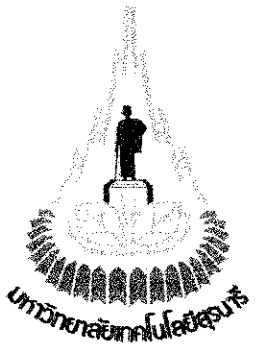
การวิเคราะห์และออกแบบระบบ MIMO ด้วย GUI
Analysis and Design of MIMO system via GUI

โดย

นายสาธิต	ลิ้มสำลี	รหัสนักศึกษา	55364230
นายสรวิทย์	บุญยาลาง	รหัสนักศึกษา	55311522
นายสรายุทธ	กิ่งแก้วเพียะ	รหัสนักศึกษา	55312401

รายงานนี้เป็นส่วนหนึ่งของการศึกษา ตามหลักสูตรวิศวกรรมศาสตรบัณฑิต
ภาคการศึกษาที่ 2 ปีการศึกษา 2559
หลักสูตรวิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมไฟฟ้า หลักสูตรปรับปรุง พ.ศ.2545
คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี

CONTRIBUTION



การวิเคราะห์และออกแบบระบบ MIMO ด้วย GUI

Analysis and Design of MIMO system via GUI

โดย

นางสาวศจี	สิงห์สง่า	B4704270
นางสาววาทินี	บุญมานาง	B4711322
นางสาวสายฝน	เรียงสันเทียะ	B4712466

รายงานนี้เป็นส่วนหนึ่งของรายวิชา 427499 โครงงานวิศวกรรมโทรคมนาคม

ประจำภาคการศึกษาที่ 3 ปีการศึกษา 2550

หลักสูตรวิศวกรรมศาสตรบัณฑิต สาขาวิศวกรรมโทรคมนาคม หลักสูตรปรับปรุง พ.ศ. 2545

สำนักวิชาวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีสุรนารี

ศูนย์บรรณสารและสื่อการศึกษา
มหาวิทยาลัยเทคโนโลยีสุรนารี

การวิเคราะห์และออกแบบระบบ MIMO ด้วย GUI

คณะกรรมการสอบโครงงาน

(อาจารย์ ดร. พีระพงษ์ อุฑารสกุล)
อาจารย์ที่ปรึกษาโครงงาน

(ผู้ช่วยศาสตราจารย์ ดร. ชุติมา พรหมมาก)
กรรมการ

(ผู้ช่วยศาสตราจารย์ ร. อ.ดร. ประโยชน์ คำสวัสดิ์)
กรรมการ

มหาวิทยาลัยเทคโนโลยีสุรนารี อนุมัติให้นับรายงานโครงงานฉบับนี้ เป็นส่วนหนึ่งของการศึกษาระดับปริญญาตรี สาขาวิชาวิศวกรรมโทรคมนาคม วิชา 427 499 โครงงานวิศวกรรมโทรคมนาคม ประจำปีการศึกษา 2550

โครงงาน	การวิเคราะห์และออกแบบระบบ MIMO ด้วย GUI
ผู้ดำเนินงาน	นางสาวศจี สิงห์สง่า
	นางสาววาทินี บุญมานาง
	นางสาวสายฝน เรียงสันเทียะ
อาจารย์ที่ปรึกษา	อาจารย์ ดร.พีระพงษ์ อุฑารสกุล
สาขาวิชา	วิศวกรรมโทรคมนาคม
ภาคการศึกษา	3/2550

บทคัดย่อ

ปัจจุบันการสื่อสารไร้สายมีการใช้กันอย่างแพร่หลาย ซึ่งเทคโนโลยีการสื่อสารไร้สายนี้ได้มีการพัฒนาขึ้นในหลายรูปแบบตามจุดประสงค์การใช้งานและระบบ MIMO ก็เป็นเทคโนโลยีที่ได้รับความสนใจมาก ลักษณะของระบบ MIMO จะเป็นการเพิ่มจำนวนสายอากาศเข้าไปให้กับอุปกรณ์ ไร้สายทั้งภาคส่งและภาครับ เพื่อให้รับสัญญาณได้ดียิ่งขึ้น แม้ระยะที่ไกลออกไปหรือสัญญาณรบกวนมากๆก็สามารถรับส่งสัญญาณได้อย่างมีประสิทธิภาพ ซึ่งโครงงานนี้ถูกจัดทำขึ้นมาเพื่อวิเคราะห์อัตราความผิดพลาดของการส่งข้อมูลและออกแบบค่าพารามิเตอร์ต่างๆ โดยทำการเขียนโปรแกรมและประมวลผลออกมาในรูปแบบกราฟฟิกที่ง่ายและสะดวกสำหรับผู้ใช้งาน เพื่อวิเคราะห์และออกแบบระบบ MIMO ได้ด้วยด้วยตนเอง

กิตติกรรมประกาศ

การทำโครงงานชิ้นนี้สำเร็จลงได้ เพราะคณะผู้จัดทำได้รับความช่วยเหลือจากบุคคลหลายท่าน เริ่มจากอาจารย์ที่ปรึกษาโครงงาน อาจารย์ ดร.พีระพงษ์ อุฑารสกุล อาจารย์สาขาโทรคมนาคมที่คอยดูแลและให้คำปรึกษาในด้านต่างๆแก่คณะผู้จัดทำอย่างใกล้ชิดมาโดยตลอด ขอขอบพระคุณท่านอาจารย์และบุคลากรของสาขาวิศวกรรมโทรคมนาคมทุกท่านที่คอยช่วยเหลือคณะผู้จัดทำและขอขอบคุณเพื่อน พี่ และน้องสาขาวิศวกรรมโทรคมนาคมที่คอยช่วยเหลือและคอยเป็นกำลังใจให้เสมอมา

สุดท้ายนี้คณะผู้จัดทำใคร่ขอกราบขอบพระคุณบิดา มารดา ที่ให้โอกาสทางด้านการศึกษา ให้การเอาใจใส่เลี้ยงดู คอยเป็นกำลังใจและให้การสนับสนุนในทุกๆด้านมาโดยตลอด จึงเห็นสมควรที่จะมอบคุณความดีและเกียติคุณแก่ทุกๆท่านที่ได้กล่าวถึงมา ณ ที่นี้ด้วย

นางสาวศจี สิงห์สง่า
นางสาววาทินี บุญมานาง
นางสาวสายฝน เรียงสันเทียะ

สารบัญ

	หน้า
บทคัดย่อ	**ก**
กิตติกรรมประกาศ	**ข**
สารบัญ	**ค**
สารบัญรูป	**ฉ**
บทที่ 1 บทนำ	1
1.1 ความเป็นมา	1
1.2 วัตถุประสงค์ของโครงงาน	1
1.3 ขอบเขตการทำงาน	2
1.4 ขั้นตอนการดำเนินงาน	2
บทที่ 2 ทฤษฎีและหลักการ	3
2.1 ทฤษฎีและหลักการของ MIMO	3
2.2 ช่องสัญญาณของระบบ MIMO	5
2.2.1 ช่องสัญญาณที่มีการลดทอนแบบเรเลย์	6
2.2.2 ช่องสัญญาณที่มีการลดทอนแบบริเชียน	7
2.3 เทคนิคการส่งสัญญาณ	9
2.4 พารามิเตอร์ที่สำคัญ	10
2.4.1 การมอดูเลชั่นแบบ Phase Shift Keying (PSK)	10
2.4.2 อัตราความผิดพลาดของการส่งข้อมูล	11
2.4.3 Signal-to-noise ratio	12
2.5 หลักการของ GUI	14
2.5.1 การสร้าง GUI ด้วย GUIDE	15
2.5.2 ส่วนประกอบสำคัญของ Application M-file ที่สร้างโดย GUIDE	16
2.5.3 การสร้าง Application M-file ของ GUIDE	20
2.5.4 การกำหนดค่าคุณสมบัติของส่วนประกอบต่างๆ	28
2.5.5 User Interface Controls	29
2.5.6 Understanding the Application M-File	30

สารบัญ(ต่อ)

	หน้า
บทที่ 3 การทำงานและโครงสร้างของโปรแกรม	35
3.1 โปรแกรมการวิเคราะห์อัตราความผิดพลาดในการส่งข้อมูล(BER)	35
3.2 โปรแกรมการออกแบบพารามิเตอร์	38
บทที่ 4 การทดสอบและการวิเคราะห์	43
4.1 วิเคราะห์ผลระหว่าง BER และ SNR	43
4.1.1 เมื่อเพิ่มจำนวน M-ary	43
4.1.2 เมื่อเพิ่มจำนวนสายอากาศภาคส่ง	44
4.1.3 เมื่อเพิ่มจำนวนสายอากาศภาครับ	45
4.1.4 เมื่อเพิ่มค่า K-Factor	47
บทที่ 5 บทสรุป ปัญหาและข้อเสนอแนะ	49
5.1 บทสรุป	49
5.2 ปัญหาที่พบ	49
5.3 ข้อเสนอแนะ	51
5.4 แนวทางการพัฒนาต่อไป	51
บรรณานุกรม	52
ภาคผนวก ก	53
โปรแกรมการวิเคราะห์อัตราความผิดพลาดในการส่งข้อมูล(BER)	53

สารบัญ(ต่อ)

หน้า

ภาคผนวก ข 55
โปรแกรมการออกแบบพารามิเตอร์ 55
โปรแกรมออกแบบจำนวนสายอากาศภาคส่ง 55
โปรแกรมออกแบบจำนวนสายอากาศภาครับ 58
โปรแกรมออกแบบจำนวน M-ary 61
โปรแกรมออกแบบ Signal-to-Noise Ratio (SNR) 63

สารบัญรูป

	หน้า
รูปที่ 2.1 บล็อกไดอะแกรมของระบบ MIMO	3
รูปที่ 2.2 ตัวอย่างผลิตภัณฑ์ซีรีส์ **Pre-N** ที่ผนวกรวมความสามารถของเทคโนโลยี **MIMO**	4
รูปที่ 2.3 ช่องสัญญาณในระบบ MIMO	6
รูปที่ 2.4 สภาพแวดล้อมที่เหมาะต่อช่องสัญญาณที่มีการลดทอนแบบเรเลย์	6
รูปที่ 2.5 การรับ-ส่งสัญญาณแบบ Line Of Sight (LOS)	7
รูปที่ 2.6 เทคนิคการส่งสัญญาณแบบ SM	9
รูปที่ 2.7 Constellation Diagram ของการผสมสัญญาณแบบ Phase Shift Keying แบบ Binary PSK ซึ่งใช้เฟส 2 เฟส แทนข้อมูล 0 และ 1	11
รูปที่ 2.8 กราฟแสดงค่าBERที่มีการเข้ารหัสแบบ BPSK, QPSK, 8-PSK and 16-PSK	12
รูปที่ 2.9 กราฟแสดงความสัมพันธ์ระหว่าง BER กับ SNR โดยใช้การเข้ารหัสสัญญาณแบบ BSPK/QPSK ,8PSK และ 16PSK	13
รูปที่ 2.10 Layout Editor	17
รูปที่ 2.11 การควรกำหนดตัวเลือกต่าง ๆ ภายใต้เมนู Tool	17
รูปที่ 2.12 หน้าต่าง GUIDE Application Options	18
รูปที่ 2.13 เมนู Context ของ figure	25
รูปที่ 2.14 หน้าต่างของฟังก์ชัน Alignment Tool	26
รูปที่ 2.15 ฟังก์ชัน Grid and Rulers ภายใต้เมนูLayout	27
รูปที่ 2.16 หน้าต่างแสดงการสร้าง Grid size	27
รูปที่ 2.17 Property Inspector ใช้แสดงคุณสมบัติของวัตถุ	29
รูปที่ 3.1 หน้าต่างแสดงให้ผู้ใช้เลือกการใช้งานของโปรแกรม	35
รูปที่ 3.2 หน้าต่างแสดงการรับค่าพารามิเตอร์จากผู้ใช้งาน	36
รูปที่ 3.3 หน้าต่างแสดงความสัมพันธ์ระหว่าง BER กับ SNR	36
รูปที่ 3.4 หน้าต่างแสดงการรับค่าพารามิเตอร์หลังจากที่ผู้ใช้งานกดปุ่ม Hold	37
รูปที่ 3.5 หน้าต่างแสดงการเปรียบเทียบระหว่างข้อมูลใหม่กับข้อมูลเดิม	38
รูปที่ 3.6 หน้าต่างแสดงให้ผู้ใช้เลือกการออกแบบพารามิเตอร์	39
รูปที่ 3.7 หน้าต่างแสดงการรับค่าพารามิเตอร์จากผู้ใช้งาน	39

รูปที่ 3.8 หน้าต่างแสดงผลการออกแบบจำนวนสายอากาศภาคส่ง 40
รูปที่ 3.9 หน้าต่างแสดงผลการออกแบบจำนวนสายอากาศภาครับ 41
รูปที่ 3.10 หน้าต่างแสดงผลการออกแบบจำนวนM-ary 41
รูปที่ 3.11 หน้าต่างแสดงผลการออกแบบ SNR 42
รูปที่ 4.1 กราฟความสัมพันธ์ระหว่าง BER และSNR ที่ 2PSK , 4PSK , 8PSK , 16PSK , 32PSK และ64PSK (ก) ช่องสัญญาณแบบ Rayleigh (ข) ช่องสัญญาณแบบ Rician 43
รูปที่4.2 กราฟความสัมพันธ์ระหว่าง BER และSNRเมื่อทำการเพิ่มจำนวนสายอากาศทางภาคส่ง โดย (ก),(ค)และ(จ) เป็นช่องสัญญาณแบบ Rayleigh และ(ข),(ง)และ(ฉ) เป็นช่องสัญญาณ แบบRician 45
รูปที่4.3 กราฟความสัมพันธ์ระหว่าง BER และSNRเมื่อทำการเพิ่มจำนวนสายอากาศทางภาครับโดย (ก),(ค)และ(จ) เป็นช่องสัญญาณแบบ Rayleighและ (ข),(ง)และ(ฉ) เป็นช่องสัญญาณแบบRician 46
รูปที่4.4 กราฟความสัมพันธ์ระหว่าง BER และSNR ที่ Tx=2,Rx=3 และK=0,1,2,4,5,10 ตามลำดับ 47
รูปที่4.5 กราฟความสัมพันธ์ระหว่าง BER และSNR ที่ Tx=4,Rx=6 และ K=1,2,3,4,5,10,20,50 ตามลำดับ 47
รูปที่ 5.1 แสดงหน้าต่างโปรแกรมในส่วนของ POP UP MENU 51

บทที่ 1
บทนำ

1.1 ความเป็นมา

MIMO ย่อจาก Multiple Input/Multiple Output เป็นหนึ่งในเทคโนโลยี Antenna Array ซึ่งเป็นระบบที่สามารถรับสัญญาณได้ดีขึ้นแม้ในระยะที่ไกลออกไปหรือแม้แต่ในที่ๆอุดมไปด้วยการรบกวนของสัญญาณ นอกจากนี้ยังสามารถดักจับสัญญาณได้อย่างมีประสิทธิภาพ นั่นคือเมื่อระยะที่ห่างจาก Access Point ที่เท่ากัน จะรับสัญญาณได้ดีกว่าเมื่อเทียบกับเทคโนโลยีแบบเดิม โดยความเร็วก็ไม่ตกลงในขณะที่เสถียรภาพก็ยังดีอยู่ ซึ่งเป็นการทลายกำแพงของกฎเกณฑ์เดิม ๆ โดยสิ้นเชิง โดยโปรแกรมที่สร้างขึ้นเป็นโปรแกรมเพื่อช่วยในการออกแบบระบบด้วย GUI ซึ่งเป็นโปรแกรมที่มีการประมวลผลได้เร็วและไม่ซับซ้อน ทำให้ง่ายต่อการใช้งานของผู้ใช้

การใช้สายอากาศหลายตัว ณ ภาครับและภาคส่งเป็นที่รู้จักกันในนามของการสื่อสารไร้สายแบบ MIMO (Multiple-Input-Multiple-Output) ซึ่งถือว่าเป็นเทคโนโลยีที่มีประสิทธิผลด้านการลงทุนในการที่จะทำให้เกิดข่ายเชื่อมโยงไร้สายด้วยความเร็วขนาด Gbps ได้จริง

เนื่องจากคุณประโยชน์ดังกล่าวจึงทำให้เกิดโครงงานนี้ขึ้นมาโดยศึกษาผลกระทบต่างๆที่มีผลกระทบต่อระบบ MIMO โดยศึกษาความสัมพันธ์ระหว่าง SNR (signal to noise ratio) และ BER (bit error rate) นั่นคือถ้า SNR มีค่ามากจะทำให้ระบบมีประสิทธิภาพ แต่ถ้ามีค่าน้อย จะส่งผลให้จำนวนของการเกิด BER เพิ่มขึ้น ทำให้ข้อมูลที่รับได้เกิดความผิดพลาดขึ้น นอกจากนี้ยังศึกษาและวิเคราะห์ถึงช่องสัญญาณต่างๆที่ใช้ในการส่งผ่านข้อมูลระหว่างเครื่องรับกับเครื่องส่ง และวิธีการ Modulationแบบต่างๆ เช่น PSK , QAM และ BPSK เป็นต้น

โดยโครงงานชิ้นนี้ทำการเขียนโปรแกรม MATLAB ซึ่งใช้ GUI ในการแสดงผลเนื่องจากเป็นโปรมแกรมที่ใช้งานง่ายและมีการประมวลผลที่รวดเร็ว

1.2 วัตถุประสงค์ของโครงงาน

1. เพื่อศึกษาหลักการทำงานของระบบ MIMO
2. เพื่อวิเคราะห์ระบบและผลกระทบต่างๆที่เกิดขึ้นในระบบ MIMO
3. เพื่อศึกษาการใช้งานโปรแกรม MATLAB และ GUI
4. เพื่อให้ผู้ใช้งานสามารถออกแบบระบบ MIMOได้ง่าย

1.3 ขอบเขตการทำงาน

1. ศึกษาหลักการทำงานของระบบ MIMO
2. ศึกษาเกี่ยวกับช่องสัญญาณและผลกระทบที่ส่งผลให้ค่า SNR และ BER ของระบบเปลี่ยนแปลง
3. เขียนโปรแกรมการออกแบบระบบ MIMO ด้วย GUI
4. ทดสอบการทำงานของโปรแกรม GUI
5. เขียนรายงานและนำเสนอโครงงาน

1.4 ขั้นตอนการดำเนินงาน

1. ศึกษาหลักการทำงานของระบบ MIMO
2. ศึกษาเกี่ยวกับช่องสัญญาณและผลกระทบที่ส่งผลให้ค่า SNR และ BER
3. เขียนโปรแกรมแสดงค่าความสัมพันธ์ระหว่าง SNR และ BER เมื่อมีการเข้ารหัสแบบต่างๆ เช่น PSK , QAM และ BPSKเป็นต้น
4. ศึกษาการทำงานและเขียนโปรแกรมสร้าง ระบบ SM-STBC
5. เขียนโปรแกรมเพื่อการออกแบบระบบ MIMO
6. นำโปรแกรมที่ออกแบบได้มาแสดงผลใน GUI
7. ทดสอบและแก้ไขการใช้งานของโปรแกรม
8. สรุปผลการทดลองและเขียนรายงาน
9. นำเสนอโครงงาน

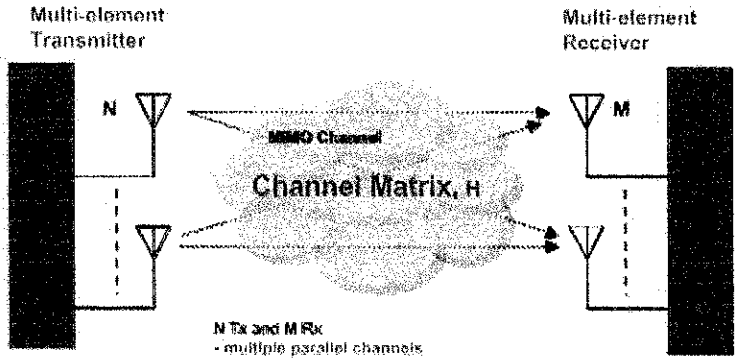
บทที่2
ทฤษฎีและหลักการ

2.1 ทฤษฎีและหลักการของ MIMO

ปัจจุบันการสื่อสารไร้สายมีการใช้อย่างแพร่หลาย ระบบสื่อสารที่มีการรับหรือส่งข้อมูลโดยใช้ชุดสายอากาศภาครับหรือภาคส่งจำนวนมากกว่าหนึ่งชุด (Multiple antenna) เป็นอีกรูปแบบหนึ่งของระบบสื่อสารที่มีความสำคัญ สามารถช่วยให้ระบบสื่อสารสามารถส่งข้อมูลด้วยปริมาณที่เพิ่มขึ้นได้ ระบบสื่อสารในลักษณะนี้ถูกเรียกว่าเป็นระบบสื่อสารแบบ Multiple-Input Multiple-Output (MIMO)

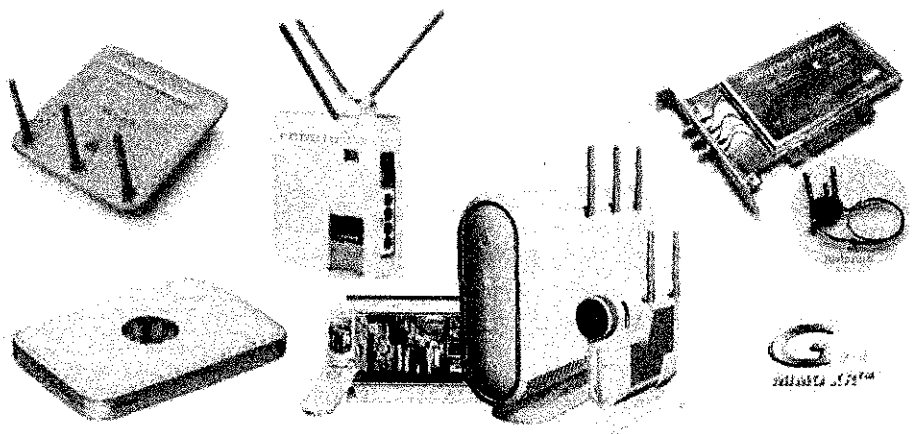
MIMO เป็นระบบ Wireless ที่สามารถให้ผู้ใช้ได้ในระยะทางที่มากขึ้นและประหยัดกว่าเดิม สำหรับ MIMO ถือว่าเป็น chip ที่ใช้เทคนิคที่เรียกว่า Orthogonal Frequency Division Multiplexing (OFDM)ที่มีการใช้งานในเครือข่ายโทรศัพท์ 802.11a/g และ Wimax อุปกรณ์ Wireless เวลาเจอสัญญาณสะท้อนมากๆ ก็จะไม่สามารถรับสัญญาณได้ เนื่องจากสัญญาณในตัวเองรบกวน ลักษณะของ MIMO จะเป็นการเพิ่มจำนวนเสาเข้าไปให้กับอุปกรณ์ Wireless เพื่อให้รับสัญญาณได้ดียิ่งขึ้น แม้ระยะที่ไกลออกไปหรือสัญญาณรบกวนมาๆ ก็สามารถรับส่งสัญญาณได้อย่างมีประสิทธิภาพ เทคนิค MIMOยังทำให้เราเตอร์มีระยะทำการที่เพิ่มขึ้นได้ด้วยการรับสัญญาณจากเสาอากาศหลายๆ อัน ทำให้สามารถเลือกสลับไปมายังเสาอากาศที่สามารถรับสัญญาณได้ดีกว่าได้ในทุกๆสถานการณ์

แนวความคิดเริ่มต้นของการสร้างระบบ MIMO คือ ผลรวมของสัญญาณจากภาคส่งหรือภาครับอันเนื่องมาจากสายอากาศหลายต้นทำให้คุณภาพของข้อมูลหรือบิตผิดพลาดน้อยลง อีกนัยหนึ่งคือทำให้อัตรารับส่งข้อมูลสูงขึ้น ระบบ MIMO มีความสามารถทนต่อสัญญาณรบกวนได้เป็นอย่างดี เนื่องจากมีการชดเชยสัญญาณที่ขาดหายไปด้วยสายอากาศต้นอื่นได้



รูปที่ 2.1 บล็อกไดอะแกรมของระบบ MIMO

จากรูปที่ 2.1 สัญญาณไบนารีซึ่งในที่นี้คือ 0100110 ส่งผ่านเข้าไปในภาคส่ง ผ่านการเข้ารหัส การมอดูเลตด้วยวิธีการต่าง ๆ เช่น QPSK (Quaternary Phase-Shift Keying), M-QAM (Multilevel-Quadrature AmplitudeModulation) หรือวิธีการอื่น ๆ ซึ่งทำให้เกิดการแยกสัญญาณโดยที่แต่ละสัญญาณจะได้รับการจับคู่กับแต่ละสายอากาศ และการจับคู่จะมีการถ่วงน้ำหนักแก่สายอากาศแต่ละต้น เมื่อปรับความถี่ให้สูงขึ้น กรองและขยายสัญญาณ แล้วนำสัญญาณส่งผ่านช่องสัญญาณ โดยที่ภาครับสัญญาณจะถูกดึงออกโดยสายอากาศแต่ละต้นตามค่าถ่วงน้ำหนัก ดีมอดูเลต และถอดรหัสเพื่อให้ได้สัญญาณเดิมกลับมา ทั้งนี้การเลือกใช้วิธีการเข้ารหัสและวิธีการจับคู่มีได้หลากหลายขึ้นกับการประยุกต์ใช้งาน



รูปที่2.2 ตัวอย่างผลิตภัณฑ์ซีรีส์ **Pre-N** ที่ผนวกรวมความสามารถของเทคโนโลยี **MIMO**

ข้อดีของเทคนิคแบบ MIMO คือ การเพิ่มความเร็วจะไม่ต้องถูกจำกัด ด้วยย่านความถี่คลื่นวิทยุที่รัฐบาลเป็นผู้จัดสรรให้อีกต่อไป ซึ่งเป็นการก้าวข้ามทฤษฎีเก่าๆ โดยสิ้นเชิง เพราะแต่ก่อนนี้คนเคยเชื่อกันว่า เมื่อต้องการเพิ่มความเร็วในการรับส่งข้อมูล ก็ต้องเพิ่มความกว้างของช่วงความถี่ในการสื่อสาร (Bandwidth) เท่านั้น แต่เนื่องจากการใช้งานคลื่นวิทยุในช่วงความถี่ต่างๆนั้น รัฐบาลของแต่ละประเทศมีการควบคุมอย่างเข้มงวด โอกาสที่จะเพิ่มความเร็วจึงแทบเป็นไปไม่ได้เลย

แต่ข้อเสียของเทคนิคนี้คือ ความยุ่งยากในการออกแบบและผลิต เสาอากาศ ที่ต้องมีความเที่ยงตรงเป็นอย่างยิ่ง และเนื่องจากการจัดวางขนาดและระยะตำแหน่งของเสาอากาศในระบบ MIMO ต้องเหมือนกัน จึงจะสามารถสื่อสารกันได้ ทำให้ตัวมาตรฐาน 802.11n ผ่านการรับรองได้

ยากขึ้นไปอีก เพราะผู้ผลิตรายต่างๆ มีมุมมองในด้านการออกแบบและผลิตเสาอากาศไม่เหมือนกัน ปัจจุบันนี้ก็เริ่มมีอุปกรณ์ WLAN ที่ใช้เทคนิค MIMO วางตลาดในสหรัฐฯ บ้างแล้ว แต่ผู้ที่ตัดสินใจใช้อุปกรณ์เหล่านั้นจะต้องเสี่ยงกับปัญหา Compatibility ในอนาคตเอาเอง ถ้าหากไม่เข้ากับมาตรฐาน 802.11n ที่กำลังจะออกมา

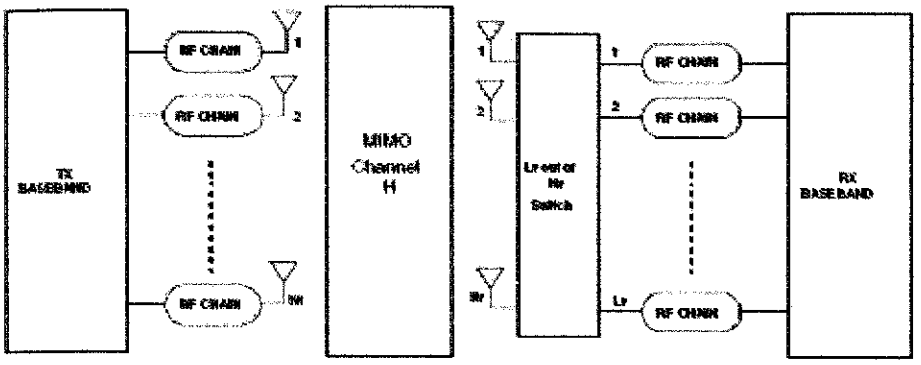
การประยุกต์ใช้เทคโนโลยี MIMO จะเป็นประโยชน์อย่างยิ่งสำหรับผู้ใช้หรือองค์กรการเชื่อมต่อไม่แน่นอนหรือ ไม่มีความเสถียร หรือใรบริเวณจุดที่อับสัญญ่ณ สำหรับอุปกรณ์สื่อสารไร้สายแบบที่ใช้งานในปัจจุบัน อีกทั้งมีการพัฒนาเพื่อใช้งานภายในบ้านแบบดิจิตอลเช่น การใช้เครือข่ายไร้สายเพื่อรับชมวีดีโอผ่านทางเครือข่ายได้ทั่วทุกมุมทุกห้องของบ้านได้อย่างต่อเนื่อง หรือหากโทรทัศน์ในอนาคตมีคุณสมบัติที่เชื่อมต่อแบบไร้สายได้แล้วอาจจะประยุกต์ใช้เทคโนโลยี MIMO ในการดาวน์โหลดข้อมูลจากกล้องวีดีโอแบบดิจิตัลมายังโทรทัศน์ได้ นอกจากนี้ในอุตสาหกรรมผลิตอุปกรณ์มือถือแบบไร้สายก็มีการพัฒนาอุปกรณ์โดยใช้เทคโนโลยี MIMO และอิงมาตรฐาน 802.11n เพื่อให้โทรศัพท์ที่ใช้เทคโนโลยี Voice-over-Internet Protocal (VoIP) สามารถใช้งานได้อย่างราบรื่น

2.2 ช่องสัญญาณของระบบ MIMO

ช่องสัญญาณในระบบMIMO ที่ใช้เป็นสื่อกลางในการส่งผ่านข้อมูลระหว่าง Transmitter และ Receiver ได้ถูกกำหนดให้อยู่ในรูปของเมตริกซ์ H ซึ่งช่องสัญญาณที่ใช้ในระบบนี้มีหลากหลายช่องสัญญาณ โดยเลือกใช้ตามความเหมาะสมของสภาพแวดล้อมในแต่ละที่

ช่องสัญญาณที่ทางโครงงานให้ความสนใจและทำการศึกษามี2ช่องสัญญาณด้วยกันคือ

1. ช่องสัญญาณที่มีการลดทอนแบบเรเลย์ (Rayleigh-fading Channels)
2. ช่องสัญญาณที่มีการลดทอนแบบริเชียน (Rician-fading Channels)



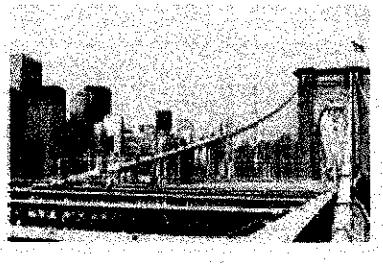
รูปที่ 2.3 ช่องสัญญาณในระบบ MIMO

2.2.1 ช่องสัญญาณที่มีการลดทอนแบบเรเลย์ (Rayleigh-fading Channels)

ช่องสัญญาณแบบนี้เป็นช่องสัญญาณที่เหมาะกับสภาพแวดล้อมที่มีวัตถุหรือสิ่งก่อสร้างจำนวนมากโดยจะเกิดการกระเจิงของสัญญาณวิทยุก่อนที่จะมาถึงเครื่องรับ ทำให้การรับส่งสัญญาณมีลักษณะเป็นแบบ Non Line Of Sight (NLOS) ซึ่งมีทฤษฎีหนึ่งกล่าวไว้ว่า ถ้ามีการกระเจิงที่มากพอ ช่องสัญญาณจะไปกระตุ้นให้เกิดรูปแบบของเกาส์เซียน (Gaussian)
โดยวิธีนี้ไม่ได้คำนึงถึงการกระจายตัวของแต่ละคอมโพแนนซ์ แต่ถ้าไม่มีการการกระเจิง ค่าเฉลี่ยเกาส์เซียนก็จะเป็นศูนย์ ซึ่งความน่าจะเป็นของการแปรค่าแบบสุ่มR สามารถเขียนเป็นความสัมพันธ์ได้ดังนี้

$$p_R(r) = \frac{r}{\Omega} e^{-r^2/\Omega}, r \geq 0 \qquad (2.1)$$

เมื่อ $\Omega = E(R^2)$



รูปที่2.4 **สภาพแวดล้อมที่เหมาะต่อ**ช่องสัญญาณที่มีการลดทอนแบบเรเลย์

ความสัมพันธ์ระหว่างINPUT และOUTPUT ของระบบช่องสัญญาณที่มีการลดทอนแบบเรเลย์ โดยมีสายอากาศภาครับ (N_r) และภาคส่ง (N_t) ดังแสดงในรูปที่ 2.3 ทำให้เกิดช่องสัญญาณที่มีเมตรกซ์ H แสดงได้ดังนี้

$$y=\sqrt{\frac{\rho}{N_t}}Hx+n \qquad (2.2)$$

เมื่อ y คือ $N_r \times 1$ เวกเตอร์ของสัญญาณภาครับ

x คือ $N_t \times 1$ เวกเตอร์ของสัญญาณภาคส่ง

n คือ สัญญาณรบกวนแบบเกาส์เซียน

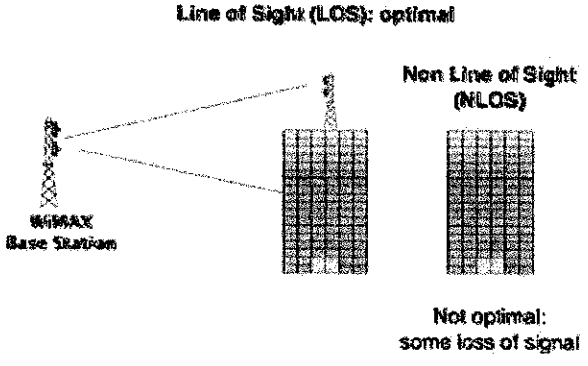
ρ คือ ค่าเฉลี่ยของ signal-to-noise ratio (SNR)

และ H คือ เมตริกซ์ของช่องสัญญาณMIMO ที่มีขนาด $N_r \times N_t$

$$H=\begin{pmatrix} h_{11} & \cdots & h_{1N_t} \\ \vdots & \ddots & \vdots \\ h_{N_r1} & \cdots & h_{N_rN_t} \end{pmatrix} \qquad (2.3)$$

2.2.2 ช่องสัญญาณที่มีการลดทอนแบบริเซียน (Rician-fading Channels)

ช่องสัญญาณแบบริเซียนมีความแตกต่างจากช่องสัญญาณแบบเรเลย์ตรงที่ ช่องสัญญาณประเภทนี้ไม่ต้องการสภาพแวดล้อมที่มีการกระเจิงสูง เหมาะกับสภาพแวดล้อมที่ไม่มีสิ่งกีดขวางมากนัก นั่นคือการรับส่งสัญญาณเป็นแบบ Line Of Sight (LOS)



รูปที่2.5 การรับ-ส่งสัญญาณแบบ Line Of Sight (LOS)

จากระบบดังกล่าวสามารถเขียนให้อยู่ในรูปของเมตริกซ์ได้ดังนี้

$$H = \sqrt{1-r}G + \sqrt{r}H_m \tag{2.4}$$

เมื่อ G คือ Independent and Identically Distributed

H_m คือ deterministic matrix

r คือ fraction of channel energy devoted to specular component

โดย H_m หาได้จากสมการ

$$H_m = \sqrt{NM}e_M e_N^T = \begin{bmatrix} \sqrt{NM} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 0 \end{bmatrix} \tag{2.5}$$

เมื่อ M คือ จำนวนของสายอากาศภาคส่ง

N คือ จำนวนของสายอากาศภาครับ

และจากระบบแบบริเชียนได้นิยามค่า K-factor คือ

$$\mathrm{K} = \frac{\text{Power in seculars component}}{\text{Power in random component}} \tag{2.6}$$

ซึ่ง K-factor มีความสัมพันธ์กับค่า r ดังนี้

$$K = \frac{\sqrt{r}}{\sqrt{1-r}} \tag{2.7}$$

ดังนั้น สามารถแก้สมการเพื่อหาค่า r ได้ดังนี้

$$K^2 = \frac{r}{1-r}$$

$$K^2(1-r) = r$$

$$K^2 - rK^2 = r$$

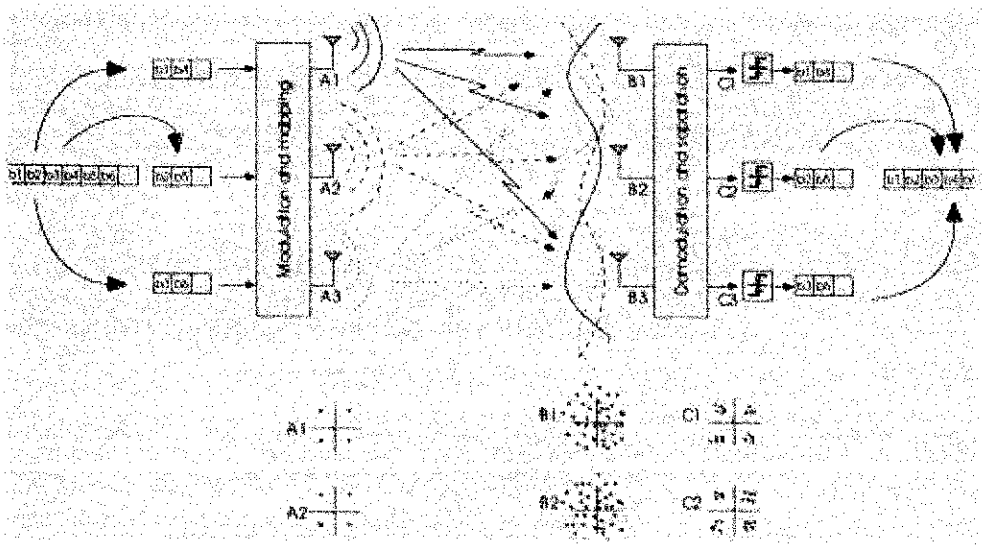
$$K^2 = r(1-K^2)$$

$$r = \frac{K^2}{1-K^2} \qquad (2.8)$$

เมื่อทราบค่าพารามิเตอร์ต่างๆครบแล้ว ก็สามารถสร้างช่องสัญญาณที่มีการลดทอนแบบ**ริเชียนได้**

2.3 เทคนิคการส่งสัญญาณ

เทคนิคในการส่งสัญญาณจากภาครับไปยังภาคส่งในระบบMIMOมีหลายเทคนิดด้วยกัน **แต่เทคนิคที่ทางโครงงานใช้คือเทคนิคที่เรียกว่า** Spatial Multiplexing **หรือเรียกสั้นว่า** SM โดยการ **ส่งข้อมูลในระบบนี้จะเป็นอิสระต่อกันและมีการแยกสัญญาณในการเข้ารหัสข้อมูล หรือที่เรียกว่า steams**



รูปที่2.6 เทคนิคการส่งสัญญาณแบบ SM

ในระบบ SM-MIMO เวกเตอร์ภาคส่ง x มักจะถูกแสดงได้ดังนี้

$$x = \left[x_1, x_2, ..., x_{N_t} \right]^T \tag{2.9}$$

ดังนั้นเวกเตอร์ภาครับ y จึงสามารถแสดงได้ว่า

$$\mathbf{y = Hx + n} \tag{2.10}$$

เมื่อ **H** คือ ช่องสัญญาณเมตริกซ์ขนาด Nr × Nt

n คือ เวกเตอร์สัญญาณรบกวนแบบAWGN

2.4 พารามิเตอร์ที่สำคัญ

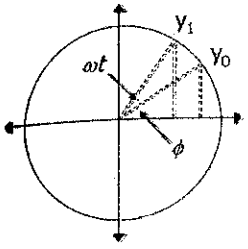
2.4.1 การมอดูเลชั่นแบบ Phase Shift Keying (PSK)

เมื่อต้องการจะส่งสัญญาณหรือข้อมูลผ่านช่องทางการสื่อสาร จำเป็นต้องอาศัยพลังงานไฟฟ้าช่วยพาสัญญาณเหล่านั้นให้เคลื่อนย้ายจากที่หนึ่งไปยังอีกที่หนึ่ง ขบวนการในการเพิ่มพลังงานไฟฟ้าดังกล่าว เรียกว่า "การมอดูเลต" หรือการกล้ำสัญญาณที่เป็นพลังงานไฟฟ้าที่มีความถี่สูงและคงที่ รวมทั้งแอมปลิจูด (ขนาด) สูงด้วยเรียกว่า "สัญญาณคลื่นพาห์" (Signal Carrier) อุปกรณ์สำหรับมอดูเลตสัญญาณ (Modulator) จะสร้างสัญญาณคลื่นพาห์ และรวมเข้ากับสัญญาณข้อมูลเพื่อให้สัญญาณมีความแรงพอที่จะส่งผ่านสื่อกลางไปยังอีกจุดหนึ่ง เมื่อถึงปลายทางจะมีอุปกรณ์ในการแยกสัญญาณคลื่นพาห์ออก เรียกวิธีการแยกสัญญาณนี้ว่า"การดีมอดูเลต" (Demodulation)

การมอดูเลตแบบ PSK คือการเปลี่ยน เฟส (ตำแหน่งเริ่มต้น) ของสัญญาณพาหะ ตามบิตข้อมูล เช่น เฟส $\phi 0$ เมื่อข้อมูลเป็น 0 เฟส $\phi 0$ เมื่อข้อมูลเป็น 1 ดังสมการ

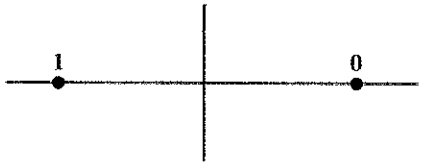
$$PSK(t) = \begin{cases} A\sin(2\pi f_c t + \phi 0) & , data(t) = 0 \\ A\sin(2\pi f_c t + \phi 1) & , data(t) = 1 \end{cases} \tag{2.11}$$

การผสมสัญญาณแบบ PSK ในรูปแบบต่างๆ สามารถแสดงโดยแผนผัง Constellation ซึ่งบอกตำแหน่ง Phase ของสัญลักษณ์ จากตัวอย่าง 1 Symbol = 1 Bit เรียกว่า Binary PSK ดังรูป



Bit	Phase
0	0
1	180

Bits



Constellation diagram

รูปที่2.7 Constellation Diagram ของการผสมสัญญาณแบบ Phase Shift Keying แบบ Binary PSK ซึ่งใช้เฟส 2 เฟส แทนข้อมูล 0 และ 1

จุดเด่นของ PSK คือสามารถทนทานต่อสัญญาณรบกวนมากกว่า ASK ในขณะที่ใช้แถบความถี่แคบกว่า FSK และ นอกจากนี้ยังสามารถแทนสัญลักษณ์ได้มากกว่า 1 บิตต่อหน่วยสัญญาณ

จุดด้อยของ PSK คือ การออกแบบเครื่องรับ/ส่ง และวิเคราะห์ ทำได้ยากกว่า ASK และ FSK

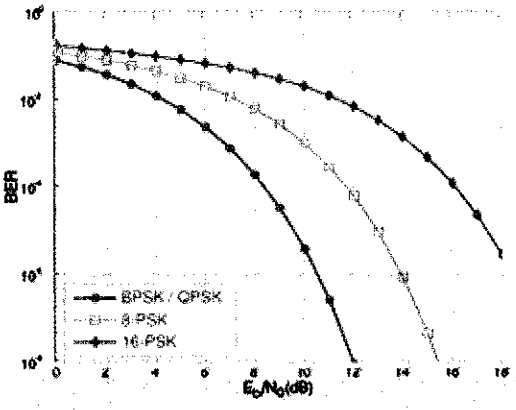
2.4.2 อัตราความผิดพลาดของการส่งข้อมูล

อัตราความผิดพลาดของการส่งข้อมูล(Bit Error Rate ,BER) คำนวณได้จากสัดส่วนเปอร์เซ็นต์ของจำนวนบิตที่ส่งผิดจากจำนวนบิตทั้งหมดตังสมการข้างล่าง ซึ่งค่าBERที่สูงนั่นหมายถึงการให้บริการสื่อสารข้อมูลที่มีคุณภาพต่ำ ทำให้ต้องมีการส่งใหม่ (Retransmission) และทำให้ เกิดการล่าช้า ในการส่งข้อมูล ดังนั้น การให้บริการสื่อสารข้อมูลที่ดี จำเป็นที่จะต้องมีระบบเครือข่ายที่สามารถรับ-ส่งข้อมูล ได้ด้วย BER ที่ต่ำ

$$\text{BER} = \frac{\text{Number of bit error}}{\text{Number of bit received}} \qquad (2.12)$$

$$\text{M-PSK} \quad : \quad P_e = 2Q\left(\sqrt{\frac{4E_s}{N_o}}\sin\left(\frac{\pi}{2M}\right)\right) \qquad (2.13)$$

จากสมการดังกล่าวเป็นสมการที่ใช้สำหรับระบบการรับ-ส่งข้อมูลที่มีสายอากาศรับ-ส่งเพียงอย่างละต้นเท่านั้น



รูปที่2.8 กราฟแสดงค่าBERที่มีการเข้ารหัสแบบ BPSK, QPSK, 8-PSK and 16-PSK.

แต่สำหรับระบบ MIMO ที่มีสายอากาศหลายต้นทำให้ไม่อาจใช้สมการดังกล่าวได้ จึงจำเป็นต้องสร้างโปรแกรมจำลองเหตุการณ์เพื่อใช้หา อัตราความผิดพลาดของการส่งข้อมูลในระบบ MIMO ขึ้น

2.4.3 Signal-to-noise ratio

Signal-to-noise ratio (SNR) คืออัตราส่วนระหว่างสัญญาณจริงที่ได้รับต่อสัญญาณที่เป็นสัญญาณรบกวน มีค่าเป็นเดซิเบล (dB) โดยหากค่า SNR มีค่าต่ำมากๆ นั้นหมายความว่าความแรงของสัญญาณรบกวน พอๆ กับสัญญาณจริง ทำให้มีการรับส่งข้อมูลได้ไม่ค่อยดี อาจจะทำให้ Sync ติดๆหลุดๆได้

ในระบบหนึ่งๆถ้ามีแรงดันของสัญญาณ Vs เข้ามา พร้อมทั้งเกิดแรงดันของสัญญาณรบกวน Vn ด้วย ค่าSNRของระบบจะสามารถอธิบายได้ดังสมการ

$$S/N = 20 \log_{10}(V_s/V_n) \qquad (2.14)$$

ถ้า $V_s = V_n$ จะได้ว่า $S/N = 0$

ในทางอุดมคติ V_s จะมีความแรงของสัญญาณมากกว่า V_n ดังนั้น S/N จะมีค่าเป็นบวก ยกตัวอย่างเช่น V_s = 10.0 microvolts และ V_n = 1.00 microvolt จะได้ว่า

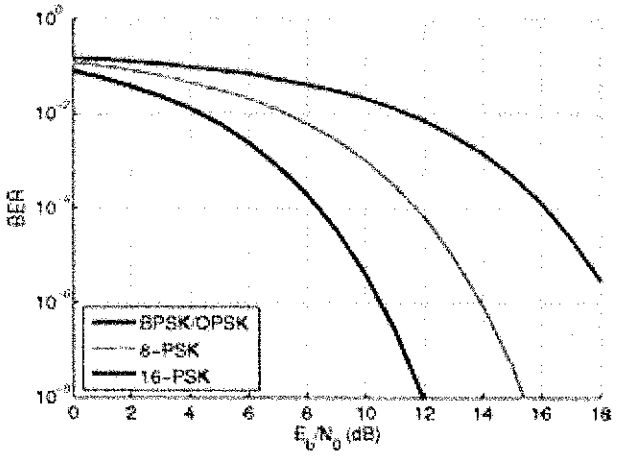
$$S/N = 20 \log_{10}(10.0) = 20.0 \text{ dB}$$

ซึ่งผลลัพธ์ดังกล่าวเป็นค่าที่พบโดยทั่วไป แต่ถ้าความแรงของสัญญาณมีค่าอ่อนมากแต่ยังคงมากกว่าสัญญาณรบกวน เช่น V_s/V_n = 1.30 microvolts จะได้ว่า

$$S/N = 20 \log_{10}(1.30) = 2.28 \text{ dB}$$

จากตัวอย่างข้างต้นอาจเกิดการลดลงของความเร็วในการส่งข้อมูลภายใต้เงื่อนไขดังกล่าว

ถ้า V_s มีความแรงของสัญญาณน้อยกว่า V_n จะได้ค่า S/N มีค่าเป็นลบ ในสถานการณ์เช่นนี้อาจไม่เกิดขึ้นเลยเว้นเสียแต่ว่าจะมีการเพิ่มความแรงของสัญญาณ และ/หรือ ลดระดับของสัญญาณรบกวนที่ปลายทางหรือภาครับนั่นเอง



รูปที่2.9 กราฟแสดงความสัมพันธ์ระหว่าง BER กับ SNR โดยใช้การเข้ารหัสสัญญาณแบบ BSPK/QPSK ,8PSK และ 16PSK

ศูนย์บรรณสารและสื่อการศึกษา
มหาวิทยาลัยเทคโนโลยีสุรนารี

2.5 หลักการของ GUI

GUI ย่อมาจาก **Graphical User Interface** (ซึ่งในรายงานฉบับนี้ขอแปลว่าโปรแกรมโต้ตอบกับผู้ใช้แบบกราฟ) คือโปรแกรมโต้ตอบกับผู้ใช้ซึ่งถูกสร้างจากวัตถุแบบกราฟ (องค์ประกอบต่างๆ ของโปรแกรมโต้ตอบกับผู้ใช้แบบกราฟ) ได้แก่ ปุ่มกด (push buttons) ตัวเลือกแบบเมนู (pop-up menu) กราฟ (axes) ฯ หากโปรแกรมโต้ตอบกับผู้ใช้ได้รับการออกแบบที่ดีจะทำให้ผู้ใช้สามารถเข้าถึงการทำงานหรือใช้งานได้อย่างไม่ต้องทราบขั้นตอนการทำงานมาก่อนเลย Graphical User Interface (GUI) เป็น user interface ที่สร้างขึ้นด้วย graphical object แบบต่าง ๆ เช่น ปุ่มเมนู slider โดยทั่วไป objects เหล่านี้ ผู้ใช้คอมพิวเตอร์ส่วนใหญ่เข้าใจถึงความหมายและวิธีการใช้ objectเหล่านี้เป็นอย่างดีอยู่แล้ว สิ่งสำคัญที่เราจะกล่าวถึงคือ หลังจากผู้ใช้ได้มีการกดปุ่มเมาส์ เลื่อน slider หรือเลือกเมนู เราจะมีวิธีการกำหนดให้เกิดขั้นตอนต่อ ๆ ไป ตามที่เราต้องการได้อย่างไรApplication ต่าง ๆ ที่ออกแบบมาเป็น GUI จะพบว่าสามารถทำให้ผู้ใช้เข้าใจการใช้ application นั้นได้อย่างรวดเร็ว เพราะแทบจะไม่มีชุดคำสั่งใดให้จดจำ การทำงานของ application จะเกิดขึ้นทันทีที่ได้input จากผู้ใช้

GUI เป็นอินเตอร์เฟซด้วยกราฟฟิคของผู้ใช้กับเครื่องคอมพิวเตอร์ เช่น เว็บบราวเซอร์ คำนี้เกิดขึ้น เนื่องจากการอินเตอร์เฟซกับคอมพิวเตอร์ในรุ่นแรกไม่ได้ใช้กราฟฟิค แต่เป็นการใช้ตัวอักษรและแป้นพิมพ์ ปกติจะเป็นคำสั่งที่จำได้ เช่น ระบบปฏิบัติการ DOS ในขั้นกลางการอินเตอร์เฟซของผู้ใช้เป็นการอ่านอินเตอร์เฟซแบบเมนู (Menu-based Interface) ซึ่งยอมให้ใช้เมาส์คลิกคำสั่งได้ นอกจากการพิมพ์แป้นพิมพ์ ระบบปฏิบัติการส่วนใหญ่จะเป็นแบบ GUI ในส่วนโปรแกรมประยุกต์จะใช้องค์ประกอบของ GUI ที่มากับระบบปฏิบัติการและเพิ่มการอินเตอร์เฟซของตัวเองเข้าไป บางครั้ง GUI ใช้อ็อบเจคมากกว่าหนึ่งในการทำงานจริง เช่น ในเครื่องตั้งโต๊ะการมองผ่าน Windows จะพบส่วนประกอบของ GUI ได้รวมถึง Windows เมนูแบบ Pull down, ปุ่ม, แถบเลื่อน, ไอคอน, Wizards, เมาส์ และรวมถึงอีกหลายสิ่งที่กำลังพัฒนา การเพิ่มขึ้นของการใช้มัลติมีเดีย เช่นเสียง ภาพเคลื่อนไหว และการอินเตอร์เฟซแบบเสมือนจริงกำลังเข้ามาเป็นส่วนหนึ่งของ GUI ความคุ้นเคยกับ GUI ในปัจจุบันทั้ง Mac, ระบบปฏิบัติการ Windows และโปรแกรมประยุกต์ต่าง ๆ ได้มีจุดเริ่มต้นที่ Xerox Palo Alto Research Laboratory ต่อมาในทศวรรษที่ 70 Apple ใช้ครั้งแรกในเครื่องคอมพิวเตอร์แมคอินทอช ต่อมา Microsoft ได้นำเอาแนวคิดมาพัฒนาเป็นระบบปฏิบัติการ Windows กับเครื่อง IBM- compatible เมื่อมีการสร้างโปรแกรมประยุกต์เครื่องมือแบบ object-oriented จะเขียนการอินเตอร์เฟซด้วยกราฟฟิค ในแต่ละสมาชิกของ GUI จะเรียกว่า class form เมื่อสร้างอ็อบเจคขึ้นมาแล้ว ซึ่งสามารถเขียนคำสั่งหรือปรับปรุงด้วยเมธอด (method) เพื่อทำให้อ็อบเจคเหล่านั้นตอบสนองต่อการใช้งานของผู้ใช้

ประโยชน์ที่เห็นได้อย่างชัดเจนของ GUI คือการที่ผู้ใช้สามารถทำงานกับโปรแกรมได้โดยไม่ต้องผ่านทางการเขียนคำสั่งทีละบรรทัดดังเช่นการเขียนโปรแกรมตามปกติ กล่าวคือผู้ใช้เพียงแต่ปฏิบัติตามคำสั่งสำเร็จรูปที่ทางผู้ออกแบบได้จัดหามาให้และทำการใช้งานได้ทันที ดังนั้นจึงเป็นการง่ายกว่าสำหรับผู้ใช้ทั้งในด้านการเรียนรู้และการใช้งานโปรแกรม

GUIDE เป็นเครื่องมือหลักที่เราจะใช้สร้าง GUI การสร้างโปรแกรมโต้ตอบกับผู้ใช้แบบกราฟ(GUI)ผ่านทาง GUIDE ซึ่งรวมถึงการ วางแบบ (laying out) องค์ประกอบ (components) และโปรแกรมองค์ประกอบเหล่านั้นเพื่อให้ทำงานตอบสนองต่อผู้ใช้ตามที่ผู้ออกแบบต้องการ

2.5.1 การสร้าง GUI ด้วย GUIDE

MATLAB จะสร้าง GUI อยู่บนหน้าต่างรูปภาพ (figure window) ซึ่งภายในหน้าต่างนี้จะมีส่วนประกอบต่าง ๆ อยู่ได้ไม่ว่าจะเป็น axes, uicontrol หรือวัตถุอื่น ๆ ตามที่เราได้กล่าวถึงมาแล้วในบทก่อนหน้านี้ ใน MATLAB version ก่อนหน้านี้ เราสามารถที่จะสร้าง uicontrol , uimenu แบบต่าง ๆ ลงในหน้าต่างรูปภาพได้แต่เป็นไปด้วยความลำบากเพราะการสร้างเป็น text base ต่อมาจนกระทั่ง version5MATLAB ได้สร้าง Graphical User Interface Development Environment หรือ GUIDE ขึ้นเพื่อช่วยให้เราสร้างบันทึก และแก้ไป GUI ได้สะดวกขึ้น
การสร้าง GUI จะประกอบด้วยขั้นตอนสองขั้นตอน

- กำหนดและวางส่วนประกอบต่าง ๆ ลงบน GUI
- เขียนโปรแกรมเพื่อกำหนดการทำงานของส่วนประกอบต่าง ๆ ใน GUI

GUIDE นั้นโดยหลักใหญ่แล้วจะมีหน้าที่ในการวางส่วนประกอบที่เราต้องการให้มีลงใน GUI จากนั้น GUIDE จะสร้าง M-file ที่บรรจุ handle ของวัตถุหรือ object ทั้งหมดที่เราสร้างขึ้นรวมทั้งคำสั่งให้ GUI ทำงาน นอกเหนือจากนั้น M-file จะให้แนวทางในการเขียนฟังก์ชัน ที่ทำงานหลังจากผู้ใช้กดเมาส์ปุ่มซ้ายหรือปรับเปลี่ยนค่าของวัตถุนั้น ซึ่งเราเรียกว่า callback ของวัตถุนั้น

ส่วนประกอบของ GUI ใน MATLABดังที่ได้กล่าวมาก่อนแล้วว่าเราสามารถสร้าง GUI ขึ้นมาได้โดยการเขียนเป็น M-file ขึ้นมาล้วน ๆ แต่การใช้ GUIDE จะทำให้การทำงานง่ายขึ้นมาก เพราะจะช่วยให้เรากำหนดตำแหน่งของวัตถุต่าง ๆ ได้โดยง่าย หลังจากนั้น GUIDE จะสร้างไฟล์ขึ้งมา 2 ไฟล์เพื่อเก็บและนำ GUI ของเรามาใช้ต่อไปซึ่งจะประกอบด้วย

- FIG-file ซึ่งจะบรรจุรายละเอียดของวัตถุต่างที่เป็นองค์ประกอบอยู่ในหน้าต่างรูปภาพที่เป็น GUI ของเรา
- M-file ที่จะบรรจุฟังก์ชันที่กำหนดการทำงานของ GUI ของเรา รวมถึง callback ทั้งหมด

ซึ่ง callback เหล่านี้จะบรรจุเป็น sub function อยู่ใน M-file และเราจะเรียก M-file ที่ควบคุมการทำงานของ GUI นี้ว่า Application M-file ดังนั้น Application M-file จะไม่มีข้อมูลใด ๆ เกี่ยวกับรูปแบบของส่วนประกอบที่บรรจุอยู่ในGUI เช่นสี ขนาด ตำแหน่ง หรือ อื่น ๆ เลย เพราะข้อมูลเหล่านั้นจะบรรจุอยู่ใน FIG-file

2.5.2 ส่วนประกอบสำคัญของ Application M-file ที่สร้างโดย GUIDE

GUIDE จะรวบรวมองค์ประกอบต่าง ๆ ภายใน GUI แล้วสร้าง Application M-file โดยอัตโนมัติโดยมีรูปแบบของการสร้างที่ชัดเจน เพื่อให้เราได้โครงสร้างของ Application M-file จากนั้นเราสามารถนำโครงสร้างที่สร้างโดยอัตโนมัตินั้นมาปรับแก้ เพื่อให้เกิดการควบคุม GUI ตามที่เราต้องการ การกระทำดังกล่าวทำให้เราได้ข้อได้เปรียบหลายประการ เช่น

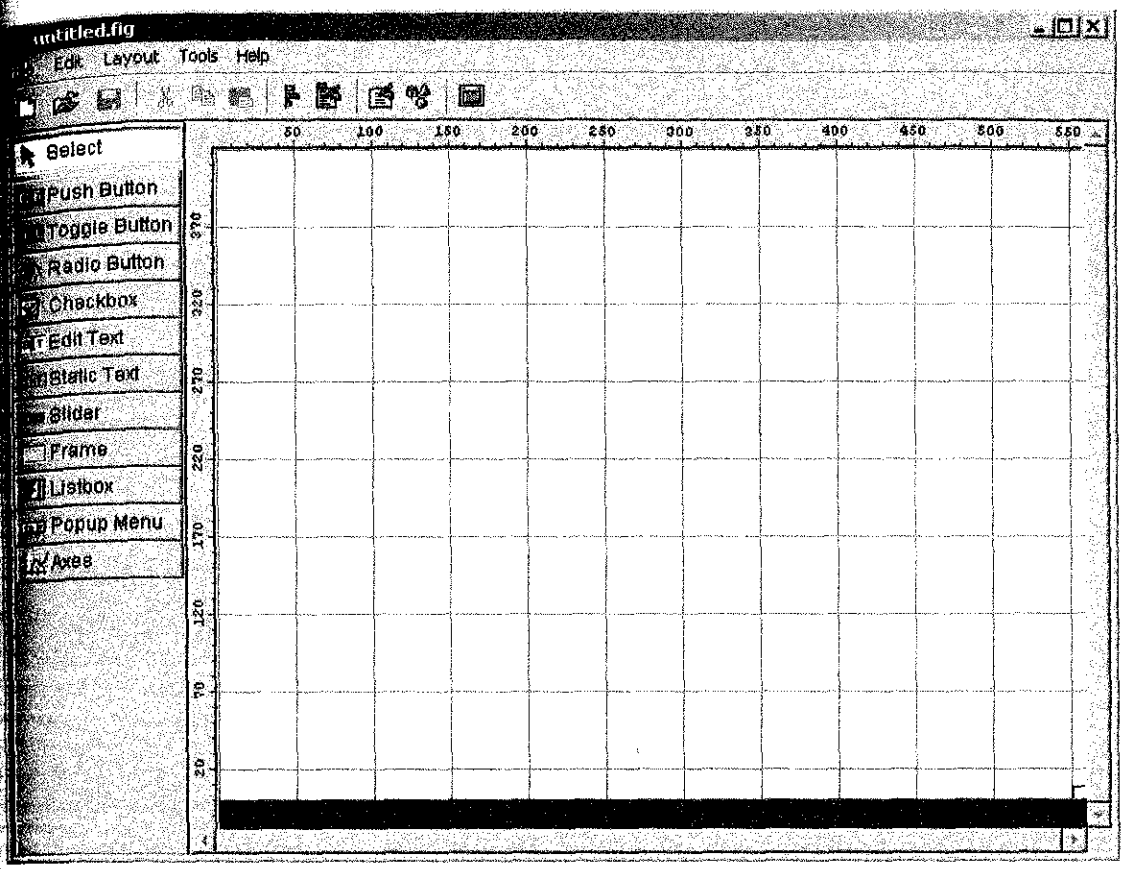
- M-file จะประกอบด้วยคำสั่งที่จำเป็นในการควบคุม GUI ครบถ้วน
- M-file จะทำให้เราส่งข้อมูลไปที่ส่วนต่าง ๆ ได้ง่าย สะดวก รวดเร็ว
- การใช้ M-file จะทำให้เราส่งข้อมูลไปที่ส่วนต่าง ๆ ภายใต้ MATLAB ได้ง่ายขึ้น
- Application M-file จะสร้าง Sub function สำหรับ uicontrols ทุกแบบที่มีใน GUI เพื่อทำให้เราเขียน callback ต่าง ๆ ได้สะดวกขึ้นแม้ว่า GUIDE จะให้ทางเลือกกับเราว่าจะให้ GUIDE สร้างเฉพาะ fig-file เพื่อเก็บและใช้เป็นข้อมูลของ GUI ที่สร้างขึ้นเพียงอย่างเดียว แล้วเราเขียน M-file ขึ้นมาเอง แต่สำหรับผู้เริ่มเขียน GUI บนMATLAB เราคิดว่าการสร้าง GUI ด้วย GUIDE จะสะดวกกว่า หากเราให้ GUIDE สร้าง Application M-fileให้เราด้วย ดังนั้นในการสร้าง GUI ด้วย GUIDE ที่เรานำเสนอในเอกสารนี้จะมีการกำหนดขั้นตอนดังนี้
- เลือก GUIDE Application option แล้วเลือกให้ GUIDE สร้างทั้ง FIG-file และ M-file
- การใช้ Layout Editor เพื่อวางรูปแบบของ GUI
- เรียนรู้การสร้าง Application M-file จาก GUIDE และเข้าใจถึงวิธีการทำเพื่อจะนำไปใช้ต่อ
- ปรับแก้ Application M-file ให้ทำงานตามที่เรากำหนด

1. การเลือก GUIDE Application Options

เมื่อเราต้องการจะใช้ GUIDE นั้น ครั้งแรก บน MATLAB COMMAND WONDOW ที่ prompt เราสั่ง

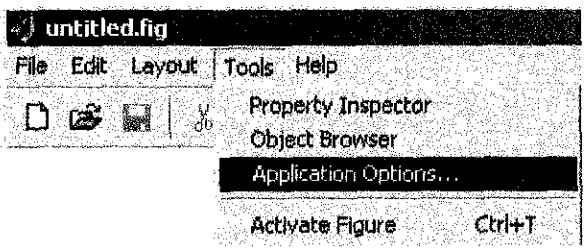
» guide

จากนั้น Layout Editor จะปรากฏขึ้น ซึ่งมีลักษณะดังรูปก่อน



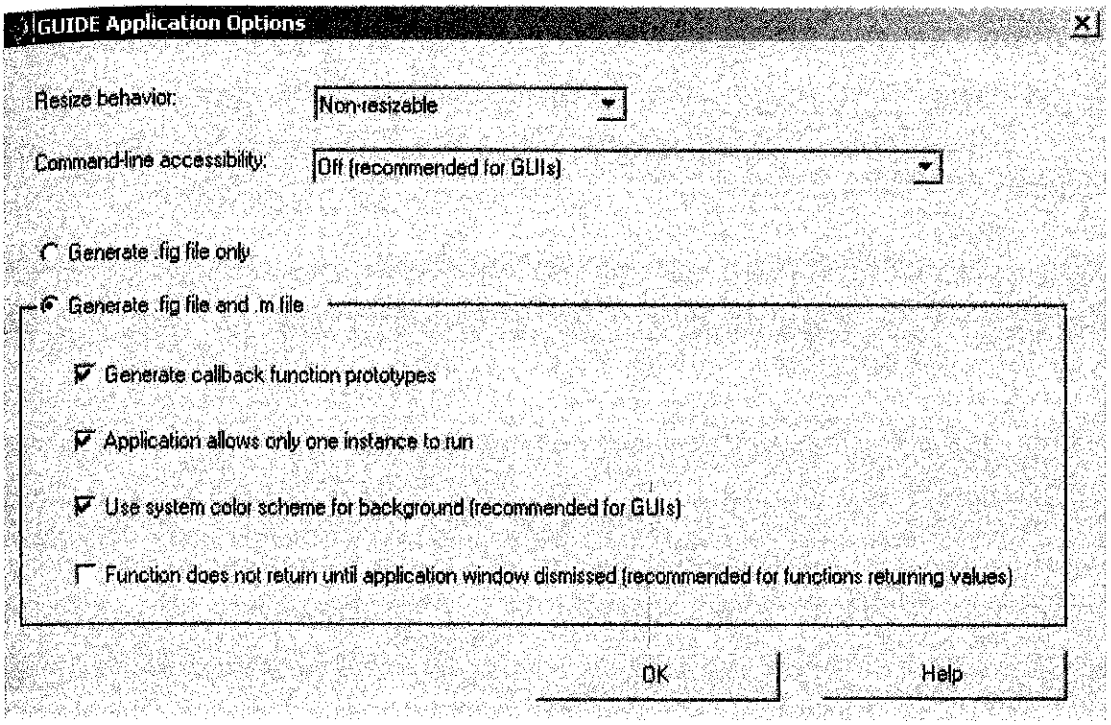
รูปที่ 2.10 Layout Editor

ก่อนที่จะทำการเพิ่มส่วนประกอบต่าง ๆ ลงใน GUI เราควรกำหนดตัวเลือกต่าง ๆ ก่อน โดยภายใต้เมนู Tool เลือก Application Options



รูปที่ 2.11 การควรกำหนดตัวเลือกต่าง ๆ ภายใต้เมนู Tool

ซึ่งจะทำให้เราได้ หน้าต่าง GUIDE Application Options ซึ่งมีลักษณะตามรูป



รูปที่ 2.12 หน้าต่าง GUIDE Application Options

นอกเหนือจากการเลือกให้ GUIDE จะสร้างเฉพาะ FIG-file หรือสร้างทั้ง Fig-file และ M-file แล้ว เรายังสามารถ กำหนดค่าต่าง ๆ ในหน้าต่างตัวเลือกนี้ได้ ซึ่งรายละเอียดมีดังนี้คือ

Resize Behavior

เป็นการกำหนดว่าผู้ใช้สามารถเปลี่ยนขนาดของหน้าต่าง GUI ที่สร้างขึ้นได้หรือไม่ และถ้าได้ จะให้MATLAB ควบคุมการเปลี่ยนขนาดโดยผู้ใช้อย่างไร ซึ่ง GUIDE ให้ตัวเลือก 3 แบบคือ

• Non-Resizable ผู้ใช้ไม่สามารถเปลี่ยนขนาดของหน้าต่างได้ (default)

• Proportional ให้ผู้ใช้สามารถปรับขนาดของหน้าต่าง GUI ได้โดย MATLAB จะปรับขนาดขององค์ประกอบต่าง ๆ ใน GUI ให้มีสัดส่วนตามขนาดของหน้าต่าง GUI ที่เปลี่ยนไป

• User-Specifide มีการเขียนโปรแกรมกำหนดให้ GUI ปรับเปลี่ยนขนาดและตำแหน่งขององค์ประกอบต่าง ๆ ใน GUI ซึ่งการเลือกตัวเลือกนี้ผู้เขียน GUI ต้องเขียนคำสั่งเพื่อปรับขนาดและตำแหน่งขององค์ประกอบต่าง ๆ ใน GUI ให้ชัดเจนสำหรับตัวเลือกตัวแรก เหมาะกับ GUI ที่ไม่ต้องการปรับปรุงขนาด ส่วนตัวเลือกที่สองProportional นั้นเหมาะสมกับการที่ให้ผู้ใช้ปรับปรุง

ขนาดหน้าต่างได้ แต่รายละเอียดต่าง ๆ ปล่อยให้เป็นไปโดยอัตโนมัติ และเราขอเตือนว่าหากปรับตำแหน่งด้วยตัวเลือก Proportional นี้ เมื่อปรับขนาดหน้าต่าง ขนาดตัวหนังสือใน GUI ไม่มีการปรับไปด้วย ดังนั้นหากเรากำหนดให้หน้าต่างมีขนาดเล็กเกินไป ตัวหนังสือต่าง ๆ อาจซ้อนทับกัน การปรับปรุงขนาดหน้าต่างด้วย User-Specify อาจจะเป็นการปรับปรุงหน้าต่างขนาดต่าง ๆ ได้หลากหลายกว่า แต่ต้องมีการเขียนโปรแกรมควบคุม อย่างไรก็ตามเรา
จะไม่ขอกล่าวถึงรายละเอียดในที่นี้

Command-Line Accessibility

เมื่อ MATLAB สร้าง graph จะ figure และ axes ที่จะต้องเป็น parents ของรูปกราฟนั้น ซึ่ง MATLABจะทำการมองหาก่อนว่ามี figure และ Axes เกิดขึ้นอยู่ในขณะนั้นหรือไม่ ถ้ามี MATLAB จะทำการเขียนกราฟลงใน figure และ Axes ที่มีอยู่ถ้าไม่ MATLAB จะทำการสร้าง figure และ Axes ขึ้นมาใหม่ในการสร้าง GUI ของเรานั้น ส่วนมากแล้ว เราคงไม่ต้องการให้ผู้ใช้เขียนกราฟลงใน axes ที่ปรากฏอยู่ใน GUI ของเรา แต่บางกรณีเราอาจจะต้องการให้ผู้ใช้เขียนกราฟลงใน axes ที่ปรากฏอยู่ในGUI ของเราก็ได้ ดังนั้น GUIDE จึงมีตัวเลือกให้เราเลือกสำหรับ Command-Line Accessibility ดังนี้

- Off - ป้องกันการสั่งเขียนกราฟผ่าน command-line บน GUI ของเรา (default)
- On – ให้มีการเขียนกราฟผ่าน command-line บน GUI ของเราได้
- User-Specified GUIDE จะให้ GUI ใช้ค่าที่กำหนดโดยคุณสมบัติ Handle Visibility และ Integer Handle ของ figure

Note คำสั่ง Findobj นี้จะใช้ไม่ได้กับวัตถุต่างที่อยู่ใน GUI Handle Visibility ของ Figure เป็น off เพราะเรากำหนดว่า handle ของ figure นี้จะมองไม่เห็นโดย MATLAB แม้ว่า figure นี้อาจจะปรากฏอยู่บนหน้าจอก็ตาม อย่างไรก็ตาม Application M-file จะมีข้อมูลเกี่ยวกับ handle ของวัตถุต่าง ๆที่เราสร้างใน GUI อยู่แล้วดังนั้นเราไม่จำเป็นต้องใช้คำสั่ง findobj เพื่อหา handle ของวัตถุต่างๆ สำหรับคุณสมบัติของ figure ที่เกี่ยวข้องกับการมองเห็น figure ที่สำคัญมี 2 ค่า คือ

- Handle Visibility ถ้าหากค่าคุณสมบัตินี้เป็น off ค่าของ handle ต่าง ๆ ที่เป็น children ของ figure นี้จะถูกลบออกจาก children ของ root object ทำให้ figure นี้ไม่เป็น current figurecurrent figure เป็นเป้าหมายในการสร้างกราฟของ MATLAB) อย่างไรก็ตาม handleเหล่านั้นยังคงใช้ได้ ดังนั้นคำสั่งต่าง ๆ ที่สั่งตรงถึง handle เหล่านั้นจึงเป็นไปได้
- Integer Handle ถ้าหากค่าคุณสมบัตินี้เป็น off แล้ว MATLAB จะกำหนดค่า handle ของวัตถุต่าง ๆ เป็นเลขจำนวนจริงที่จะไม่มีการกำหนดซ้ำอีกเช่น (68.0001224) แทนที่จะเป็นเลขจำนวนเต็ม ซึ่งจะเป็นการลดโอกาสที่จะมีวัตถุใน GUI อื่น ๆ ที่มี handle ซ้ำกับวัตถุใน GUI ของเรา

2.5.3 การสร้าง Application M-file ของ GUIDE

เมื่อเราสร้าง GUI โดย GUIDE และเลือกให้ GUIDE สร้าง FIG-file และ M-file เมื่อเราเลือกตัวเลือกนี้จะมีตัวเลือกให้ผู้ใช้เลือกเพิ่มขึ้นได้อีก 4 ตัวเลือกเพื่อกำหนดลักษณะของ Application M-file ซึ่งตัวเลือกต่างๆ มีดังนี้

- Generate callback function prototypes
- Application allows only one instance to run
- Use system color scheme for background
- Function does not return until application window dismissed

โดยรายละเอียดของตัวเลือกต่าง ๆ มีดังนี้

1. การสร้างต้นแบบของ Application M-file

เมื่อเราเลือกตัวเลือก Generate Callback Function Prototype ในการเลือกตัวของ GUIDE ApplicationOption ก็จะทำให้ GUIDE เพิ่ม sub function ให้กับ application M-file สำหรับทุกวัตถุที่เราสร้างขึ้นใน GUI(ยกเว้น frame และ Static text) อย่างไรก็ตาม GUEDE จะสร้างเฉพาะ sub function เป็นต้นแบบไว้เท่านั้นส่วนคำสั่งต่าง ๆนั้นเราต้องเป็นผู้เขียนใน sub function นั้นเอง นอกเหนือจากนั้น GUIDE ยังจะเพิ่ม subfunction ทุกครั้งเมื่อเราแก้ไข callback จากการใช้เมาส์ปุ่มขวาในเมนู contextสำหรับการสร้างต้นแบบของ callback sub function นั้นจะสร้างขึ้นโดยมีลักษณะดังนี้

function object.Tag_Callback(h,eventdata,handles.varagin)

โดย arguments ต่าง ๆ จะเป็นดังนี้

h เป็น handle ของวัตถุที่เรียก callback นี้

eventdata ว่างสำรองเก็บไว้ใช้ในอนาคต

handles เป็นตัวแปรแบบ structure ที่บรรจุ handle ของทุกวัตถุที่อยู่ใน GUI โดยชื่อของfileจะเป็นชื่อ tag ของวัตถุนั้น เราสามารถใช้ตัวแปรนี้ส่งข้อมูลเกี่ยวกับ handleของวัตถุต่าง ๆ ใน GUI ไปที่ callback ตัวอื่นหรือโปรแกรมตัวอื่น ๆ ในMATLAB ได้

varargin เป็น variable-length แสดงผลที่เราต้องการส่งผ่านไปที่ callback function

ตัวอย่างเช่น ถ้าเราวาง push button ที่เรากำหนด Tag เป็น pushbutton1 จะทำให้GUIDE สร้าง sub function ใน application ดังนี้

function pushbutton1_Callback(h,evendata,handles,varargin)

หลังจากนั้น GUIDE จะกำหนดคุณสมบัติ Callback ของ push button นี้เป็น

Mygui('pushbutton1_callback',gcbo,[],guidata(gcbo))

โดย

Mygui เป็นชื่อของ FIG-file ที่เก็บ GUI นี้

Pushbutton1 Callback-เป็นชื่อของ callback sub funcion

gcbo เป็นคำสั่งที่ให้ handle ของวัตถุนี้ ในที่นี้คือ push button

[ ] เป็น เมตริกส์ว่างใช้เป็นที่เก็บ eventdata

guidata(gcbo) เป็น handles structure ที่ได้จากข้อมูลที่บรรจุอยู่ใน figure ที่เป็น GUI

ถ้าหากเราต้องการจะส่งข้อมูลอื่น ๆ ผ่านต่อเข้าไปสู่ sub function นอกเหนือจากที่ GUIDE ได้สร้างเป็นต้นแบบ เราสามารถทำได้คดยการเพิ่ม argument เข้าไปด้วยการแยกข้อมูลต่าง ๆ ด้วยเครื่องหมาย , ซึ่งการแก้ไขจะต้องทำทั้งใน (1) sub function ใน application M-file และ (2) ในคุณสมบัติcallback ของวัตถุนั้น เช่นถ้าเราต้องการส่งผ่านตัวแปร gx1 และ gx2 ไปสู่ subfunction เพิ่มเติมให้เราแก้ไขคำสั่ง function ใน M-file เป็น

MYGUI('pushbutton1_callback',gcbo,[],guidata(gcbo),gx1,gx2)

กำหนดให้ GUI ทำงานที่ละครั้งตัวเลือกนี้เป็นการกำหนดตัวเลือกว่าจะให้ GUI นั้นทำงานอย่างไร

- ยอมให้ MATLAB แสดง GUI นี้เพียงที่ละหนึ่งหน้าต่าง ในเวลาหนึ่ง ๆ
- ยอมให้ MATLAB แสดง GUI นี้ได้หลายหน้าต่างพร้อมกันในเวลาหนึ่ง ๆ

ถ้าเราเลือกให้ MATLAB ใช้ GUI นี้ได้เพียงหน้าต่างเดียวในแต่ละเวลาหนึ่ง ๆ จะทำให้ MATLAB เลือกใช้ GUI รูปเดิมขึ้นมาแสดงเมื่อมีการเรียกใช้ GUI นี้ซ้ำแทนที่จะสร้างขึ้นมาใหม่ แต่ถ้าเราไม่เลือกตัวเลือกนี้ MATLAB จะสร้าง GUI ตัวใหม่ขึ้นมาทุกครั้งที่มีการเรียกคำสั่งใช้ GUI นี้

สำหรับ code ใน application M-file ที่กำหนดตัวเลือกนี้จะอยู่ด้านบนของ application M-file โดยใช้คำสั่ง

fig = openfig(mfilename,'reuse')

กรณีที่เลือกตัวเลือกนี้และ

fig = openfig(mfilename,'new')

สำหรับกรณีที่เราไม่เลือกตัวเลือกนี้

Note ให้แน่ใจว่าเรามีการสั่ง openfig ใน application M-file รวมถึงใน command line เพียงครั้งเดียว

2. ใช้สีพื้นที่กำหนดด้วย System ที่ MATLAB ทำงานอยู่

สีที่ใช้ในระบบและส่วนประกอบของ GUI จะเปลี่ยนไปตามระบบคอมพิวเตอร์ที่ใช้ ตัวเลือกนี้ยอมให้เราใช้สีพื้นของ uicontrol เป็นสีเดียวกับสีพื้นของ figure ซึ่งจะทำให้ GUI ของเราดูมีความกลมกลืนเข้ากับสีพื้น แต่หากเราต้องการปรับเปลี่ยนสีพื้นของ uicontrol ที่ใช้ให้เป็นไป

ตามต้องการ เราก็ไม่ต้องเลือกตัวเลือกนี้สำหรับใน application M-file จะมีคำสั่งในการเลือกตัวเลือกนี้คือ

%Use system color scheme for figure

set(fig,'color',get(0,'Default UIcontrol Background Color'));

Note ขอให้แน่ใจว่ามีคำสั่งนี้ใน M-file รวมถึงใน command line เพียงครั้งเดียว

3. การให้รอ input ของผู้ใช้

สำหรับตัวเลือกใน GUIDE application option

Function does not return until application window dismissed

เป็นการสร้าง application M-file ที่ออกแบบให้รอ input จากผู้ใช้ ซึ่งสามารถทำได้โดยใช้ฟังก์ชันuiwait ซึ่งจะป้องกันการทำงานต่อไปของ M-file

ในขณะการทำงานนี้ถูกสั่งให้รอ MATLAB จะจัดคิวให้กับคำสั่งต่าง ๆ ที่เราสั่งเข้าไปไว้ตามลำดับแต่ยังไม่ให้เกิดการทำงานตามคำสั่งนั้น จนกว่าจะเกิดปรากฏการณ์ต่อไปนี้เกิดขึ้น

- รูป GUI ถูกลบ
- ใน GUI มี callback ที่ให้คำสั่ง uiresume

คำสั่งนี้มีประโยชน์ที่จะป้องกัน MATLAB ที่จะใช้คำสั่งจาก command line จนกว่ามีการตอบสนองต่อ dialog box แต่ในขณะเดียวกันยอมให้ callback ทำงานได้

ใน application M-file จะมีการสร้าง code ดังนี้

%wait for callbacks to run and window to be dismissed

uiwait(fig);

เมื่อ fig เป็น handle ของ figure นี้

4. การตั้งชื่อไฟล์และ Tag

ในการกำหนดชื่อของไฟล์ หรือวัตถุต่าง ๆ ที่ใช้ใน GUI ซึ่งจะตั้งชื่อโดยคุณสมบัติ Tag สำหรับGUIDE กำหนดค่าคุณสมบัติ Tag (หรือกำหนดชื่อของวัตถุนั้น) ให้กับวัตถุทุกแบบที่สร้างขึ้นโดยอัตโนมัติ เช่น pushbutton1 และให้ string นี้ จะนำไปใช้เป็นชื่อ callback sub function เช่น pushbutton1_callback อย่างไรก็ดี เพื่อให้ชื่อของวัตถุบ่งบอกถึงหน้าที่ของมันมากขึ้น เราอาจจะตั้งชื่อของวัตถุนั้นให้สื่อถึงหน้าที่ของมันมากขึ้น ดังนั้นเราแนะนำว่า หลังจากที่เราสร้างวัตถุนั้นขึ้นมา

แป้ว เราควรจะตั้งชื่อให้มันด้วย การตั้งชื่อของมันก็คือ การกำหนด คุณสมบัติ Tag ของมันนั่นเอง และเราควรจะทำก่อนที่จะ active หรือ save GUI นี้ด้วย

การใช้ save as จะทำให้ GUIDE ได้สร้าง application M-file ขึ้นมาใหม่ และปรับค่าคุณสมบัติcallback ให้เหมาะสมกับ callbacks ที่มีอยู่ด้วย

Note เนื่องจาก GUIDE ใช้คุณสมบัติ Tag เพื่อสร้างเป็น function และตัวแปรใน structure file ดังนั้น ชื่อ Tagที่เราเลือกจะต้องเป็นตัวแปรที่ใช้ได้ตามข้อกำหนดของ MATLAB

การเปลี่ยนชื่อคุณสมบัติ Tag เราควรจะมีการปรับเปลี่ยนก่อนจะ Activate หรือ save รูป GUI และสร้าง Application M-file เพื่อป้องกันการสับสน อย่างไรก็ตามหากว่าเราทำการปรับเปลี่ยน Tag ของคุณสมบัติใด ๆ หลังจากเคยสร้าง application M-file ขึ้นมาแล้ว อาจจะมีปัญหาบางประการเกิดขึ้น เพราะมีบางส่วนที่ GUIDE จะไม่เปลี่ยนแปลงชื่อใน application M-file ให้โดยอัตโนมัติ ทำให้เราต้องตามเข้าไปแก้ไขใน application M-file เอง

ถ้าเราเปลี่ยน Tag หลังจากสร้าง application M-file GUIDE จะไม่สร้าง sub function ใหม่ให อย่างไรก็ตาม เนื่องจาก handles นั้นจะสร้างขึ้นในเวลาที่ MATLAB ทำงาน ดังนั้น GUIDE จะใช้ชื่อ Tag ใหม่ในการสร้าง file ในตัวแปร handles ดังนั้นถ้าเดิมใน application M-file ที่ใช้คำสั่ง

x = get(handles,listbox1,'string')

ถ้าเราเปลี่ยน Tag จาก Listbox1 เป็น graph เราจะต้องเปลี่ยนคำสั่งใหม่เป็น

x = get(handles,graph,'string')

เพราะในการทำงานใหม่ของ GUI จะไม่มี file ใน structure handles ที่ชื่อ listbox1 อีกต่อไปแล้วและจะเกิด error ขึ้นเมื่อเราสั่ง MATLAB ทำงาน ถ้าไม่เปลี่ยนคำสั่งตามที่กล่าวไว้ ดังนั้นของแนะนำให้มีการเปลี่ยนชื่อที่สร้างโดยอัตโนมัติ หรือที่เราสร้างเป็นคำสั่งในภายหลังให้เหมาะสม

ข้อสำคัญอีกประการหนึ่งคือการสร้าง FIG-file และ M-file และ M-file ที่จะนำมาใช้งานร่วมกันGUIDE ก็จะสร้างชื่อ file เหมือนกัน แต่มี extension ต่างกัน (fig และ M) แต่ถ้าเราเปลี่ยนชื่อ file ใด fileหนึ่ง การทำงานจะ

GUI Building Tools (GUI Layout Tool)

MATLAB จะมีเครื่องมือในการช่วยสร้าง GUI อยู่หลายส่วนโดยจะเริ่มต้นจาก GUI Layout tools ซึ่งถือว่าเป็นขั้นตอนแรกในการสร้าง GUI เพราะจะเป็นการกำหนดว่าใน GUI นี้จะมี uicontrol และชื่อaxes อะไร อยู่ใน GUI อะไรบ้าง และแต่ละตัวจะมีตำแหน่งอยู่ที่ใด ขนาดเท่าไร รูปแบบ สี เป็นอย่างไรซึ่งเครื่องมือนี้จะประกอบด้วย

• Layout Editor – เพิ่มและจัดวัตถุต่าง ๆ ใน GUI

• Alignment Tool – จัดวางวัตถุเทียบกับวัตถุอื่น ๆ ใน GUI ให้เป็นระเบียบมากยิ่งขึ้น

• Property Inspector – ตรวจสอบและตั้งค่าคุณสมบัติต่าง ๆ ของวัตถุแต่ละอัน

• Object Browser – ตรวจสอบและแสดงลำดับขั้นของวัตถุที่มี handle ใน MALAB ขณะนั้น

• Menu Editor – สร้างเมนูของหน้าต่าง และ context menu

เราจะเข้าสู่เครื่องมือต่าง ๆ นี้ได้โดยผ่านเข้าทาง GUIDE Layout Editor ในการเริ่มการทำงานของLayout Editor ให้ใช้คำสั่ง

»guide

จากนั้น MATLAB จะแสดง GUI ใหม่ขึ้นมาโดยยังไม่มีวัตถุวางอยู่ใน GUI นั้น หรือถ้าเราสั่ง

»guide mygui.fig

(จะมี .fig หรือไม่ก็ได้) จะเป็นการเรียก GUI เก่าขึ้นมาเพื่อแก้ไข หรือถาต้องการแก้ไข GUI เก่าเราอาจใช้คำสั่ง open ภายใต้เมนู File ของ Layout Editor ก็ได้

5. การวางส่วนประกอบต่างๆ ลงใน GUI โดย Layout Editor

การใช้ Layout Editor จะช่วยให้เราสามารถกำหนดส่วนประกอบต่าง ๆ ว่าจะมีอะไร และมีตำแหน่งอยู่ที่ใด การวางตำแหน่งวัตถุต่าง ๆ ซึ่งก็จะมี uicontrol และ axes ก็จะเหมือนการใช้โปรแกรมวาดรูปทั่ว ๆ ไป มีขั้นตอนดังนี้คือ

5.1 เลือก uicontrol หรือ axes ที่ต้องการจะเพิ่มไปใน GUI จาก component palette

5.2 เลื่อนเมาส์เข้ามาในบริเวณพื้นที่ของ GUI ลักษณะ cursor จะเปลี่ยนเป็นรูปกากบาทซึ่งเราจะสามารถใช้กำหนดตำแหน่ง มุมซ้ายบน ของวัตถุนั้นได้ โดยการกดเมาส์ปุ่มซ้ายที่ตำแหน่งที่เราต้องการ แล้วลากเมาส์ขณะกดปุ่มเมาส์ด้านซ้ายอยู่ เพื่อกำหนดตำแหน่งด้านขวาล่างของวัตถุเมื่อให้ตำแหน่งที่ต้องการให้เราปล่อยปุ่มเมาส์

5.3 เราสามารถปรับปรุงขนาดและเลื่อนตำแหน่งของวัตถุนั้นได้ โดยใช้เมาส์เลือกวัตถุนั้นแล้วเลื่อนหรือปรับขนาดได้ตามต้องการ

6. Activating the Figure

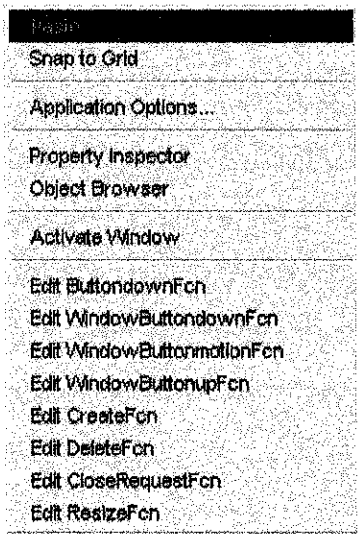
เราสามารถสร้างการทำงานของ GUI ได้โดยสั่ง activate figure ที่เราได้ออกแบบมาแล้วด้วยLayout Editor เราสามารถ Activate รูปได้โดยเลือก activate figure ภายใต้เมนู Tool หรือโดยการกด Activatorbutton บน Toolbarเมื่อเราสั่ง activate figure สิ่งต่อไปนี้จะเกิดขึ้นก่อนอื่น GUIDE

จะทำการ SAVE ไฟล์ทั้ง M-file และ Fig-file เป็นอันดับแรก ถ้า file ทั้งสองไม่เคยถูกSAVE มาก่อน ก็จะมี Dialog box SAVE AS เกิดขึ้น เพื่อถามชื่อ FILE ที่เราต้องการ SAVE ถ้าเราใช้ชื่อไฟล์ที่มีอยู่แล้ว MATLAB จะถามว่าเราต้องการเขียนทับ เขียนเพิ่ม (Append) หรือยกเลิกการใช้ชื่อนั้น

Note เพื่อให้เราสามารถสร้าง GUI ที่มีประสิทธิภาพและควบคุมการทำงานได้อย่างสะดวก เราควรตั้งชื่อTag ของวัตถุต่างๆ ใน GUI ให้เรียบร้อยก่อนจะมีการ Activate Figure เพราะการแก้ไขชื่อนี้ภายหลังจะมีปัญหายุ่งยากตามมาได้

Layout Editor Context Menu

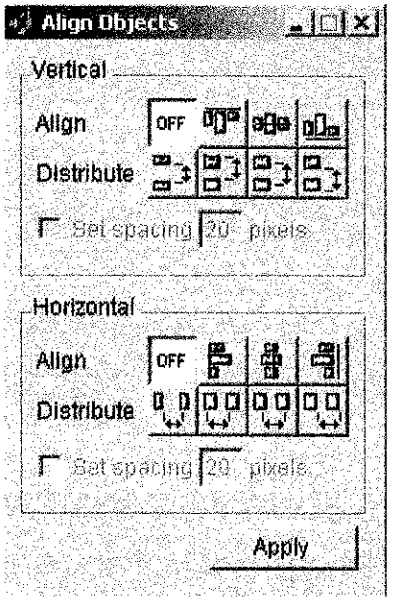
เมื่อเราทำงานภายใต้ Layout Editor เราสามารถเลือกวัตถุนั้นด้วยเมาส์ปุ่มซ้าย และเมื่อเรากดเมาส์ปุ่มขวาเหนือวัตถุนั้น ก็จะปรากฏ context menu ขึ้น นอกเหนือจากที่เราจะใช้เมนูที่ปรากฏอยู่ด้านบนของหน้าต่าง ซึ่งเราสามารถใช้ context menu นี้ สร้าง subfunction ให้กับ application M-file ของเราได้ สำหรับทุกวัตถุชนิดที่มี callback routine ในรูปข้างล่างนี้แสดง เมนู Context ของ figure



รูปที่ 2.13 เมนู Context ของ figure

6. Aligning Component in The Layout Editor

ในการจัดเรียงส่วนประกอบต่างๆ ที่มีอยู่ แม้ว่าสามารถที่จะใช้เมาส์เลื่อนวัตถุต่างๆ ได้อยู่แล้วแต่การจัดเรียงส่วนประกอต่างๆ ให้วางอยู่ในแนวเดียวกัน มีระยะห่างเท่าๆ กันนั้น จะมีความสะดวกขึ้นหากเราใช้ Alignment Tool เราสามารถเลือก Alignment Tool ได้โดยเลือก Alignment Tool จากปุ่มบนเมนูซึ่งAlignment Tool จะมีลักษณะดังนี้

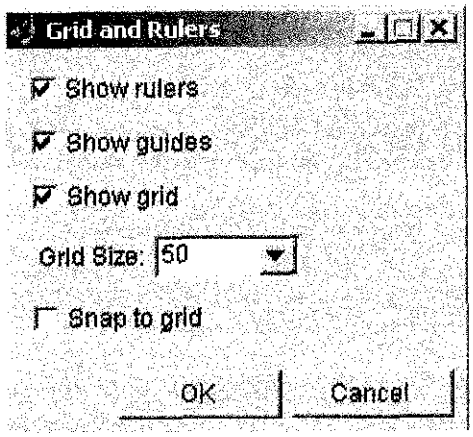


รูปที่ 2.14 หน้าต่างของฟังก์ชัน Alignment Tool

ก่อนที่เราจะใช้ Alignment Tool เราต้องเลือกกลุ่มวัตถุที่จะจัดเรียงเสียก่อน ซึ่งเราสามารถทำได้โดย

• เลือกลูกศร (select) จาก component palette แล้ว กำหนดพื้นที่กรอบสี่เหลี่ยมที่บรรจุวัตถุทั้งหมดที่ต้องการ Align เมื่อปล่อยเมาส์ วัตถุเหล่านั้นจะถูกเลือก

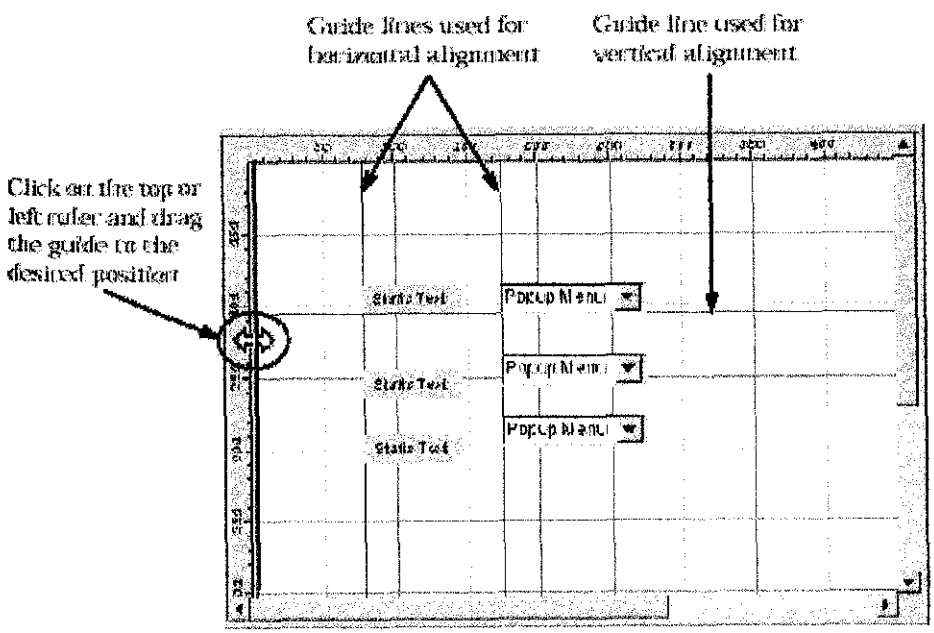
• เลือกวัตถุที่ละอัน โดยกดแป้น Shift บนแป้นพิมพ์ค้างไว้ แล้วเลือกวัตถุที่ต้องการไปเรื่อยๆเมื่อเลือกกรอบแล้ว จึ้งปล่อยแป้นพิมพ์หลังจากที่เราเลือกวัตถุครบถ้วนแล้ว เราจึงเลือกวิธีการจัดเรียงวัตถุเหล่านั้นว่า เราต้องการให้จัดเรียงอย่างไร ทั้งในแนว Vertical และ Horizontal เมื่อเลือกลักษณะการจัดเรียงเรียบร้อยแล้วจึงกดปุ่มApply เพื่อจัดแนวและตั้งระยะห่างวัตถุทั้งในแนวตั้งและแนวนอนให้เป็นไปตามที่เราต้องการนอกเหนือจากการใช้ Alignment Tool เพื่อจัดเรียงวัตถุนั้นแล้วเรายังสามารถใช้ Grids และ Rulersเพื่อช่วยในการจัดเรียงโดย grid ที่สร้างขึ้นนั้นสามารถปรับปรุงได้โดยเรียก Grid and Rulers ภายใต้เมนูLayout



รูปที่ 2.15 ฟังก์ชัน Grid and Rulers ภายใต้เมนูLayout

โดยเราสามารถกำหนด Grid size ได้ระหว่าง 10 57' 200 Pixel โดยค่า 50 เป็น default นอกจากนี้เรายังมีตัวเลือก Snap-to-grid เพื่อกำหนดให้วัตถุที่มีการเคลื่อนที่หรือปรับขนาดที่อยู่ในระยะ 9 pixels ของเส้น grid จะเคลื่อนที่เข้าหาเส้น grids การเลือก snap-to-grid นี้จะทำงานทั้งที่เราแสดงหรือไม่แสดงเส้นgrid บน Layout Editor

นอกเหนือจากนั้นเรายังสามารถสร้าง guide line ขึ้นมาเพื่อสะดวกในการกำหนดตำแหน่ง การสร้าง guide line นี้ทำได้โดยใช้เมาส์ปุ่มซ้ายกดที่ ruler ด้านบนหรือด้านซ้ายมือ จากนั้นดึงเส้นเข้ามาภายในพื้นที่ของ gui จะมีเส้นตรงตามเมาส์เข้ามาได้ เมื่อเราปล่อยเมาส์ เส้นตรงใหม่ก็จะกลายเป็นเส้นgrid เส้นใหม่แต่แสดงสีที่แตกต่างออกไป การสร้าง grid แสดงในรูป



รูปที่ 2.16 หน้าต่างแสดงการสร้าง Grid size

และการจัดเรียงวัตถุใน GUI ที่จะกล่าวถึง เป็นแบบสุดท้ายในที่นี้คือการจัดเรียงลำดับการวางทับกันบน GUI ซึ่งปกติวัตถุที่สร้างที่หลังจะวางอยู่ด้านบนวัตถุที่สร้างก่อน แต่เราสามารถปรับลำดับได้โดยกดเมาส์ปุ่มขวาเมื่อเลือก context menu แล้วเลือก

Bring to Front, Send to back, Bring Forward หรือ **send Backward**

ตามต้องการ

2.5.4 การกำหนดค่าคุณสมบัติของส่วนประกอบต่างๆ

เราสามารถที่จะกำหนดค่าคุณสมบัติของส่วนประกอบต่างๆใน GUI ได้ด้วยการใช้ PropertyInspector ซึ่งจะให้รายการคุณสมบัติทั้งหมดของวัตถุที่เราเลือกและแสดงค่าปัจจุบันของคุณสมบัติเหล่านั้น สำหรับคุณสมบัติแต่และตัวในรายการนั้น จะมีอุปกรณ์ที่ใช้ในการแก้ไขคุณสมบัติแต่ละตัวไว้ด้วย คุณสมบัติบางตัวซึ่งมีตัวเลือกอุปกรณ์แก้ไขก็จะแสดงตัวเลือกไว้ให้ ส่วนคุณสมบัติบางตัวต้องเป็นการกำหนดค่า ก็จะเป็นการกำหนดค่าลงไปการที่เราจะให้ Property Inspector ปรากฏขึ้นเราสามารถทำได้หลายวิธีคือ

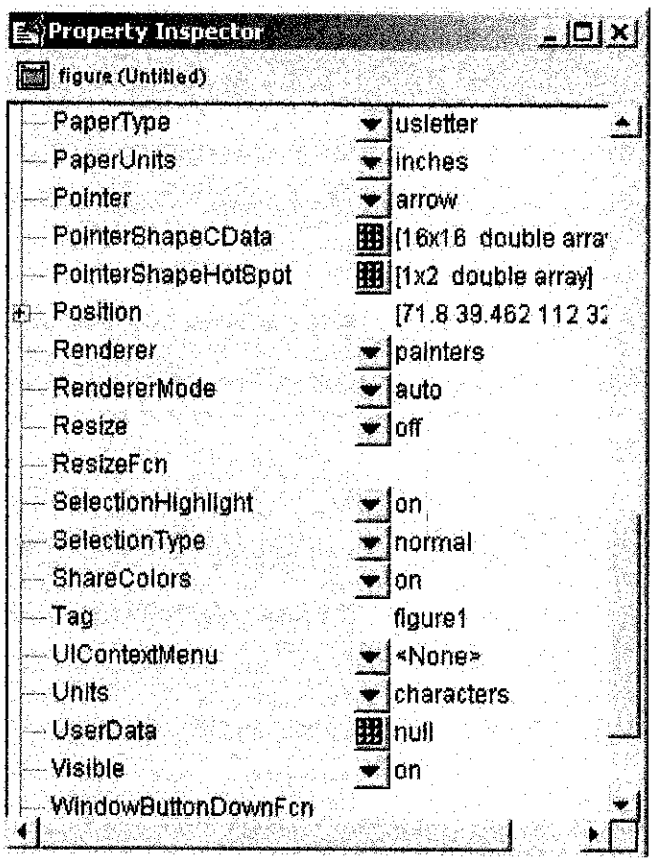
- กดเมาส์ปุ่มซ้ายสองครั้ง ส่วนประกอบที่ต้องการแสดงคุณสมบัติ
- เลือก Property Inspector ภายใต้เมนู Tools
- เลือก Inspect property ภายใต้เมนู Edit
- กดเมาส์ปุ่มขวาบนวัตถุนั้น แล้วเลือก Inspect Properties จากเมนู context
- กดเมาส์ปุ่มซ้ายที่ Property Inspector ที่ Toolbar

และ Property Inspector จะแสดงคุณสมบัติของวัตถุที่เราเลือกบน Layout Editor เมื่อเราเปลี่ยนวัตถุที่เลือกไป Property ที่แสดงก็จะเปลี่ยนไปตามวัตถุนั้นด้วย

เมื่อเราตรวจคุณสมบัติเหล่านั้น เราก็สามารถจะปรับแก้คุณสมบัติต่างๆ ได้ตามต้องการ สำหรับคุณสมบัติที่มีเครื่องหมาย อยู่ด้านหน้าชื่อคุณสมบัติ หมายความว่าเราสามารถขยายคุณสมบัติเหล่านั้นได้ เพื่อปรับแก้คุณสมบัติย่อยแต่ละตัวอย่างอิสระ

ในกรณีที่เราเลือกวัตถุหลายวัตถุพร้อมกัน Property Inspector จะแสดงคุณสมบัติที่วัตถุนั้นมีร่วมกัน ส่วนค่าที่แสดงนั้นหากวัตถุแต่ละชิ้นมีค่าไม่เท่ากัน ค่าที่แสดงจะปรากฏเป็น **Mixed** ขึ้น หมายความว่าเป็นค่ารวมหลายๆ ค่าอยู่ โดยแต่ละวัตถุนี้มีคุณสมบัตินี้ไม่เท่ากัน ถ้าเราปรับเปลี่ยนค่า

ดังกล่าว คุณสมบัติของวัตถุทุกตัวที่เราเลือกก็จะเปลี่ยนไปมีค่าเท่ากัน ซึ่งจะมีประโยชน์ในการกำหนดขนาด สี แบบตัวอักษรของวัตถุหลายๆชนิด ที่เราต้องการให้มีคุณสมบัติบางอย่างเหมือนกันในการกำหนดครั้งเดียวแทนที่จะปรับแก้ทีละตัว



รูปที่ 2.17 Property Inspector ใช้แสดงคุณสมบัติของวัตถุ

2.5.5 User Interface Controls

สำหรับ User interface control นั้นจะประกอบด้วย 1) Check Boxes, 2) Editable Text, 3) Frames 4)List boxes, 5) Push Buttons, 6) Radio Buttons, 7) Sliders, 8) Static Text, 9) Toggle Buttonsซึ่งรายละเอียดของคุณสมบัติเหล่านี้ เราได้กล่าวถึงไปแล้วในบทของ object Properties ที่ผ่านมาดังนั้นเราจะไม่ขอกล่าวถึงคุณสมบัติ และวิธีการใช้ของมันอีกในบทนี้ เพียงแต่ใน GUIDE ของ MATLAB6.0 นี้มีการสร้าง Callback subfunction ให้กับวัตถุต่างๆเหล่านี้ (ยกเว้น frame และ Static text) โดยอัตโนมัติซึ่งต้องใช้ชื่อ Tag เป็นองค์ประกอบในชื่อ subfunction นั้นด้วย ดังนั้นเราจะขอแสดงชื่อที่ DUIDE ตั้งให้Control แต่ละตัว เป็น default ให้กับวัตถุเหล่านี้ อย่างไรก็ตามอย่าลืมว่าเราควรจะกำหนดชื่อ Tag เหล่านี้ใหม่ให้เหมาะสม กับการทำงานของมัน

2.5.6 Understanding the Application M-File

Application M-file เป็นโครงร่างโปรแกรมที่ใช้ควบคุมการทำงานของ GUI ซึ่งจะถูกสร้างขึ้นโดยอัตโนมัติพร้อมกับ Fig-file เมื่อใช้ GUIDE ในการสร้าง GUI โดยที่ Application M-file จะช่วยให้ มีความสะดวก และรวดเร็วมากขึ้น ซึ่งโปรแกรมหรือ Code ทุกส่วนรวมถึง Subfunction จะรวมอยู่ใน ApplicationM-file โดย Callback ทั้งหมดจะถูกเขียนเป็น Subfunction ซึ่งทำให้วิธีการเขียน Callback ง่ายขึ้นและทำให้สามารถปรับค่าเริ่มต้น หรือทำการ Initialize ให้กับ GUI ของเราได้ หัวข้อที่จะกล่าวต่อไปนี้จะเป็นประโยชน์กับผู้ที่จะทำการเขียน M-file ที่ใช้ควบคุม GUI ไม่ว่าจะสร้างขึ้นเอง หรือจะเป็น application Mfileที่สร้างขึ้นโดย GUIDE

อย่างไรก็ตาม ในเอกสารนี้จะเน้นถึงการพิจารณา Application M-file ที่สร้างขึ้นโดย GUIDE เป็นหลัก และเมื่อผู้อ่านเข้าใจถึงการเขียนคำสั่งต่างๆ ที่ถูกสร้างขึ้นมาแล้วก็จะสามารถสร้าง M-file ขึ้นมาควบคุมการทำงานของ GUI ได้ด้วยตนเอง ส่วนประกอบสำคัญ Application M-file มีดังต่อไปนี้

- การสร้างและกำหนดชื่อ Callback โดยอัตโนมัติ
- ขั้นตอนการทำงานของ Application M-file
- การกำหนดค่าเริ่มต้นใน Application M-file

1. Automatic Naming of Callback Routines

GUIDE จะทำการสร้างและตั้งชื่อ Callback subfunction ขึ้นมาโดยอัตโนมัติ ให้กับวัตถุต่างๆ ที่สร้างขึ้นมาใน Layout Editor ยกเว้น Static Text และ Frame ดังนั้นเมื่อมีการเลือกวัตถุเหล่านั้นใน GUI จะทำให้ MATLAB มองไปที่ Callback ซึ่งจะชี้ไปที่ Callback subfunction ของแต่ละตัวใน application M-file แล้ว

MATLAB จะทำงานตามที่กำหนดใน Subfunction ดังนั้นขั้นตอนการทำงานของ GUIDE และของ GUI ที่สร้างโดย GUIDE จะมีขั้นตอนโดยสรุปดังนี้

1.1 สร้างรูปแบบ GUI ใน GUIDE Layout Editor และปรับแก้คุณสมบัติต่างๆ ให้เป็นตามต้องการ

2.2 เมื่อทำการ Save GUI ใน Layout Editor แล้ว GUIE ก็จะทำขั้นตอนต่อไปนี้

- สร้าง Application M-file และ Fig-file โดยใช้ชื่อเดียวกันกับชื่อ GUI
- ทำการกำหนดค่า Callback ของวัตถุต่างๆ ให้เป็น

my_gui('ObjectTag_Callback',gcbo,[],guidata(gcbo))

โดย

- **my_gui** จะเป็นชื่อของ application M-file ที่เรากำหนดในตอน save

- **ObjectTag_Callback** จะเป็นชื่อของ Subfunction ที่จะเกิดขึ้นใน Application M-file โดยชื่อObject Tag นั้นจะเปลี่ยนไปตามที่เรากำหนดคุณสมบัติ Tag ของวัตถุนั้นเช่น ถ้าเรากำหนดให้วัตถุนั้นมี Tag ชื่อ pushbutton1 ก็จะทำให้ชื่อของ subfunction นี้มีชื่อเป็น pushbutton1_Callback เป็นต้นซึ่งจะเป็นการสะดวกที่เราจะทราบว่า subfunction ใดใน application M-file เป็นของวัตถุใดใน GUIนั่นเอง

- **gcbo** มาจาก get callback object นั่นคือจะให้ค่าเป็น handle ของวัตถุที่กำลังเรียก Callback จากผู้ใช้

- [ ] เป็นเมทริกส์ว่าง ที่เก็บสำรองไว้ ซึ่งอาจมีการใช้ในอนาคต

- **guidata(gcbo)** เป็นข้อมูลของ handle ของวัตถุทุกตัวที่อยู่ใน GUI ซึ่งจะทำให้เราเรียกหาวัตถุต่างๆที่มีอยู่ใน GUI นั้นได้สะดวกยิ่งขึ้นเมื่อมีการสร้างวัตถุใดๆ ลงใน GUI แล้ว GUIDE ก็จะทำการสร้าง Subfunction ที่ชื่อObjectTag_Callback ขึ้นมาให้ด้วย ดังนั้นเมื่อมีการเรียกวัตถุนั้นใน GUI ก็จะทำให้ MATLAB ไปที่Application M-file ตามที่เรากำหนด (ในตัวอย่างข้างบนก็จะไปที่ฟังก์ชันไฟล์ my_gui.m) และมองหาSubfunction ที่ชื่อ ObjectTag_CallBack โดยใน application M-file ก็จะปรากฏ Subfuction ในลักษณะต่อไปนี้ขึ้นมา

```
% --------------------------------------------------------------------
function varargout = ObjectTag_Callback(h, eventdata, handles, varargin)
% Stub for Callback of the uicontrol handles.ObjectTag.
disp('ObjectTag Callback not implemented yet.')
```

ซึ่งสำหรับ input ที่กำหนดให้กับ Subfunction นี้จะมี

h เป็น handle ของวัตถุนี้

eventdata เป็นเมทริกส์ว่างซึ่งสำรองไว้ใช้ในอนาคต

handles จะเป็นตัวแปรประเภท structure โดยจะมี field ต่างๆ เป็น handle ของวัตถุที่มีใน GUI และชื่อของ field จะมีชื่อตาม Tag ของวัตถุนั้น ยกตัวอย่างเช่นถ้า GUI นั้นประกอบด้วยวัตถุอยู่ 3 วัตถุคือ

1. หน้าต่างรูปภาพหรือ Figure ซึ่งมีชื่อ Tag เป็น figure1 และ handle มีค่าเป็น1.00256
2. ปุ่ม Push Buttom ซึ่งมีชื่อ Tag เป็น ObjectTag และ handle มีค่าเป็น 5.00123
3. มี Slider ซึ่งมีชื่อ Tag เป็น slider1 และ handle มีค่าเป็น 6.00301

ก็จะทำให้ตัวแปร handles ซึ่งเป็นตัวแปรแบบ Structure นี้มี 3 field คือ

1. handles.figure1 มีค่า 1.00256

2. handles.ObjectTag มีค่า 5.00123

3. handles.slider1 มีค่า 6.00301

ดังนั้นจะทำให้เราทราบ handle ของวัตถุทั้งหมดที่มีอยู่ใน GUI จึงทำให้ไม่จำเป็นต้องใช้คำสั่ง findobj เพื่อหาวัตถุเหล่านั้นเหมือนในการเขียน GUI ใน version ก่อนๆ ของ MATLAB

varargin เป็นตัวแปรที่ส่งผ่านเข้าไปสู่ฟังก์ชัน (variable argument input)ส่วนคำสั่ง

disp('ObjectTag Callback not implemented yet.')

มีไว้เพื่อเมื่อเรากดที่วัตถุนั้นแล้ว บน Command Window จะแสดงข้อความว่าวัตถุนี้ยังไม่ได้มีการปรับเปลี่ยน Callback และจะทำงานยังไม่ได้ เมื่อเราเข้ามาที่ Subfuction นี้ ก็ให้เราลบบรรทัดนี้ออกแล้วเขียนชุดคำสั่งตามที่เราต้องการ

2. ขั้นตอนการทำงานและกำหนดค่าเริ่มต้นของ Application –M file

ขั้นตอนการทำงานของ Application M-file จะเกิดขึ้นเมื่อผู้ใช้สั่งชื่อ M-file นี้จาก CommandWindows ซึ่งผู้ใช้จะสั่งการได้สองแบบคือ

2.1 เรียก M-file โดยไม่มีข้อกำหนดใดเพิ่มเติม

2.2 เรียก M-file โดยมีข้อกำหนดเพิ่มเติม ซึ่งโดยส่วนใหญ่แล้วก็จะเป็นชื่อของ Subfunction และมักจะเป็น callback

ซึ่งทำให้ GUIDE ต้องทำส่วนที่เรียกว่า Switchyard Code ขึ้นมา เพื่อให้การทำงานของ GUI สามารถเกิดขึ้นได้ด้วยวิธีการเรียกใช้ทั้งสองแบบของการเรียกใช้ GUI

The Switchyard Code

เป็นส่วนของโปรแกรมที่เขียนขึ้นมาเพื่อตรวจสอบว่าผู้ใช้ได้เรียกใช้คำสั่งกับ GUI ของเราอย่างไร และจะกำหนดให้เกิดการทำงานตามที่ผู้ใช้ต้องการ ซึ่ง Switchyard Code นี้จะเป็นโปรแกรมที่ถูกGUIDE สร้างขึ้นอยู่ในส่วนต้น ของโปรแกรม และทำหน้าที่กำหนดสภาพเริ่มต้นการทำงานของโปรแกรมด้วยอีกหน้าที่หนึ่ง โดยมีลักษณะดังนี้

```
if nargin == 0 % 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);
        if nargout > 0
        varargout{1} = fig;
        end
elseif ischar(varargin{1}) % INVOKE NAMED SUBFUNCTION OR CALLBACK
        try
        [varargout{1:nargout}]=feval(varargin{:});%FEVAL switchyard
        catch
        disp(lasterr);
        end
end
```

โดยเราจะขออธิบายถึงส่วนต่างๆ ใน code บางส่วนดังนี้

if nargin == 0 % LAUNCH GUI เริ่มจากคำสั่ง if ซึ่งขั้นแรกจะเป็นการตรวจสอบว่าคำสั่งที่มาจาก Command line นั้นมีข้อความต่อจากชื่อไฟล์ของ GUI ของเราหรือไม่ ถ้าไม่มีคือเท่ากับ 0 ก็จะทำให้ MATLAB ทำการเปิด GUI ขึ้นทันที โดยเมื่อเปิด GUI จะทำการต่อไปนี้

fig = openfig(filename,'reuse'); เปิดหน้าต่างที่เป็น GUI ของเราขึ้นมา โดยจะเป็นการเปิด FIG-file ที่เก็บไว้นั่นเอง ส่วนข้อกำหนด **reuse** นั้นหมายความว่าจะยอมให้มี GUI นี้เกิดขึ้นได้เพียงหน้าต่างเดียวในเวลาหนึ่งๆ (ซึ่งเป็นไปตามข้อกำหนดที่เราเลือก ซึ่งได้กล่าวถึงไปก่อนหน้านี้แล้ว)แต่ถ้าหากว่าเราเลือกให้มี GUI นี้ได้พร้อมกันหลายหน้าต่าง ข้อกำหนดนี้จะเปลี่ยนเป็น **new**

set(fig,'Color',get(0,'defaultUicontrolBackgroundColor')); คำสั่งนี้จะทำให้ uicontrol หรือวัตถุต่างๆที่ปรากฏอยู่ใน GUI มีสีพื้นเหมือนกับสีพื้นของหน้าต่าง GUI ในส่วนนี้ก็มาจากข้อกำหนดที่เราเลือกในตอนสร้าง GUI ที่กล่าวมาก่อนหน้านี้นั่นเอง

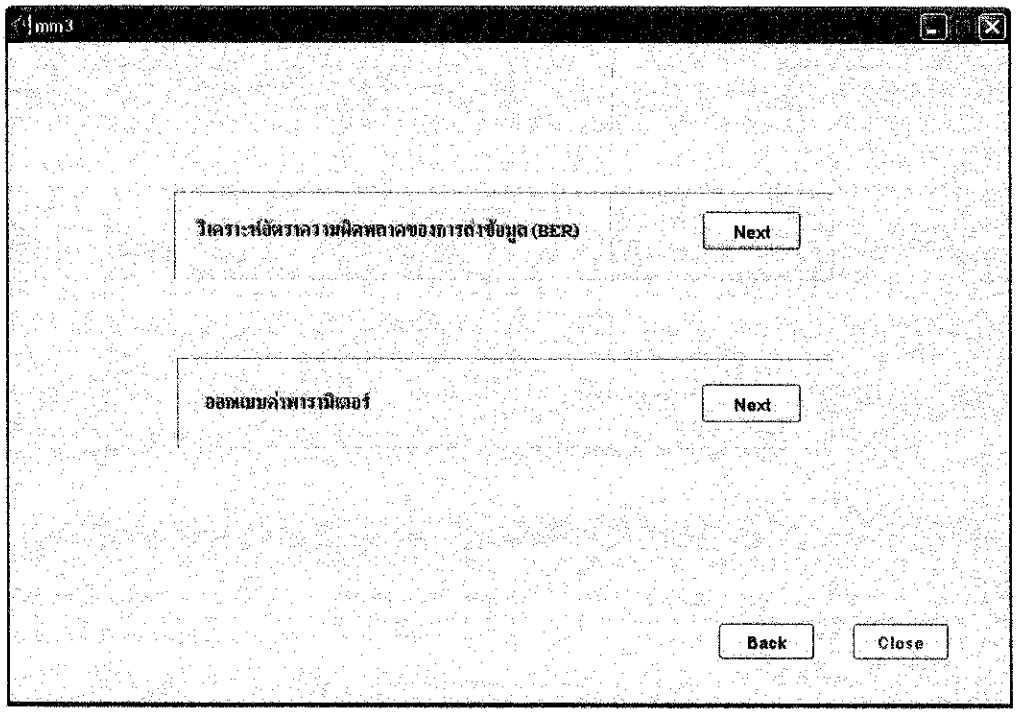
handles = guihandles(fig); คำสั่ง **guihandles** จะสร้างตัวแปรแบบ structure จากรูปภาพที่สร้างขึ้น โดยให้ field ทั้งหมดในตัวแปรนั้นคือวัตถุต่างๆที่มีอยู่ในรูปภาพที่กำหนด ในตัวอย่างนี้จะสร้างตัวแปรที่มีชื่อว่า **handles** สำหรับรูปที่ handle มีค่าเท่ากับ **fig** และชื่อ field นี้ก็จะใช้ชื่อ Tag ของวัตถุต่างๆ ทำให้เราได้ตัวแปรโครงสร้าง handles มีลักษณะตามที่ได้กล่าวมาก่อนหน้านี้แล้ว

guidata(fig, handles) คำสั่งนี้เป็นคำสั่งที่ให้เก็บค่าตัวแปร **handles** เข้าไปเป็นส่วนหนึ่งของรูปที่สร้างขึ้น คำสั่ง **guidata** นี้เป็นคำสั่งที่ใช้เก็บข้อมูล หรือดึงข้อมูลที่มีอยู่จากรูปออกมาการเขียนคำสั่งในลักษณะเช่นนี้จะทำให้ MATLAB ทำการเก็บตัวแปรชื่อ **handles** นี้เข้าไว้ให้อยู่ในรูปซึ่งเราสามารถที่จะเรียกข้อมูลจากรูปออกมาได้ในภายหลัง ส่วนสำคัญของคำสั่งนี้ก็คือหากในภายหลังเราได้มีการเพิ่มเติมข้อมูลสำคัญเข้าไปอยู่ใน GUI และต้องการส่งต่อหรือมีการปรับเปลี่ยนค่าตัวแปร

handles นี้ไม่ว่าด้วยเหตุผลใด เราจะต้องใช้คำสั่งนี้ใหม่อีกครั้งหนึ่ง เพื่อให้เก็บข้อมูลที่ได้รับการปรับแก้แล้วเข้าไปอยู่ในรูปส่วนในชุดคำสั่งที่มี if หรือ elseif ที่อยู่ต่อมานั้นเป็นคำสั่งที่ในกรณีที่ผู้ใช้เรียกใช้ GUI โดยมีข้อกำหนดเพิ่มเติม ซึ่ง MATLAB ก็จะทำการตรวจสอบข้อกำหนดนั้นว่าเป็นอย่างไร และถ้าหากว่าข้อกำหนดนั้นเป็นชื่อของ Subfunction ที่อยู่ในโปรแกรมนี้ ก็จะสั่งให้ Subfunction นี้ทำงานก่อน

บทที่ 3
การทำงานของโปรแกรม

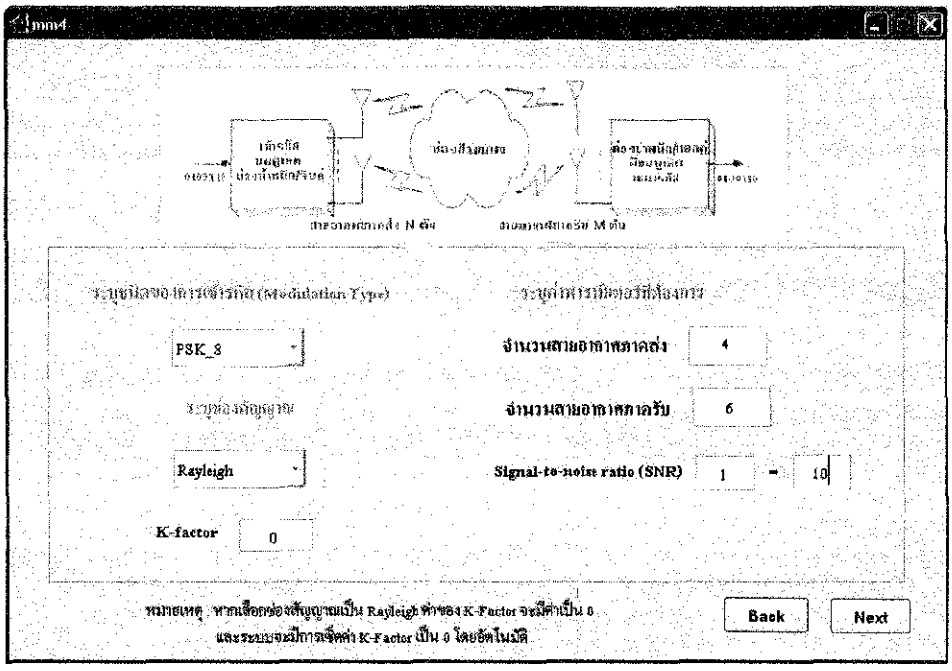
เมื่อเข้าสู่โปรแกรมวิเคราะห์และออกแบบระบบ MIMO ด้วย GUI ผู้ใช้งานจะเจอกับหน้าต่างดังรูปที่ 3.1 ซึ่งเป็นหน้าต่างสำหรับให้ผู้ใช้งานเลือกความต้องการในการใช้งานโปรแกรม โดยโปรแกรมแบ่งการใช้งานออกเป็น2ส่วนคือ 1) วิเคราะห์อัตราการผิดพลาดของข้อมูล(BER) และ 2) การออกแบบพารามิเตอร์



รูปที่ 3.1 หน้าต่างแสดงให้ผู้ใช้เลือกการใช้งานของโปรแกรม

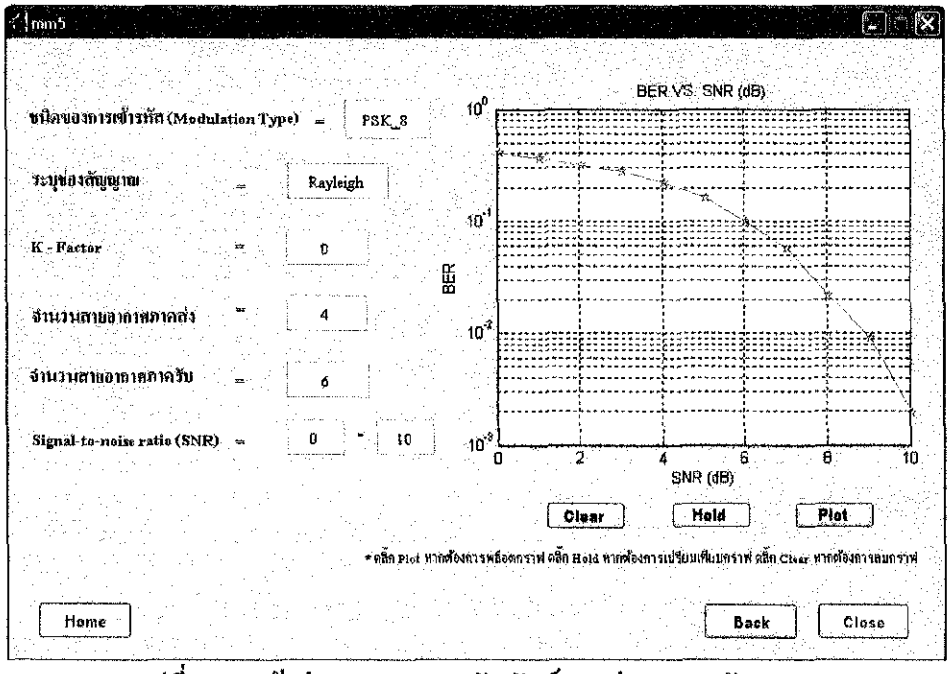
3.1 โปรแกรมการวิเคราะห์อัตราการผิดพลาดของการส่งข้อมูล (BER)

หลังจากที่ผู้ใช้งานเลือกการใช้งานวิเคราะห์อัตราการผิดพลาดของข้อมูล(BER) ก็จะพบกับหน้าต่างดังรูปที่ 3.2 ซึ่งมีรายละเอียดให้ผู้ใช้สามารถป้อนค่าพารามิเตอร์ที่ต้องการเพื่อนำไปประมวลผลในการวิเคราะห์อัตราการผิดพลาดของข้อมูล(BER)



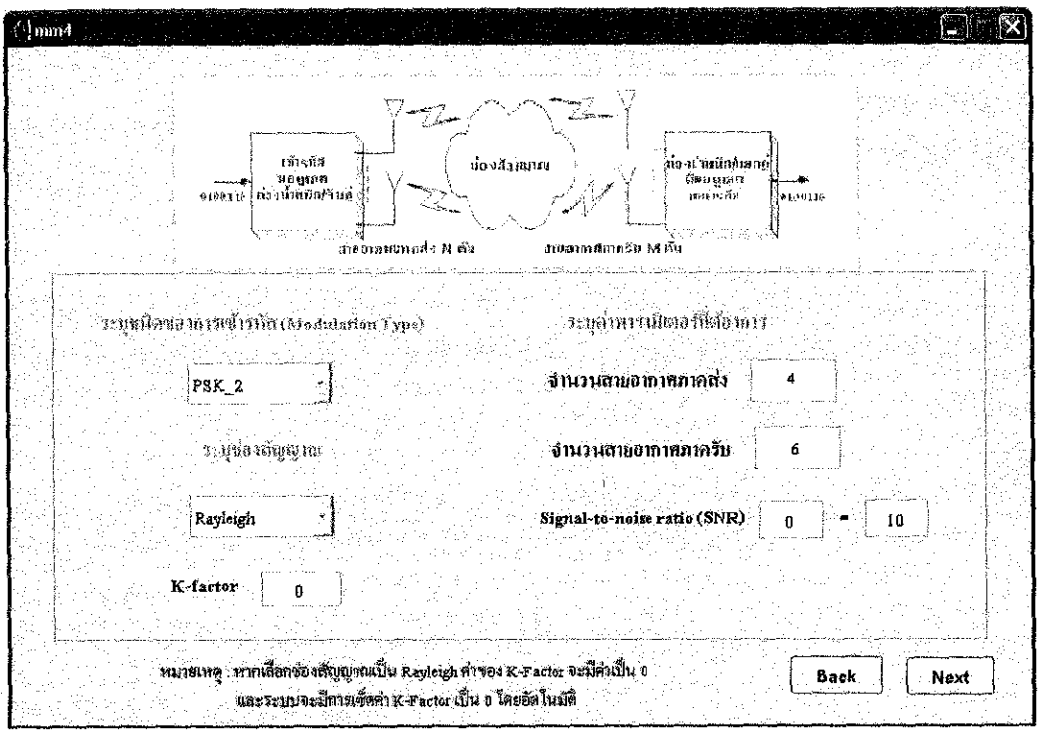
รูปที่ 3.2 หน้าต่างแสดงการรับค่าพารามิเตอร์จากผู้ใช้งาน

เมื่อโปรแกรมรับค่าพารามิเตอร์ที่ผู้ใช้งานป้อนเข้ามาครบทุกตัวแล้ว โปรแกรมจะทำการประมวลผลและแสดงค่าความสัมพันธ์ระหว่างอัตราการผิดพลาดของข้อมูล(BER)กับ Signal-to-Noise Ratio (SNR) ให้อยู่ในรูปของกราฟความสัมพันธ์ดังที่แสดงในรูปที่ 3.3



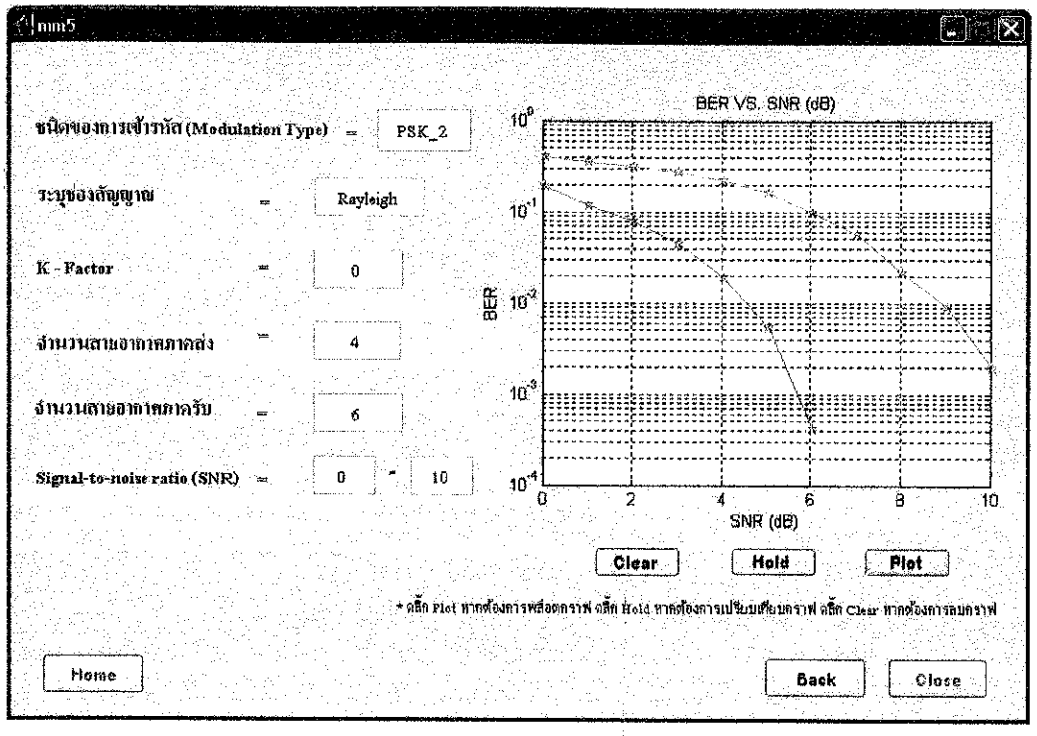
รูปที่ 3.3 หน้าต่างแสดงความสัมพันธ์ระหว่าง BER กับ SNR

และเมื่อผู้ใช้งานต้องการป้อนข้อมูลใหม่เพื่อเปรียบเทียบกับข้อมูลเดิม ผู้ใช้งานสามารถกดปุ่ม Hold เพื่อกลับไปป้อนค่าพารามิเตอร์ใหม่ได้อีกครั้งดังรูปที่ 3.4 จากนั้นโปรแกรมจะทำการเปรียบเทียบกราฟความสัมพันธ์ระหว่างข้อมูลใหม่และข้อมูลเดิมดังรูปที่ 3.5



รูปที่ 3.4 หน้าต่างแสดงการรับค่าพารามิเตอร์หลังจากที่ผู้ใช้งานกดปุ่ม Hold

ผู้ใช้งานสามารถทำการเปรียบเทียบค่าความสัมพันธ์ระหว่าง BER และ SNR เมื่อผู้ใช้งานใช้งานโปรแกรมเสร็จสิ้นแล้ว ผู้ใช้งานสามารถกดปุ่ม Close เพื่อปิดโปรแกรม หรือกดปุ่ม Home เพื่อกลับสู่หน้าเริ่มต้น

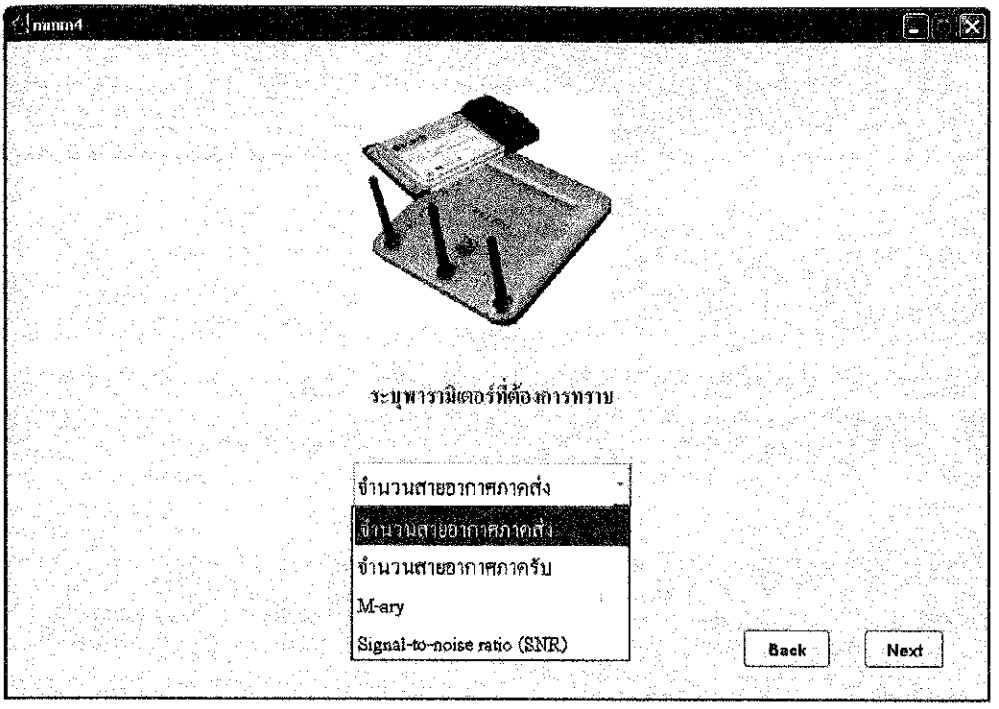


รูปที่ 3.5 หน้าต่างแสดงการเปรียบเทียบระหว่างข้อมูลใหม่กับข้อมูลเดิม

3.2 โปรแกรมการออกแบบพารามิเตอร์

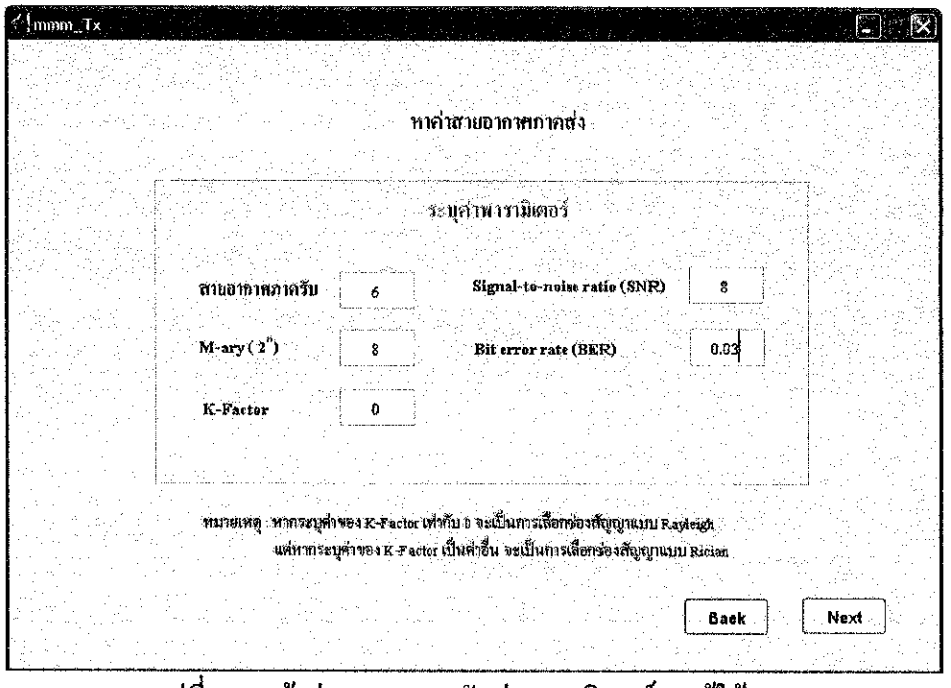
จากรูปที่ 3.1 เมื่อผู้ใช้งานเลือกการใช้งานโปรแกรมการออกแบบพารามิเตอร์ก็จะปรากฏกับหน้าต่างดังรูปที่ 3.6 ซึ่งเป็นหน้าต่างที่ผู้ใช้งานสามารถเลือกได้ว่าต้องการออกแบบพารามิเตอร์ตัวใดซึ่งโปรแกรมมีให้เลือก 4 พารามิเตอร์ที่สำคัญด้วยกันคือ

1. ออกแบบจำนวนสายอากาศภาคส่ง
2. ออกแบบจำนวนสายอากาศภาครับ
3. ออกแบบจำนวน M-ary
4. ออกแบบ Signal-to-Noise Ratio (SNR)



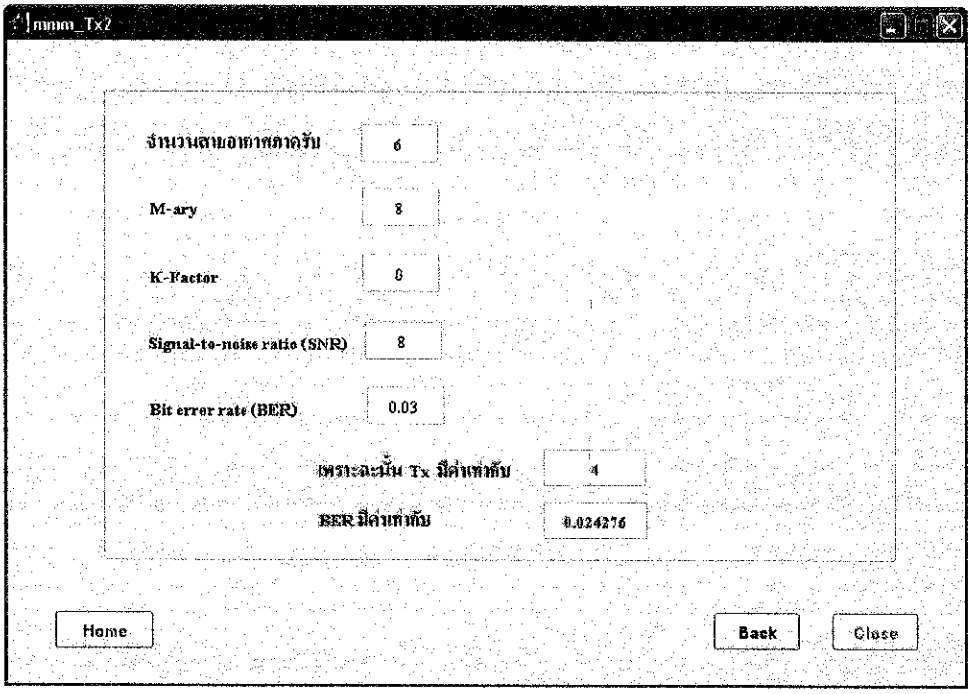
รูปที่ 3.6 หน้าต่างแสดงให้ผู้ใช้เลือกการออกแบบพารามิเตอร์

จากรูปที่ 3.7 เมื่อผู้ใช้งานเลือกออกแบบจำนวนสายอากาศภาคส่ง ผู้ใช้งานจำเป็นต้องป้อนพารามิเตอร์ที่ทราบค่า เพื่อโปรแกรมจะนำค่าเหล่านี้ไปประมวลต่อไป



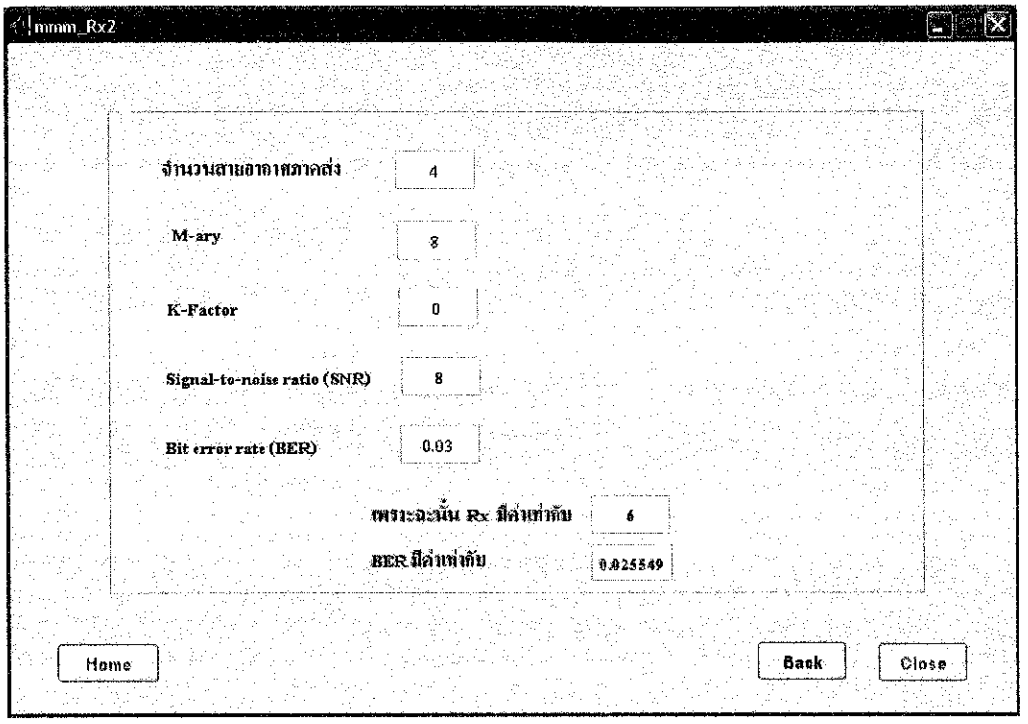
รูปที่ 3.7 หน้าต่างแสดงการรับค่าพารามิเตอร์จากผู้ใช้งาน

หลังจากรับค่าจากผู้ใช้งานแล้วโปรแกรมจะทำการประมวลผลเพื่อหาค่าจำนวนสายอากาศภาคส่งที่เหมาะสมที่สุด โดยพิจารณาจากค่าBERที่ระบบคำนวนได้ให้มีค่าใกล้เคียงกับค่าBERที่ผู้ใช้ป้อนเข้ามามากที่สุดและทำการแสดงผลดังรูปที่ 3.8

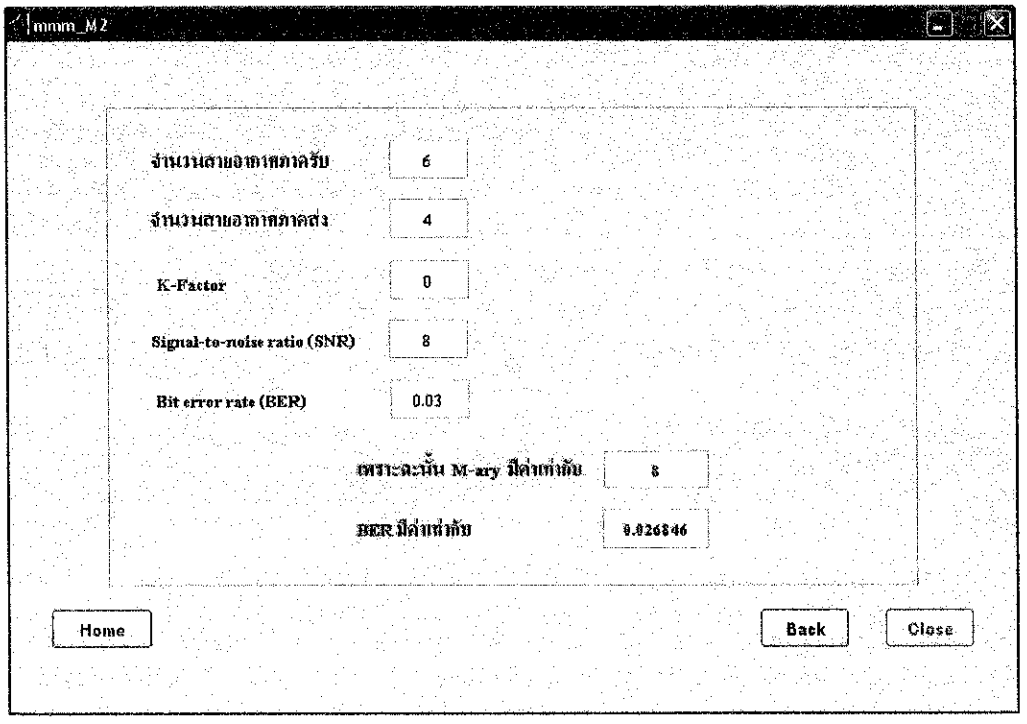


รูปที่ 3.8 หน้าต่างแสดงผลการออกแบบจำนวนสายอากาศภาคส่ง

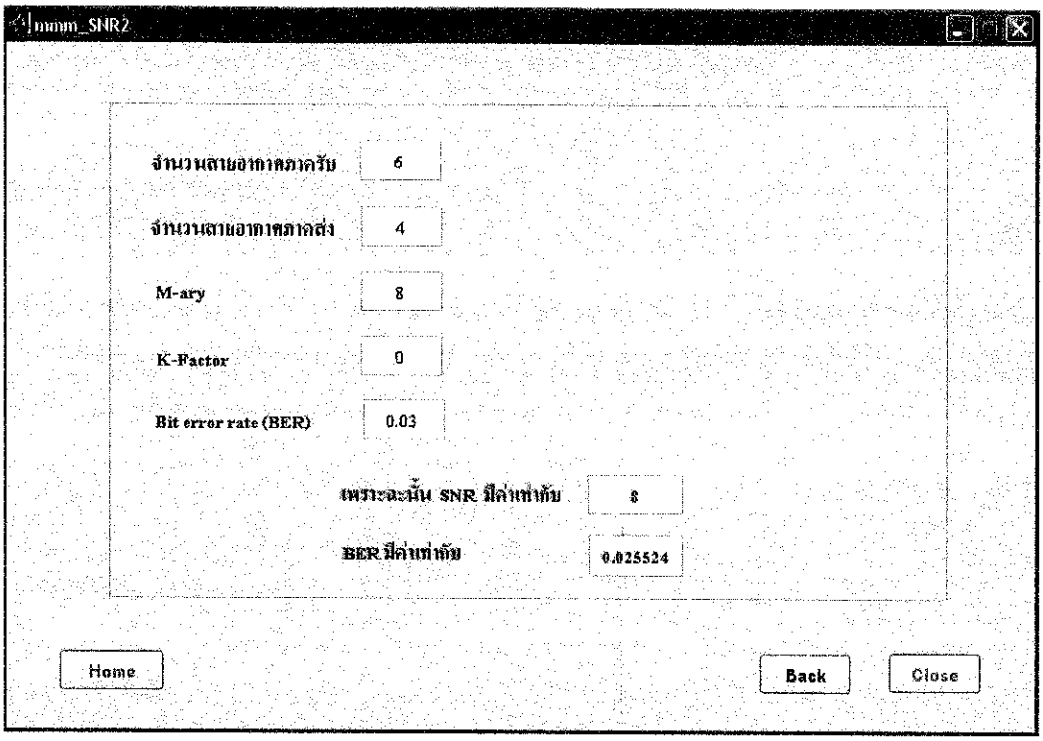
และเมื่อผู้ใช้งานต้องการออกแบบจำนวนสายอากาศภาครับ M-ary หรือSignal-to-Noise Ratio (SNR) ผู้ใช้สามารถทำตามขั้นตอนการออกแบบจำนวนสายอากาศภาคส่งและจากนั้นโปรแกรมจะทำการแสดงผลการออกแบบดังรูปที่ 3.9 รูปที่ 3.10 และรูปที่ 3.11 ตามลำดับ



รูปที่ 3.9 หน้าต่างแสดงผลการออกแบบจำนวนสายอากาศภาครับ



รูปที่ 3.10 หน้าต่างแสดงผลการออกแบบจำนวนM-ary



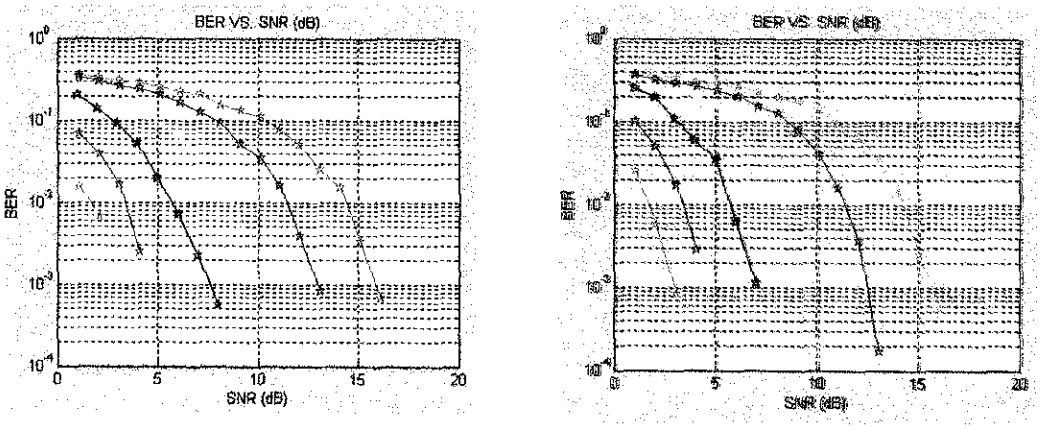
รูปที่ 3.11 หน้าต่างแสดงผลการออกแบบ SNR

บทที่ 4
การทดสอบและการวิเคราะห์

4.1 การวิเคราะห์ผลระหว่าง BER และ SNR

ในการพิจารณาระบบMIMOระบบหนึ่งๆว่ามีประสิทธิภาพดีหรือไม่นั้น สามารถพิจารณาได้จากความสัมพันธ์ระหว่างBERและSNR ซึ่งระบบที่ดีควรมีค่าSNRที่สูงและค่าBERที่ต่ำ โดยความสัมพันธ์ของค่าดังกล่าวนี้สามารถเปลี่ยนแปลงได้เมื่อมีการเปลี่ยนแปลงพารามิเตอร์บางอย่าง ดังนั้นจึงทำการศึกษาผลกระทบต่างๆที่ส่งผลให้ระบบเปลี่ยนแปลง โดยการเพิ่มหรือลดค่าพารามิเตอร์ต่างๆที่สำคัญดังต่อไปนี้

4.1.1 เมื่อเพิ่มจำนวน M-ary



(ก) ช่องสัญญาณแบบ Rayleigh (ข) ช่องสัญญาณแบบ Rician

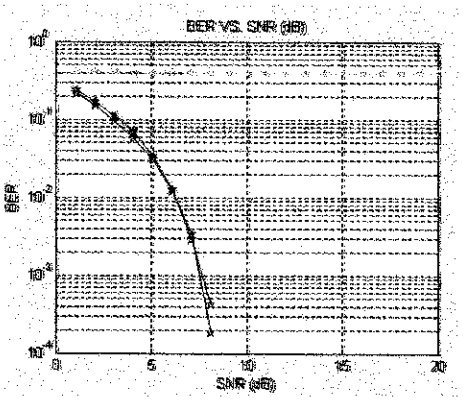
รูปที่ 4.1 กราฟความสัมพันธ์ระหว่าง BER และSNR ที่ 2PSK , 4PSK , 8PSK , 16PSK , 32PSK และ64PSK (ก) ช่องสัญญาณแบบ Rayleigh (ข) ช่องสัญญาณแบบ Rician

พิจารณาจากรูปที่ 4.1 เมื่อทำการเพิ่มจำนวน M-ary ($M=2^n$; $n=1,2,3,\ldots,8$) พบว่าแนวโน้มของค่าBERจะแปรผันตามM-ary นั่นคือ เมื่อM-aryเพิ่มขึ้น ค่าBERจะมีค่าเพิ่มขึ้นตามไปด้วย ซึ่งค่าBERที่เพิ่มขึ้นเช่นนี้ไม่ส่งผลที่ดีนักต่อระบบ แต่ในระบบการรับ-ส่งข้อมูลทั่วไปยังคงสามารถใช้ค่า

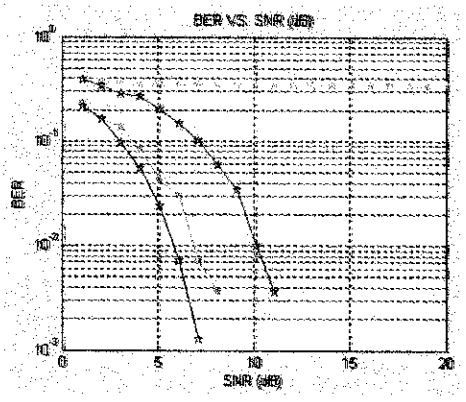
M-aryที่สูงอยู่ได้นั่นเป็นเพราะ เมื่อยิ่งใช้M-ary มาก บิตในการส่งข้อมูลก็จะมากขึ้นไปด้วย ดังนั้น ถึงแม้ว่าจะมีBERที่สูง แต่ระบบก็ยังคงมีวิธีในการตรวจสอบและแก้ไขจุดที่ผิดพลาดนี้ด้วยเช่นกัน

4.1.2 เมื่อเพิ่มจำนวนสายอากาศภาคส่ง

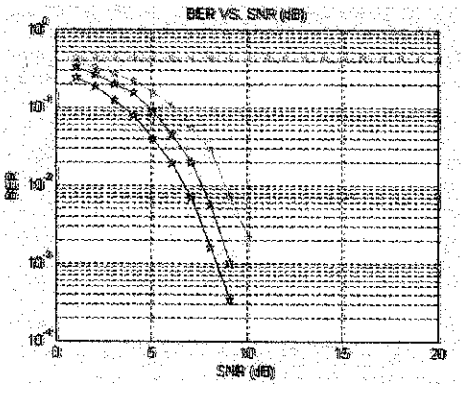
เมื่อทำการเพิ่มจำนวนสายอากาศทางภาคส่งโดยคงค่าพารามิเตอร์ตัวอื่นๆไว้ จะได้กราฟความสัมพันธ์ระหว่าง BER และSNR ดังแสดงในรูปที่ 4.2 ระบบจะมีประสิทธิภาพก็ต่อเมื่อจำนวนของสายอากาศภาคส่งมีจำนวนไม่เกินกว่าจำนวนของสายอากาศภาครับ แต่ถ้าจำนวนสายอากาศภาคส่งมีจำนวนมากกว่าจำนวนสายอากาศภาครับแล้ว ค่าของBERจะมีค่ามากแม้ว่าค่า SNR จะมีค่าเพิ่มขึ้นเท่าไรก็ตาม



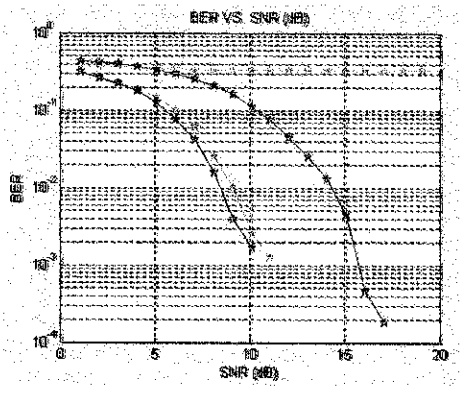
(ก) Rx=4 และ Tx=2,3,4,5



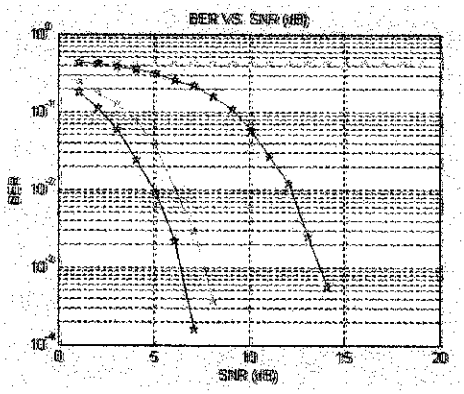
(ข) Rx=4 และ Tx=2,3,4,5



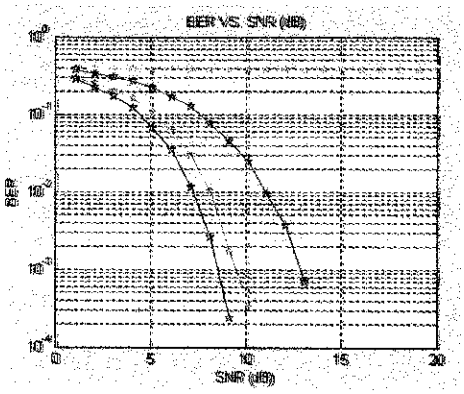
(ค) Rx=6 และ Tx=4,5,6,7



(ง) Rx=6 และ Tx=4,5,6,7



(จ) Rx=8 และ Tx=6,7,8,9

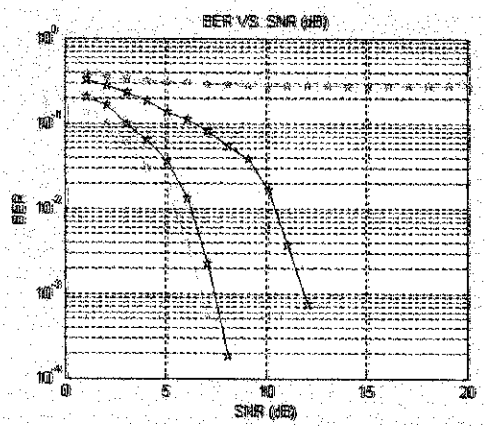


(ฉ) Rx=8 และ Tx=6,7,8,9

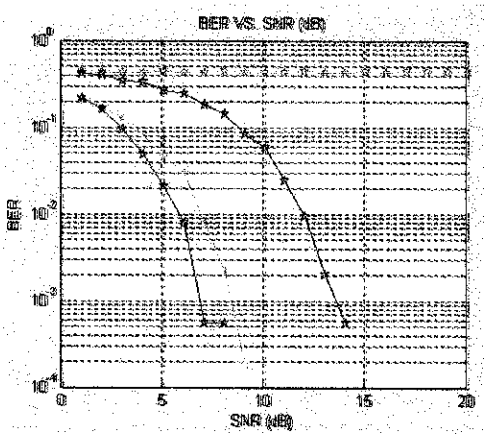
รูปที่4.2 กราฟความสัมพันธ์ระหว่าง BER และSNRเมื่อทำการเพิ่มจำนวนสายอากาศทางภาคส่ง โดย (ก),(ค)และ(จ) เป็นช่องสัญญาณแบบ Rayleigh และ(ข),(ง)และ(ฉ) เป็นช่องสัญญาณแบบRician

4.1.3 เมื่อเพิ่มจำนวนสายอากาศภาครับ

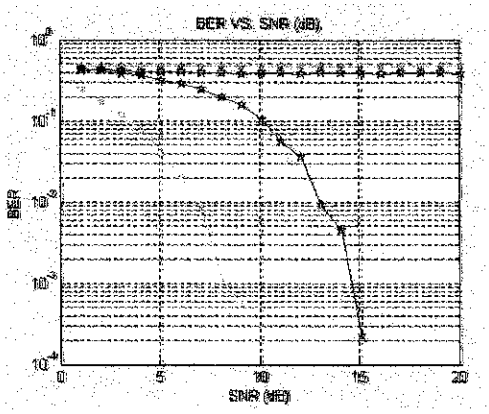
เช่นเดียวกับทางภาคส่ง เมื่อทำการเพิ่มจำนวนสายอากาศทางภาครับ โดยที่คงค่าพารามิเตอร์ตัวอื่นๆไว้ จะได้กราฟความสัมพันธ์ระหว่าง BER และSNR ดังแสดงในรูปที่ 4.3



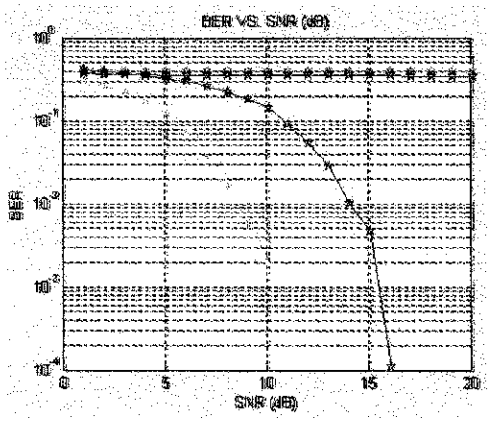
(ก) Tx=3 และ Rx=2,3,4,5



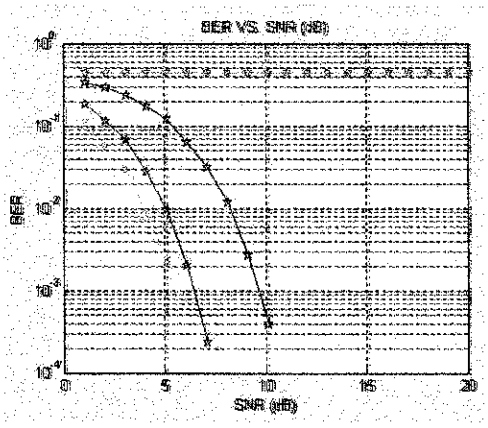
(ข) Tx=3 และ Rx=2,3,4,5



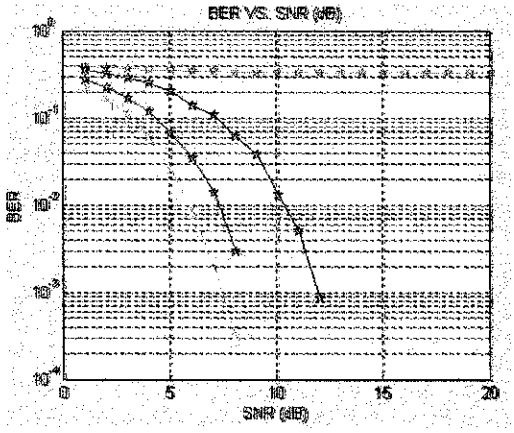
(ค) Tx=5 และ Rx=3,4,5,6



(ง) Tx=5 และ Rx=3,4,5,6



(จ) Tx=7 และ Rx=6,7,8,9



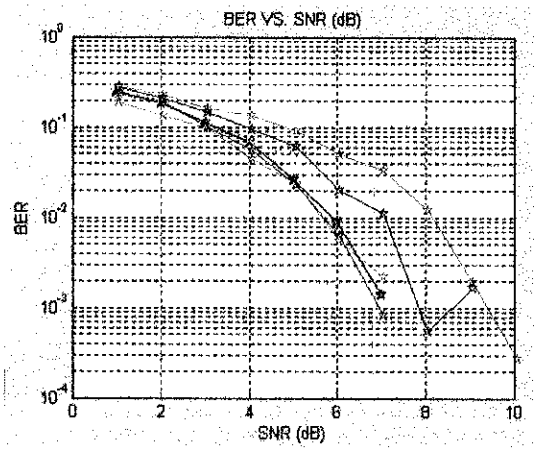
(ฉ) Tx=7 และ Rx=6,7,8,9

รูปที่4.3 กราฟความสัมพันธ์ระหว่าง BER และSNRเมื่อทำการเพิ่มจำนวนสายอากาศทางภาครับ โดย (ก),(ค)และ(จ) เป็นช่องสัญญาณแบบ Rayleigh และ (ข),(ง)และ(ฉ) เป็นช่องสัญญาณแบบRician

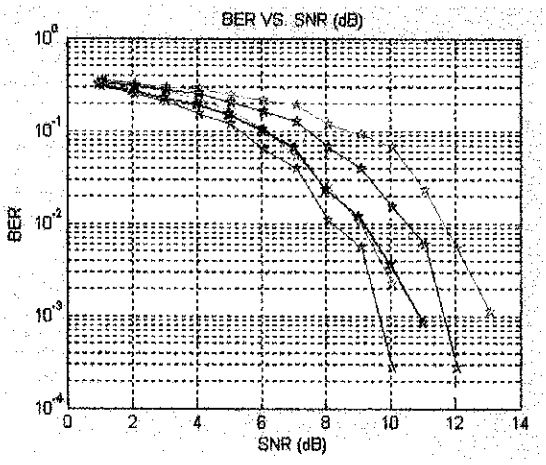
จากรูปที่ 4.3 เมื่อทำการเพิ่มจำนวนสายอากาศภาครับพบว่า ถ้าจำนวนของสายอากาศภาครับมีจำนวนสายอากาศมากกว่าหรือเท่ากับจำนวนของสายอากาศภาคส่ง จะส่งผลให้ระบบจะมีประสิทธิภาพดี แต่ถ้าจำนวนสายอากาศภาครับมีจำนวนน้อยกว่าจำนวนสายอากาศภาคส่งแล้ว ค่าของBERจะมีค่ามากแม้ว่าค่า SNR จะมีค่าเพิ่มขึ้นเท่าไรก็ตาม

4.1.4 เมื่อเพิ่มค่า K-Factor

เมื่อทำการเพิ่มค่า K-Factor และพิจารณากราฟความสัมพันธ์ระหว่าง BER และSNR ดังแสดงในรูปที่ 4.4 และรูปที่ 4.5 พบว่าเมื่อค่าK เพิ่มขึ้น ส่งผลให้แนวโน้มของค่าBERเพิ่มขึ้นตามไปด้วย ในการเลือกใช้ค่าKนั้น ควรพิจารณาจากสภาพแวดล้อมภายในระบบ โดยระบบที่มีการสะท้อนของสัญญาณที่สูงนั้น ค่าKจะมีค่าเข้าใกล้ ∞ ในทางตรงกันข้าม ระบบที่มีการสะท้อนของสัญญาณที่ต่ำหรือมีการรับ- ส่งข้อมูลเป็นแบบLOS ค่าKจะยิ่งมีค่าเข้าใกล้ 0



รูปที่4.4 กราฟความสัมพันธ์ระหว่าง BER และSNR ที่ Tx=2,Rx=3 และ K=0,1,2,4,5,10 ตามลำดับ



รูปที่4.5 กราฟความสัมพันธ์ระหว่าง BER และSNR ที่ Tx=4,Rx=6 และ K=1,2,3,4,5,10,20,50 ตามลำดับ

จากการศึกการผลกระทบที่ส่งผลให้ค่าBERและSNRของระบบเปลี่ยนแปลง โดยทำการเปลี่ยนแปลงค่าพารามิเตอร์ต่างๆดังที่กล่าวมาพบว่า ค่าพารามิเตอร์ต่างๆส่งผลต่อการเปลี่ยนแปลงค่าBERที่แตกต่างกันไป ในการเลือกค่าBERให้แก่ระบบหนึ่งๆนั้นจึงจำเป็นต้องดูองค์ประกอบต่างๆของระบบเช่น สภาพแวดล้อมภายในระบบ จำนวนสายอากาศที่ใช้รับ-ส่ง วิธีการมอดูเลชั่นของสัญญาณ เป็นต้น และทำการเลือกค่าBERที่เหมาะสมที่สุดให้กับระบบนั้นๆ เพื่อให้เกิดประสิทธิภาพสูงสุดแก่ระบบ

บทที่ 5
บทสรุป ปัญหาและข้อเสนอแนะ

5.1 บทสรุป

โปรแกรมวิเคราะห์และออกแบบระบบ MIMO ด้วย GUI เป็นโปรแกรมที่ถูกสร้างขึ้นมาเพื่อสนองตอบต่อความต้องการของผู้ใช้งานในระบบMIMO ที่มีการประมวลผลได้เร็วและไม่ซับซ้อนทำให้ง่ายต่อการใช้งานของผู้ใช้ แบ่งการใช้งานของโปรแกรมออกเป็น 2 ส่วนใหญ่ๆ ดังนี้

5.1.1 โปรแกรมเพื่อการวิเคราะห์อัตราการผิดพลาดของข้อมูล(BER) เป็นโปรแกรมที่แสดงความสัมพันธ์ระหว่างค่าBER กับ SNR ให้อยู่ในรูปของกราฟความสัมพันธ์ ทำให้ผู้ใช้งานสามารถวิเคราะห์และเปรียบเทียบข้อมูลเพื่อเลือกค่าที่เหมาะสมไปใช้งานได้

5.1.2 โปรแกรมเพื่อการออกแบบพารามิเตอร์ เป็นโปรแกรมที่สามารถคำนวณหาค่าพารามิเตอร์ที่ต้องการได้อย่างถูกต้องและแม่นยำ โดยโปรแกรมสามารถหาค่าพารามิเตอร์ต่างๆ ได้ดังนี้

1. ออกแบบจำนวนสายอากาศภาคส่ง
2. ออกแบบจำนวนสายอากาศภาครับ
3. ออกแบบจำนวน M-ary
4. ออกแบบ Signal-to-Noise Ratio (SNR)

5.2 ปัญหาที่พบ

5.2.1 เนื่องจากระบบMIMO เป็นระบบที่ใหม่ จึงทำให้ข้อมูลในการศึกษาค้นคว้ามีน้อย ทำให้การค้นคว้าและการศึกษาข้อมูลเกิดความล่าช้า

5.2.2 ในส่วนของตัวโปรแกรมภาคออกแบบพารามิเตอร์เกิดความไม่เสถียรของระบบ โดยค่าพารามิเตอร์ที่โปรแกรมคำนวณได้มีค่าความผิดพลาด ±1 จึงทำให้โปรแกรมไม่น่าเชื่อถือ จึงทำการตรวจสอบโปรแกรม

```
for i1=1:1
        G = randn(nr,nt)+(i*randn(nr,nt)) ;
        Hnlos = G*(sqrt(1/(K+1)));
%=============================LOS=============================
    x = sqrt(nr*nt);
    Hm=zeros(nr,nt);
    Hm(1,1)=x;
    Hlos = Hm*(sqrt(K/(K+1)));
%============================ H-Rician=========================
    H = Hnlos+Hlos;
    for index=1:20
      kk=1/(10^(snr/10));
      amplitudenoise_a = randn(nr,L).*kk;
      amplitudenoise_b = randn(nr,L).*kk;
      amplitudenoise = amplitudenoise_a+(amplitudenoise_b.*i);
      Y = (H*SM)+amplitudenoise ;    %received vector

%=========================Receivtion=========================
      BB  =inv(H'*H)*H'*Y;
      X = reshape(BB,len,1) ;      %transmit vertor

%=========================psk Demodulation=========================
      recovpsk = pskdemod(X,M);      %received Data
      [numerrs_psk(index,i1),Rat(index,i1)] = biterr(msg,recovpsk);

%=========================Find BER=========================
    end
    mRat(nr)=mean(mean(Rat,2));
  end
```

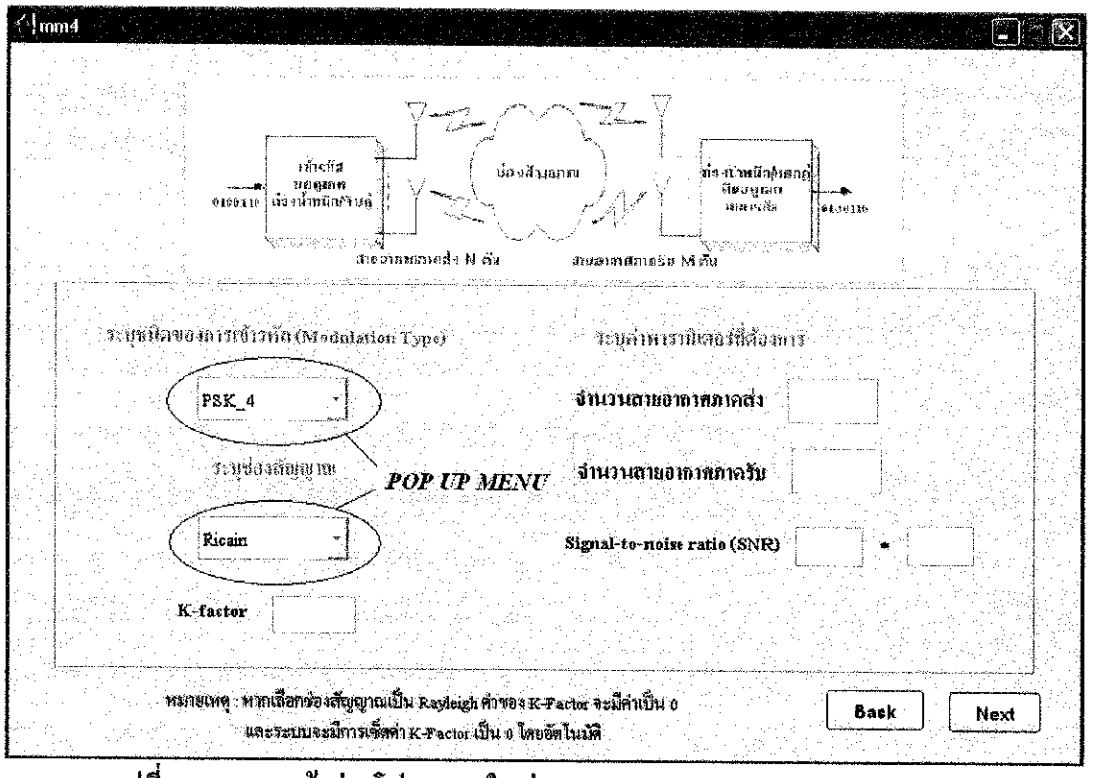
จากโปรแกรมข้างต้นพบว่าตัวแปร G ที่ใช้ในการสร้างช่องสัญญาณมีการสุ่มค่าใหม่ทุกรอบ ทำให้ช่องสัญญาณมีการสร้างใหม่ขึ้นทุกครั้งเช่นกัน จึงทำให้โปรแกรมไม่เสถียรและเกิดความผิดพลาดขึ้น ดังนั้นจึงทำการแก้ไขโดยเพิ่มคำสั่ง

```
randn('seed',1111212);
```

ซึ่งคำสั่งนี้ทำให้ตัวแปร G สุ่มค่าตัวเดิมทุกรอบการวนลูป

5.3 ข้อเสนอแนะ

เนื่องจากการรับค่าจากผู้ใช้งานในส่วนของ POP UP MENU นั้น ควรมีการเลือกค่าใหม่ทุกครั้ง เนื่องจากโปรแกรมจะมีการจำค่าเดิมที่เคยเลือกไว้



รูปที่ 5.1 แสดงหน้าต่างโปรแกรมในส่วนของ POP UP MENU

5.4 แนวทางการพัฒนาต่อไป

5.4.1 เนื่องจากโปรแกรมนี้ใช้การเข้ารหัสแบบPSK แต่ในการใช้งานทั่วไปยังมีการเข้ารหัสแบบอื่นๆอีกเช่น FSK , ASK และ QAM เป็นต้น จึงควรพัฒนาให้มีการเข้ารหัสได้ทุกรูปแบบ

5.4.2 เทคนิคการส่งสัญญาณในระบบMIMOมี2ชนิดด้วยกันคือ Spatial Multiplexing (SM) และ Space–Time Block Code (STBC) โดยเทคนิคที่ทางโครงงานเลือกใช้คือ SM เพราะฉะนั้นจึงควรพัฒนาให้สามารถเลือกการส่งสัญญาณได้ทั้ง2ชนิด

บรรณานุกรม

[1] http://mimo.wikidot.com/introduction

[2]http://209.85.175.104/search?q=cache:6olrWfpUyTMJ:www.antthai.com/home/article/mimo.htm+mimo&hl=th&ct=clnk&cd=1&gl=th&lr=lang_th

[3] http://eng.sut.ac.th/tce/SeniorProjects/2550/50_2.html

[4] http://www.eecs.umich.edu/~hero/Preprints/ursi02_final.pdf

[5] http://en.wikipedia.org/wiki/Rayleigh_fading

[6]http://images.google.co.th/imgres?imgurl=http://users.ece.utexas.edu/~jandrews/molabview_fizes/mimoofdm.gif&imgrefurl=http://users.ece.utexas.edu/~jandrews/molabview.html&h=200&w=550&sz=19&hl=th&start=2&tbnid=qgb4clNlH6lfsM:&tbnh=48&tbnw=133&prev=/images%3Fq%3Dspatial%2Bmultiplexing%26gbv%3D2%26hl%3Dth%26sa%3DG

[7] http://en.wikipedia.org/wiki/Spatial_multiplexing

[8] http://www.ist-astrals.org/Public_Docs/CNF%20-%20IST%20Mobile%20Summit,%20Jun%202006.pdf

[9] http://en.wikipedia.org/wiki/Spatial_multiplexing

[11]http://images.google.co.th/imgres?imgurl=http://upload.wikimedia.org/wikipedia/commons/thumb/7/77/PSK_BER_curves.svg/280px-PSK_BER_curves.svg.png&imgrefurl=http://en.wikipedia.org/wiki/Phase-shift_keying&h=221&w=280&sz=21&hl=th&start=1&um=1&tbnid=NucBoFCb1VuzhM:&tbnh=90&tbnw=114&prev=/images%3Fq%3Dbit%2Berror%2Brate%2Bpsk%26um%3D1%26hl%3Dth

[10]http://www.highfrequencyelectronics.com/Archives/Jan03/HFE0103_Tutorial.pdf

ภาคผนวก ก

โปรแกรมการวิเคราะห์อัตราการผิดพลาดของการส่งข้อมูล (BER)

```
function[snr,MeanRat] = pBer_rician(nnt,nnr,MM,KF,SSNR1,SSNR2);
%create psk modulation
nt=str2double(nnt);
nr=str2double(nnr);
M=MM;
K=str2double(KF);
SNR1=str2double(SSNR1);
SNR2=str2double(SSNR2);

A=600;
len = A*nt;
L = len/nt;
msg = randint(len,1,M) ;
txpsk = pskmod(msg,M);
%create SM modulation
SM = reshape(txpsk,nt,L) ;
%========================= In freespace ==========================
snr = SNR1:1:SNR2; % dB
randn('seed',1111212);
for i1=1:40
   randn('seed',1111212);
   GG = randn(nr,nt)+(i*randn(nr,nt)) ;
   Hnlos = GG*(sqrt(1/(K+1)));
```

```
%LOS
   x = sqrt(nr*nt);
   Hm=zeros(nr,nt);
   Hm(1,1)=x;
   Hlos = Hm*(sqrt(K/(K+1)));
   % H-Rician
   H = Hnlos+Hlos;
   % noise
   for index=1:length(snr)
      kk=1/(10^(snr(index)/10));
      amplitudenoise_a = randn(nr,L).*kk;
      amplitudenoise_b = randn(nr,L).*kk;
      amplitudenoise = amplitudenoise_a+(amplitudenoise_b.*i);
% H-channal
      %NLOS
      Y = (H*SM)+amplitudenoise ;    %received vector
      %=======================Receivtion=============================
      BB  =inv(H'*H)*H'*Y;
      X = reshape(BB,len,1) ;       %transmit vertor
 %=======================psk Demodulation=========================
      recovpsk = pskdemod(X,M);      %received Data
      [numerrs_psk(index,i1),Rat(index,i1)] = biterr(msg,recovpsk); %Find BER
      index
   end
end
MeanRat = mean(Rat,2);
```

ภาคผนวก ข

โปแกรมการออกแบบพารามิเตอร์

1. โปรแกรมออกแบบจำนวนสายอากาศภาคส่ง

```
function[N_nt,mbb_ber] = nt_rician(nnr,bber,ssnr,MM,KF)
% ================== Fix paramiter=================================
nr=str2double(nnr);
ber=str2double(bber);
snr=str2double(ssnr);
M=str2double(MM);
K=str2double(KF);
mRat=1:20;
for   nt=1:1:20;
   %create psk modulation
   A=5/ber;
   A=round(A);
   len = A*nt;
   L = len/nt;
   msg = randint(len,1,M) ;
   txpsk = pskmod(msg,M);
   %create SM modulation
   SM = reshape(txpsk,nt,L) ;
   %================== In freespace ==========================
   randn('seed',1111212);
   for i1=1:1
      G = randn(nr,nt)+(i*randn(nr,nt)) ;
      Hnlos = G*(sqrt(1/(K+1)));
```

```
%LOS
    x = sqrt(nr*nt);
    Hm=zeros(nr,nt);
    Hm(1,1)=x;
    Hlos = Hm*(sqrt(K/(K+1)));
    % H-Rician
    H = Hnlos+Hlos;
    % noise
    for index=1:20
      kk=1/(10^(snr/10));
      amplitudenoise_a = randn(nr,L).*kk;
      amplitudenoise_b = randn(nr,L).*kk;
      amplitudenoise = amplitudenoise_a+(amplitudenoise_b.*i);
 % H-channal
1.      %NLOS
2.      Y = (H*SM)+amplitudenoise ;    %received vector
3.%=====================Receivtion=====================================
4.      BB  =inv(H'*H)*H'*Y;
        X = reshape(BB,len,1) ;       %transmit vertor
 %=====================psk Demodulation=================================
        recovpsk = pskdemod(X,M);      %received Data
        [numerrs_psk(index,i1),Rat(index,i1)] = biterr(msg,recovpsk); %Find BER
    end
    mRat(nt)=mean(mean(Rat,2));
  end
```

```
if mRat(nt)>=ber
      N_nt=nt-1;
      mbb_ber=mRat(nt);
      break;
   end
end
mRat2=mRat(1);
N_nt=1;
for ppp=2:1:length(mRat)
   if (mRat(ppp) > mRat2) && (mRat(ppp) < ber)



      mRat2=mRat(ppp);
      N_nt=ppp;
      mbb_ber=mRat(ppp);
   end
end
```

2. โปรแกรมออกแบบจำนวนสายอากาศภาครับ

```
function[N_nr,mbb_ber] = nr_rician(nnt,bber,ssnr,MM,KF);
%========== Fix paramiter ==========
nt=str2double(nnt);
ber=str2double(bber);
snr=str2double(ssnr);
M=str2double(MM);
K=str2double(KF);
mRat=1:10;
%create psk modulation
A=5/ber;
A=round(A);
len = A*nt;
L = len/nt;
msg = randint(len,1,M) ;
txpsk = pskmod(msg,M);
%create SM modulation
SM = reshape(txpsk,nt,L) ;
check = false;
for   nr=1:1:20;
%========== In freespace ==========
   randn('seed',1111212);
   for i1=1:1
      G = randn(nr,nt)+(i*randn(nr,nt)) ;
      Hnlos = G*(sqrt(1/(K+1)));
%LOS
      x = sqrt(nr*nt);
```

```
Hm=zeros(nr,nt);
Hm(1,1)=x;
Hlos = Hm*(sqrt(K/(K+1)));
% H-Rician
H = Hnlos+Hlos;
% noise
for index=1:20
  kk=1/(10^(snr/10));
  amplitudenoise_a = randn(nr,L).*kk;
  amplitudenoise_b = randn(nr,L).*kk;
  amplitudenoise = amplitudenoise_a+(amplitudenoise_b.*i);
% H-Rician
H = Hnlos+Hlos;
% noise
for index=1:20
  kk=1/(10^(snr/10));
  amplitudenoise_a = randn(nr,L).*kk;
  amplitudenoise_b = randn(nr,L).*kk;
  amplitudenoise = amplitudenoise_a+(amplitudenoise_b.*i);
  % H-channal
  %NLOS
  Y = (H*SM)+amplitudenoise ;    %received vector
  %=========================Receivtion=============================
  BB  =inv(H'*H)*H'*Y;
  X = reshape(BB,len,1) ;      %transmit vertor
%=========================psk Demodulation=========================
  recovpsk = pskdemod(X,M);     %received Data
 [numerrs_psk(index,i1),Rat(index,i1)] = biterr(msg,recovpsk); %Find BER
```

```
        end
        mRat(nr)=mean(mean(Rat,2));
    end
    if mRat(nr)<=ber
        if check
            if mRat(nr)==0
                break;
            end
        end
        check = true;
    end
end
check2 = true;
mRat2=0.00000001;
for ppp=1:length(mRat)
    if mRat(ppp) < ber
        if mRat(ppp) > mRat2
            mRat2 = mRat(ppp);
            N_nr = ppp;
            mbb_ber=mRat(nr);
        end
        if check2
            N_nr=ppp;
            mbb_ber=mRat(ppp);
            check2 = false;
        end
    end
end
```

3. โปรแกรมออกแบบจำนวน M-ary

```
function[M_arry,mbb_ber] = M_rician(nnt,nnr,bber,ssnr,KF);
nt=str2double(nnt);
nr=str2double(nnr);
ber=str2double(bber);
snr=str2double(ssnr);
K=str2double(KF);
%========== Fix paramiter ==========
%create psk modulation
A=5/ber;
A=round(A);
len = A*nt;
L = len/nt;
for   nn=1:1:8;
   M = 2^nn;
   msg = randint(len,1,M) ;
   txpsk = pskmod(msg,M);
   %create SM modulation
   SM = reshape(txpsk,nt,L) ;
   %========== In freespace ==========
   randn('seed',1111212);
   for i1=1:1
      G = randn(nr,nt)+(i*randn(nr,nt)) ;
      Hnlos = G*(sqrt(1/(K+1)));
%LOS
      x = sqrt(nr*nt);
      Hm=zeros(nr,nt);
      Hm(1,1)=x;
      Hlos = Hm*(sqrt(K/(K+1)));
```

```
% H-Rician
      H = Hnlos+Hlos;
      % noise
      for index=1:20
        kk=1/(10^(snr/10));
        amplitudenoise_a = randn(nr,L).*kk;
        amplitudenoise_b = randn(nr,L).*kk;
        amplitudenoise = amplitudenoise_a+(amplitudenoise_b.*i);
        % H-channal
        %NLOS
        Y = (H*SM)+amplitudenoise ;    %received vector
%=========================Receivtion=============================
        BB  =inv(H'*H)*H'*Y;
        X = reshape(BB,len,1) ;       %transmit vertor
%=========================psk Demodulation=======================
        recovpsk = pskdemod(X,M);      %received Data
        [numerrs_psk(index,i1),Rat(index,i1)] = biterr(msg,recovpsk); %Find BER
      end
      mRat=mean(mean(Rat,2));
      mmRat(nn)=mRat;
    end
    if   mRat>=ber
      nn_A=nn-1
      M_arry=2^nn_A;
      mbb_ber=mmRat(nn_A);
      break;
    end
  end
```

4. โปรแกรมออกแบบ Signal-to-Noise Ratio (SNR)

```
function[snr_dB,mbb_ber] = snr_rician(nnr,nnt,bber,MM,KF);
nr=str2double(nnr);
nt=str2double(nnt);
ber=str2double(bber);
M=str2double(MM);
K=str2double(KF);
A=5/ber;
A=round(A);
len = A*nt;
L = len/nt;
%create psk modulation
msg = randint(len,1,M) ;
txpsk = pskmod(msg,M);
%create SM modulation
SM = reshape(txpsk,nt,L) ;
for   snr=0:1:100;
  %========================= In freespace ==========================
  randn('seed',1111212);
  for i1=1:1
    G = randn(nr,nt)+(i*randn(nr,nt)) ;
    Hnlos = G*(sqrt(1/(K+1)));
    %LOS
    x = sqrt(nr*nt);
    Hm=zeros(nr,nt);
    Hm(1,1)=x;
    Hlos = Hm*(sqrt(K/(K+1)));
```

```
% H-Rician
    H = Hnlos+Hlos;
    % noise
    for index=1:20
       kk=1/(10^(snr/10));
       amplitudenoise_a = randn(nr,L).*kk;
       amplitudenoise_b = randn(nr,L).*kk;
       amplitudenoise = amplitudenoise_a+(amplitudenoise_b.*i);
       % H-channal
       %NLOS
       Y = (H*SM)+amplitudenoise ;    %received vector
%=========================Receivtion=============================
       BB  =inv(H'*H)*H'*Y;
       X = reshape(BB,len,1) ;       %transmit vertor
%=========================psk Demodulation=======================
       recovpsk = pskdemod(X,M);      %received Data
       [numerrs_psk(index,i1),Rat(index,i1)] = biterr(msg,recovpsk); %Find BER
    end
    mRat=mean(mean(Rat,2));
  end
  if    mRat<=ber
    mbb_ber=mRat
    snr_dB=snr
    break;
  end
end
```

ศูนย์บรรณสารและสื่อการศึกษา
มหาวิทยาลัยเทคโนโลยีสุรนารี