'figure1_CloseRequestFcn', 'axis1_ButtondownFcn'.
%|
%| H is the callback object's handle (obtained using GCBO).
%|
%| EVENTDATA is empty, but reserved for future use.
%|
%| HANDLES is a structure containing handles of components in GUI using
%| tags as fieldnames, e.g. handles.figure1, handles.slider2. This
%| structure is created at GUI startup using GUIHANDLES and stored in
%| the figure's application data using GUIDATA. A copy of the structure
%| is passed to each callback. You can store additional information in
%| this structure at GUI startup, and you can change the structure
%| during callbacks. Call guidata(h, handles) after changing your
%| copy to replace the stored original so that subsequent callbacks see

```

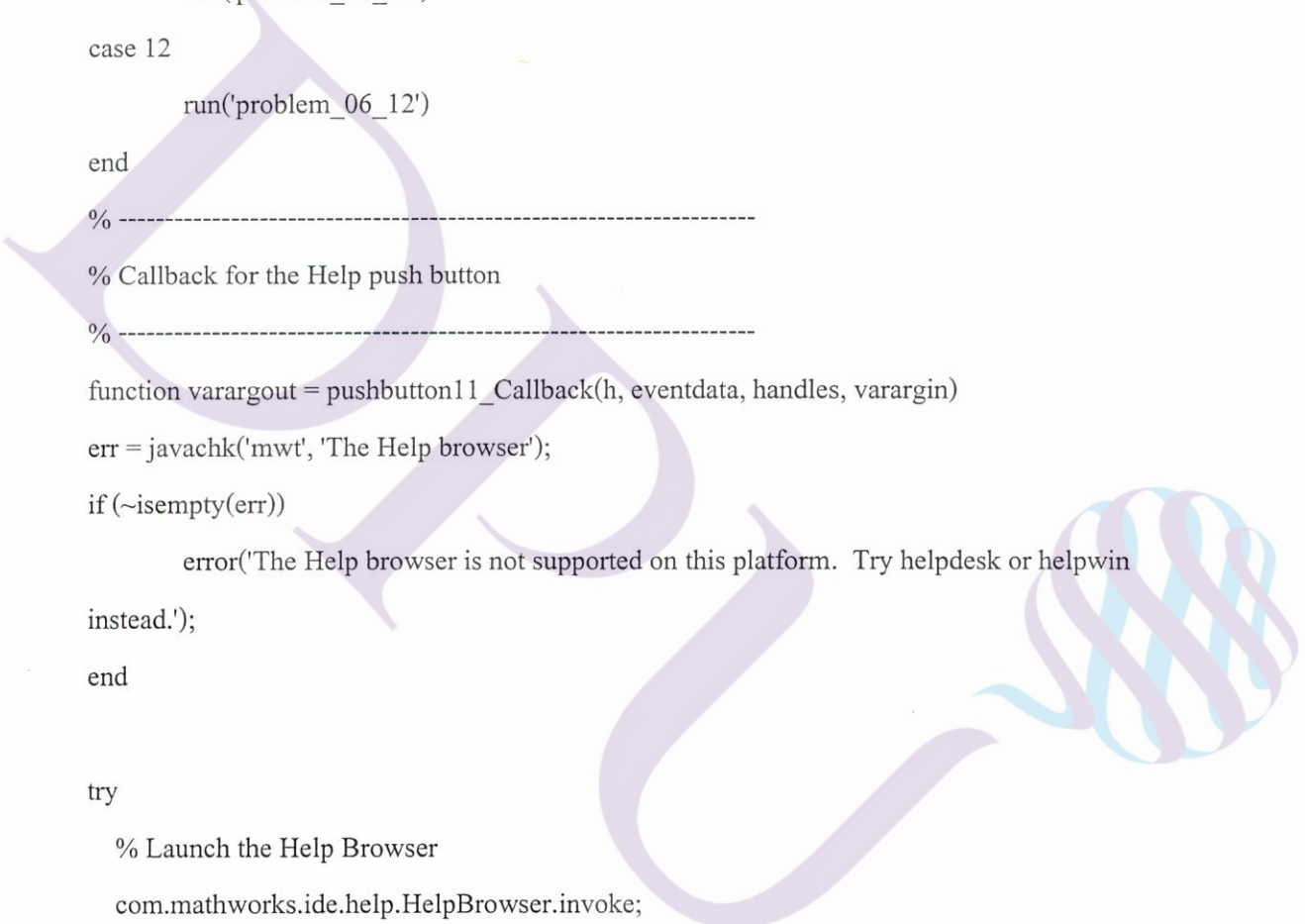


```
%| the updates. Type "help guihandles" and "help guidata" for more
%| information.
%|
%| VARARGIN contains any extra arguments you have passed to the
%| callback. Specify the extra arguments by editing the callback
%| property in the inspector. By default, GUIDE sets the property to:
%| <MFILENAME>(<SUBFUNCTION_NAME>', gcbo, [], guidata(gcbo))
%| Add any extra arguments after the last argument, before the final
%| closing parenthesis.
% -----
% Callback for the switch
% -----
function varargout = listbox4_Callback(h, eventdata, handles, varargin)
val = get(h,'Value');
switch val
case 1
    run('problem_06_01')
case 2
    run('problem_06_02')
case 3
    run('problem_06_03')
case 4
    run('problem_06_04')
case 5
    run('problem_06_05')
case 6
    run('problem_06_06')
case 7
    run('problem_06_07')
case 8
```

```

        run('problem_06_08')
case 9
        run('problem_06_09')
case 10
        run('problem_06_10')
case 11
        run('problem_06_11')
case 12
        run('problem_06_12')
end
% -----
% Callback for the Help push button
% -----

function varargout = pushbutton11_Callback(h, eventdata, handles, varargin)
err = javachk('mwt', 'The Help browser');
if (~isempty(err))
    error('The Help browser is not supported on this platform. Try helpdesk or helpwin
instead.');
```



```

end

try
    % Launch the Help Browser
    com.mathworks.ide.help.HelpBrowser.invoke;
catch
    % Failed. Bail
    error('Failed to open Help Browser');
end
% -----
% Callback for the Close button
% -----
```



```

function varargout = pushbutton10_Callback(h, eventdata, handles, varargin)
pos_size = get(handles.figure1,'Position');
user_response = modaldlg([pos_size(1)+pos_size(3)/5 pos_size(2)+pos_size(4)/5]);
switch user_response
case {'no','cancel'}
    % take no action
case 'yes'
    % Prepare to close GUI application window
    % .
    % .
    % .
    delete(handles.figure1)
end
% -----
% Callback for the GUI figure CloseRequestFcn property
% -----
function varargout = figure1_CloseRequestFcn(h, eventdata, handles, varargin)
% -----
% Callback for the run('menu')
% -----
function varargout = pushbutton13_Callback(h, eventdata, handles, varargin)
run('menu')
% -----
% Callback for the run('digital_transmission')
% -----
function varargout = pushbutton12_Callback(h, eventdata, handles, varargin)
run('digital_transmission')

```

4. อ้างอิงจากหัวข้อ 3.3.4 คือ CODE ในการสร้างหน้าต่างแสดงผล PROBLEM

```
function varargout = PROBLEM_06_03(varargin)
```

```

% PROBLEM_06_03 Application M-file for PROBLEM_06_03.fig
% FIG = PROBLEM_06_03 launch PROBLEM_06_03 GUI.
% PROBLEM_06_03('callback_name', ...) invoke the named callback.
% Last Modified by GUIDE v2.5 19-Aug-2005 20:47:20

    if nargin == 0 % LAUNCH GUI
        fig = openfig(mfilename,'reuse');
        % Generate a structure of handles to pass to callbacks, and store it.
        handles = guihandles(fig);
        guidata(fig, handles);
        if nargin > 0
            varargout{1} = fig;
        end
    elseif ischar(varargin{1}) % INVOKE NAMED SUBFUNCTION OR CALLBACK
        try
            if (nargout)
                [varargout{1:nargout}] = feval(varargin{:}); % FEVAL switchyard
            else
                feval(varargin{:}); % FEVAL switchyard
            end
        catch
            disp(lasterr);
        end
    end

end

%| ABOUT CALLBACKS:
%| GUIDE automatically appends subfunction prototypes to this file, and
%| sets objects' callback properties to call them through the FEVAL
%| switchyard above. This comment describes that mechanism.
%|
%| Each callback subfunction declaration has the following form:
%| <SUBFUNCTION_NAME>(H, EVENTDATA, HANDLES, VARARGIN)

```

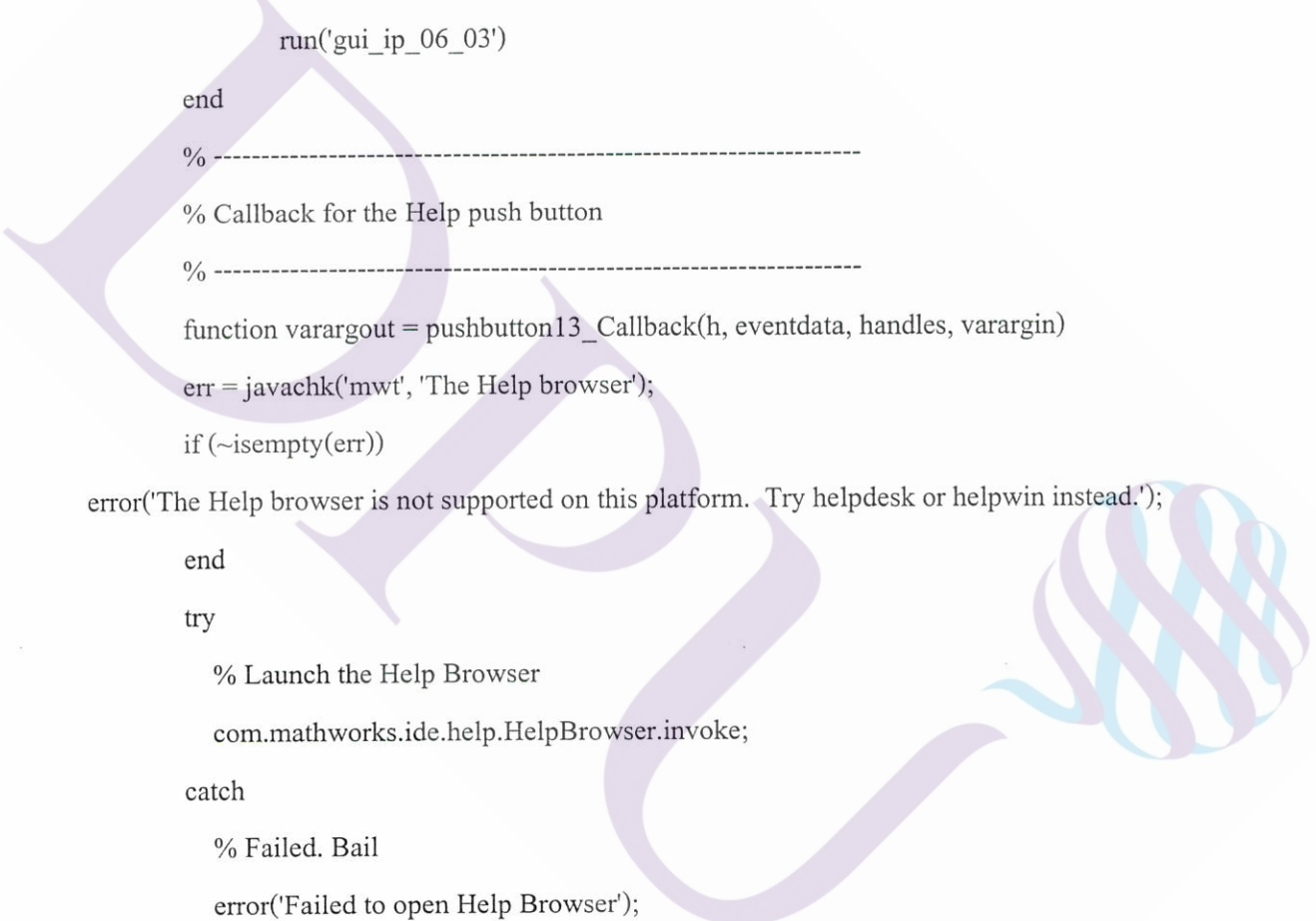
```

%|
%| The subfunction name is composed using the object's Tag and the
%| callback type separated by '_', e.g. 'slider2_Callback',
%| 'figure1_CloseRequestFcn', 'axis1_ButtondownFcn'.
%|
%| H is the callback object's handle (obtained using GCBO).
%|
%| EVENTDATA is empty, but reserved for future use.
%|
%| HANDLES is a structure containing handles of components in GUI using
%| tags as fieldnames, e.g. handles.figure1, handles.slider2. This
%| structure is created at GUI startup using GUIHANDLES and stored in
%| the figure's application data using GUIDATA. A copy of the structure
%| is passed to each callback. You can store additional information in
%| this structure at GUI startup, and you can change the structure
%| during callbacks. Call guidata(h, handles) after changing your
%| copy to replace the stored original so that subsequent callbacks see
%| the updates. Type "help guihandles" and "help guidata" for more
%| information.
%|
%| VARARGIN contains any extra arguments you have passed to the
%| callback. Specify the extra arguments by editing the callback
%| property in the inspector. By default, GUIDE sets the property to:
%| <MFILENAME>('<SUBFUNCTION_NAME>', gcbo, [], guidata(gcbo))
%| Add any extra arguments after the last argument, before the final
%| closing parenthesis.
% -----
% Callback for the switch
% -----
function varargout = listbox4_Callback(h, eventdata, handles, varargin)

```

```

val = get(h,'Value');
switch val
case 1
    open('D_02_03.html')
case 2
    open('gui_ip_06_03.m')
case 3
    run('gui_ip_06_03')
end
% -----
% Callback for the Help push button
% -----
function varargout = pushbutton13_Callback(h, eventdata, handles, varargin)
err = javachk('mwt', 'The Help browser');
if (~isempty(err))
error('The Help browser is not supported on this platform. Try helpdesk or helpwin instead.');
```



```

end
try
    % Launch the Help Browser
    com.mathworks.ide.help.HelpBrowser.invoke;
catch
    % Failed. Bail
    error('Failed to open Help Browser');
```

```

end
% -----
% Callback for the Close button
% -----
function varargout = pushbutton12_Callback(h, eventdata, handles, varargin)
pos_size = get(handles.figure1,'Position');
user_response = modaldlg([pos_size(1)+pos_size(3)/5 pos_size(2)+pos_size(4)/5]);

```



```

switch user_response
case {'no','cancel'}
    % take no action
case 'yes'
    % Prepare to close GUI application window
    %      .
    %      .
    %      .
    delete(handles.figure1)
end
% -----
% Callback for the GUI figure CloseRequestFcn property
% -----
function varargout = figure1_CloseRequestFcn(h, eventdata, handles, varargin)
% -----
function varargout = pushbutton15_Callback(h, eventdata, handles, varargin)
run('menu')
% -----
function varargout = pushbutton14_Callback(h, eventdata, handles, varargin)
run('digital_transmission')
% -----
function varargout = pushbutton16_Callback(h, eventdata, handles, varargin)
run('chapter_06')
% --- Executes on selection change in listbox5.
function listbox6_Callback(hObject, eventdata, handles)
% hObject    handle to listbox5 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = get(hObject,'String') returns listbox5 contents as cell array

```

```

% contents {get(hObject,'Value')} returns selected item from listBox5
open('gui_ip_06_03.m')
% --- Executes during object creation, after setting all properties.
function listBox5_CreateFcn(hObject, eventdata, handles)
% hObject handle to listBox5 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called
% Hint: listBox controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc
    set(hObject,'BackgroundColor','white');
else
    set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end
% --- Executes during object creation, after setting all properties.
function listBox6_CreateFcn(hObject, eventdata, handles)
% hObject handle to listBox6 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called
% Hint: listBox controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc
    set(hObject,'BackgroundColor','white');
else
    set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end

```

5. อ้างอิงจากหัวข้อ 3.3.5 คือ CODE ในการสร้างหน้าต่างแสดงผล M-file ใน matlab editor ก่อนสร้างเป็น GUI

% MATLAB script for Illustrative Problem 3, Chapter 6.

```

echo on
f_cutoff=2000;           % the desired cut-off frequency
f_stopband=2500;       % the actual stopband frequency
fs=10000;              % the sampling frequency
f1=2*f_cutoff/fs;      % the normalized passband frequency
f2=2*f_stopband/fs;    % the normalized stopband frequency
N=41;                  % this number is found by experiment
F=[0 f1 f2 1];
M=[1 1 0 0];          % describes the low-pass filter
B=remez(N-1,F,M);      % returns the FIR tap coefficients
% plotting command follows
figure(1);
[H,W]=freqz(B);
H_in_dB=20*log10(abs(H));
plot(W/(2*pi),H_in_dB);
figure(2);
plot(W/(2*pi),(180/pi)*unwrap(angle(H)));
%plot of the impulse response follows
figure(3);
stem([0:N-1],B);

```

6. อ้างอิงจากหัวข้อ 3.3.6 คือ CODE ในการสร้างหน้าต่างแสดงผล RUN PROGRAMME

```

function varargout = gui_ip_06_03(varargin)

% GUI_IP_06_03 Application M-file for gui_ip_06_03.fig
% FIG = GUI_IP_06_03 launch gui_ip_06_03 GUI.
% GUI_IP_06_03('callback_name', ...) invoke the named callback.
% Last Modified by GUIDE v2.0 09-Sep-2003 23:38:16

if nargin == 0 % LAUNCH GUI

    fig = openfig(mfilename,'reuse');

    % Generate a structure of handles to pass to callbacks, and store it.

```

```

handles = guihandles(fig);
guidata(fig, handles);
if nargout > 0
    varargout{1} = fig;
end
elseif ischar(varargin{1}) % INVOKE NAMED SUBFUNCTION OR CALLBACK
    try
        if (nargout)
            [varargout{1:nargout}] = feval(varargin{:}); % FEVAL switchyard
        else
            feval(varargin{:}); % FEVAL switchyard
        end
    catch
        disp(lasterr);
    end
end
%| ABOUT CALLBACKS:
%| GUIDE automatically appends subfunction prototypes to this file, and
%| sets objects' callback properties to call them through the FEVAL
%| switchyard above. This comment describes that mechanism.
%|
%| Each callback subfunction declaration has the following form:
%| <SUBFUNCTION_NAME>(H, EVENTDATA, HANDLES, VARARGIN)
%|
%| The subfunction name is composed using the object's Tag and the
%| callback type separated by '_', e.g. 'slider2_Callback',
%| 'figure1_CloseRequestFcn', 'axis1_ButtondownFcn'.
%|
%| H is the callback object's handle (obtained using GCBO).
%|

```



```

%| EVENTDATA is empty, but reserved for future use.
%|
%| HANDLES is a structure containing handles of components in GUI using
%| tags as fieldnames, e.g. handles.figure1, handles.slider2. This
%| structure is created at GUI startup using GUIHANDLES and stored in
%| the figure's application data using GUIDATA. A copy of the structure
%| is passed to each callback. You can store additional information in
%| this structure at GUI startup, and you can change the structure
%| during callbacks. Call guidata(h, handles) after changing your
%| copy to replace the stored original so that subsequent callbacks see
%| the updates. Type "help guihandles" and "help guidata" for more
%| information.
%|
%| VARARGIN contains any extra arguments you have passed to the
%| callback. Specify the extra arguments by editing the callback
%| property in the inspector. By default, GUIDE sets the property to:
%| <MFILENAME>(<SUBFUNCTION_NAME>', gcbo, [], guidata(gcbo))
%| Add any extra arguments after the last argument, before the final
%| closing parenthesis.
% -----
function varargout = pushbutton1_Callback(h, eventdata, handles, varargin)
f_cutoff = str2double(get(handles.f_cutoff_input,'String')); % the desired cut-off
frequency
f_stopband = str2double(get(handles.f_stopband_input,'String')); % the actual stopband
frequency
fs = str2double(get(handles.fs_input,'String')); % the sampling frequency
N = str2double(get(handles.N_input,'String')); % this number is found by experiment
% MATLAB script for Illustrative Problem 3, Chapter 6.
echo on
f1=2*f_cutoff/fs; % the normalized passband frequency

```

```

f2=2*f_stopband/fs;           % the normalized stopband frequency
F=[0 f1 f2 1];
M=[1 1 0 0];                 % describes the low-pass filter
B=remez(N-1,F,M);           % returns the FIR tap coefficients

% Create time plot
axes(handles.f1) % Select the proper axes
[H,W]=freqz(B);
H_in_dB=20*log10(abs(H));
plot(W/(2*pi),H_in_dB-80);
set(handles.f1,'XMinorTick','on')
grid on

% Create time plot
axes(handles.f2) % Select the proper axes
plot(W/(2*pi),(180/pi)*unwrap(angle(H)));
set(handles.f2,'XMinorTick','on')
grid on

% Create time plot
axes(handles.n1) % Select the proper axes
stem([0:N-1],B);
set(handles.n1,'XMinorTick','on')
grid on

% Create time plot
axes(handles.n2) % Select the proper axes
x = 0:0.0001:20 ;
plot(x,square(x))
set(gca,'XTick',-pi:pi/2:pi)
set(gca,'XTickLabel',{'-pi','-pi/2','0','pi/2','pi'})
axis([0 20 -5 5])
set(handles.n2,'XMinorTick','on')
grid on

```

```

% Create time plot
axes(handles.axes3) % Select the proper axes
x = 0:0.0001:20 ;
plot(x,square(x))
set(gca,'XTick',-pi:pi/2:pi)
set(gca,'XTickLabel',{'-pi','-pi/2','0','pi/2','pi'})
axis([0 20 -5 5])
set(handles.axes3,'XMinorTick','on')
grid on

```

7. อ้างอิงจากหัวข้อ 3.3.7 คือ CODE ในการสร้างหน้าต่างแสดงผล CLOSE

```

function answer = modaldlg(varargin)
% modaldlg Application M-file for modaldlg.fig
% answer = modaldlg return the answer.
% modaldlg('callback_name') invoke the named callback.
% modaldlg([left bottom]) locates the dialog.
% Last Modified by GUIDE v2.0 20-Jul-2000 13:59:31
error(nargchk(0,4,nargin)) % function takes only 0 or 4 argument
if nargin == 0 | isnumeric(varargin{1}) % LAUNCH GUI
    fig = openfig(mfilename,'reuse');
    % Use system color scheme for figure:
    set(fig,'Color',get(0,'defaultUicontrolBackgroundColor'));
    % Generate a structure of handles to pass to callbacks, and store it.
    handles = guihandles(fig);
    guidata(fig, handles);
    % Position figure
    if nargin == 1
        pos_size = get(fig,'Position');
        pos = varargin{1};
        if length(pos) ~= 2

```

```

error('Input argument must be a 2-element vector')
end
new_pos = [pos(1) pos(2) pos_size(3) pos_size(4)];
set(fig,'Position',new_pos,'Visible','on')
figure(fig)
    end
    % Wait for callbacks to run and window to be dismissed:
    uiwait(fig);
    % UIWAIT might have returned because the window was deleted using
    % the close box - in that case, return 'cancel' as the answer, and
    % don't bother deleting the window!
    if ~ishandle(fig)
    answer = 'cancel';
    else
    % otherwise, we got here because the user pushed one of the two buttons.
    % retrieve the latest copy of the 'handles' struct, and return the answer.
    % Also, we need to delete the window.
    handles = guidata(fig);
        answer = handles.answer;
    delete(fig);
    end
elseif ischar(varargin{1}) % INVOKE NAMED SUBFUNCTION OR CALLBACK
try
if (nargout)
[varargout{1:nargout}] = feval(varargin{:}); % FEVAL switchyard
else
feval(varargin{:}); % FEVAL switchyard
end
catch
disp(lasterr);

```



```
end
end
% -----
% No button callback stores 'no' in the handles struct, and
% stores the modified handles struct
% (so the main function can see the change).
% -----
function varargout = noButton_Callback(h, eventdata, handles, varargin)
handles.answer = 'no';
guidata(h, handles);
uiresume(handles.figure1);
% -----
% Yes button callback uses uiresume to
% continue the blocked code in the main function.
% -----
function varargout = yesButton_Callback(h, eventdata, handles, varargin)
handles.answer = 'yes';
guidata(h,handles);
uiresume(handles.figure1);
```

ไม่มีเอกสารจากต้นฉบับ
หน้าประวัติผู้แต่ง