

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

เครื่องรับสัญญาณดาวเทียมบีคอนแบบดิจิทัลเพื่อการคำนวณ
ความหนาแน่นอิเล็กทรอนิกส์ในชั้นบรรยากาศ

DIGITAL BEACON RECEIVER FOR TOTAL ELECTRON
CONTENT MEASUREMENT DEVELOPED WITH GNU RADIO



T117528



เลขหมู่.....
เลขทะเบียน 117528
วัน,เดือน,ปี..... - 5 ส.ค. 2554

b. 12328005
i.....

ปริญญาโทนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมโทรคมนาคม
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2553

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เครื่องรับสัญญาณดาวเทียมบีคอนแบบดิจิทัลเพื่อการคำนวณ
ความหนาแน่นอิเล็กทรอนิกส์ในชั้นบรรยากาศ
DIGITAL BEACON RECEIVER FOR TOTAL ELECTRON
CONTENT MEASUREMENT DEVELOPED WITH GNU RADIO



ปริญญาานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมโทรคมนาคม
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2553

ผ่านการตรวจเรียบร้อยแล้ว

(ลงชื่อ) *alsh* ผู้ตรวจ

ผ่านการตรวจเรียบร้อยแล้ว

(ลงชื่อ) *Stonuyaporn* ผู้ตรวจ

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น หากมีการเปลี่ยนแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาานิพนธ์ปีการศึกษา 2553

สาขาวิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง เครื่องรับสัญญาณดาวเทียมบีคอนแบบดิจิทัลเพื่อการคำนวณ


ความหนาแน่นอิเล็กตรอนในชั้นบรรยากาศ

DIGITAL BEACON RECEIVER FOR TOTAL ELECTRON CONTENT

MEASUREMENT DEVELOPED WITH GNU RADIO

ผู้จัดทำ

1. นายเจริญชัย ดวงรัตนธาดา 50010267
2. นายวิภัต ผลสุขการ 50011472
3. นางสาวอัสมา แท้จินดา 50011927


อาจารย์ที่ปรึกษา
(รศ.ดร.ปราโมทย์ วาดเขียน)


อาจารย์ที่ปรึกษาร่วม
(รศ.ดร.พรชัย ทร์พจน์นิธิ)

กิตติกรรมประกาศ

โครงการนี้จะสำเร็จลุล่วงไม่ได้เลย ถ้าปราศจากการช่วยเหลือ แนะนำ และให้คำปรึกษาของบรรดาผู้มีพระคุณเหล่านี้ คือ รศ.ดร.ปราโมทย์ วาดเขียน และ รศ.ดร.พรชัย ทรัพย์นิธิ ผู้ซึ่งเป็นที่ปรึกษาให้กับโครงการนี้ Professor Mamoru Yamamoto ที่คอยให้คำแนะนำและให้ข้อมูลที่จำเป็นในการทำโครงการนี้ พี่ๆ Supervisor ทุกคนและพี่ๆ ทุกคน รวมทั้งเพื่อนๆ ที่คอยช่วยเหลือมาตลอด พร้อมทั้งให้คำปรึกษาด้านโปรแกรมที่จำเป็นสุดท้ายนี้ขอขอบพระคุณครอบครัวที่คอยเป็นแรงผลักดันโครงการนี้เรื่อยมา ขอขอบพระคุณทุกๆ คนอย่างสูง



นายเจริญชัย

ดวงรัตนฉายา

นายวิภัส

ผลสุขการ

นางสาวจันทร์หอม

แต่จินดา

ผู้จัดทำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**เครื่องรับสัญญาณดาวเทียมบีคอนแบบดิจิทัลเพื่อการคำนวณ
ความหนาแน่นอิเล็กตรอนในชั้นบรรยากาศ
DIGITAL BEACON RECEIVER FOR TOTAL ELECTRON CONTENT
MEASUREMENT DEVELOPED WITH GNU RADIO**

โดย นายเจริญชัย	ดวงรัตน์ฉายา 50010267
นายวิภัส	ผลสุขการ 50011472
นางสาวจันทร์หอม	แต่จินดา 50011927

อาจารย์ที่ปรึกษา	รศ.ดร.ปราโมทย์ วาดเขียน
อาจารย์ที่ปรึกษาร่วม	รศ.ดร.พรชัย ทรัพย์นิธิ

บทคัดย่อ

โครงการนี้ทำการออกแบบและสร้างวงจรกรองแถบความถี่ผ่าน เพื่อกรองสัญญาณบีคอน (Beacon signal) ที่รับมาจากดาวเทียมวงโคจรต่ำ 2 ความถี่ คือ 150 MHz และ 400MHz โดยวงจรกรองมีแบนวิดท์ 10 MHz และ 27 MHz ตามลำดับ จากนั้นนำสัญญาณบีคอนที่ได้ มาแปลงเป็นสัญญาณดิจิทัลด้วยชุดพัฒนา Universal Software Radio Peripheral ซึ่งในที่นี้เราใช้เทคโนโลยี Software defined Radio ในการควบคุม จากนั้นวิเคราะห์คำนวณหาค่าความต่างเฟสของสัญญาณสองความถี่ซึ่งเกิดจากผลของการหักเหของคลื่นที่แปรผันตามปริมาณอิเล็กตรอนบนชั้นบรรยากาศก่อนลงมาสู่สถานีรับภาคพื้นดิน และบันทึกไว้ในคอมพิวเตอร์ โครงการนี้เป็นส่วนหนึ่งของการศึกษาวิจัย เรื่อง “New development of digital beacon receiver base on GNU radio” ซึ่งสามารถนำค่าความต่างเฟสที่หาได้ไปคำนวณหาปริมาณอิเล็กตรอนรวมในชั้นบรรยากาศต่อไป

ABSTRACT

This project will design two band pass filters whose center frequencies are at 150 MHz and 400 MHz with 10 MHz and 27 MHz bandwidth, respectively. Then the beacon signals will be changed to the digital signal by the USRP board controlled by Software defined Radio then send to the PC for analyze signal to find the phase difference between two types of frequencies. We use the characteristic of frequency dependence of the refractive index of radio waves in the ionospheric plasma before the signals coming to the ground base station.

สารบัญ

		หน้า
	กิตติกรรมประกาศ	I
	บทคัดย่อ	II
	สารบัญ	III
	สารบัญรูปภาพ	V
	สารบัญตาราง	XI
บทที่ 1	บทนำ	1
	1.1 หลักการและเหตุผลของโครงการวิจัย	1
	1.2 วัตถุประสงค์	3
	1.3 ขอบเขตของการทำโครงการ	3
บทที่ 2	ทฤษฎีและหลักการที่เกี่ยวข้อง	4
	2.1 ปริมาณอิเล็กทรอนิกส์บนชั้นบรรยากาศไอโอโนสเฟียร์	4
	2.2 วิทยุควบคุมด้วยซอฟต์แวร์ (Software-Defined Radio)	5
	2.3 The Universal Software Radio Peripheral (USRP)	8
	2.4 GNU Radio	13
	2.5 วงจรกรองความถี่ (Filter)	14
	2.6 ประเภทของวงจรกรอง	15
	2.7 วงจรก้ำรเข้าคู่ (Coupled Resonator)	18
	2.8 วงโคจรดาวเทียม	24
	2.9 Keplerien Elements (Two line Element : TLE)	28
	2.10 กราฟฟิคอล ยูเซอร์ อินเตอร์เฟส (Graphical user interface (GUI))	32
	2.11 ภาษาไพธอน	33
บทที่ 3	การออกแบบและการจัดทำปริญญานิพนธ์	36
	3.1 หลักการทำงาน โดยรวมของระบบ	36
	3.2 การออกแบบวงจรกรองความถี่ผ่านแบบเข้าคู่ (Coupled Resonator)	36
	3.3 โปรแกรมควบคุมการทำงานของระบบ	48
	3.4 การประเมินผลข้อมูล / คำนวณหาค่าเฟส	60

สารบัญ (ต่อ)

	หน้า
3.5 เครื่องมือที่ใช้ในการทดลอง	63
3.6 การจัดเก็บผลการทดลอง	63
บทที่ 4 ผลการทดลอง	65
4.1 วงจรกรองแถบความถี่ผ่าน	65
4.2 โปรแกรมควบคุมการบันทึกสัญญาณบีคอน	83
4.3 อินเตอร์เฟซโปรแกรมการคำนวณการโคจรผ่านของดาวเทียม ในการบันทึกสัญญาณบีคอน	88
4.4 ภาคการประมวลผลข้อมูลสัญญาณบีคอน	92
บทที่ 5 สรุปผลและข้อเสนอแนะ	102
5.1 สรุปผล	102
5.2 ข้อเสนอแนะ	103
บรรณานุกรม	104
ภาคผนวก	105

สารบัญรูปภาพ

รูปที่	หน้า
1.1	2
1.2	2
2.1	5
2.2	6
2.3	7
2.4	9
2.5	9
2.6	10
2.7	11
2.8	12
2.9	12
2.10	15
2.11	16
2.12	17
2.13	17
2.14	19
2.15	20
2.16	25
2.17	26
2.18	26
2.19	27
2.20	29
2.21	35
3.1	36
3.2	38

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ (ต่อ)

รูปที่	หน้า
3.2 รูปแบบทั่วไปของวงจรรองความถี่ผ่านแบบ Coupled Resonator	38
3.3 วงจรรองความถี่ผ่าน 150 MHz	38
3.4 วงจรรองความถี่ผ่านหลังจากการกำหนด โหนด	39
3.5 วงจร Transmission Line	39
3.6 รูปแบบทั่วไปของวงจรรองความถี่ผ่านแบบ Coupled Resonator	42
3.7 วงจรรองความถี่ผ่าน 400 MHz	42
3.8 วงจรรองความถี่ผ่านหลังจากการกำหนด โหนด	43
3.9 วงจร Transmission Line	43
3.10 วงจรรองความถี่ผ่าน 150 MHz	44
3.11 วงจรรองความถี่ผ่านหลังจากการกำหนด โหนด	45
3.12 วงจร Transmission Line	45
3.13 วงจรรองความถี่ผ่าน 400 MHz	46
3.14 วงจรรองความถี่ผ่านหลังจากการกำหนด โหนด	47
3.15 วงจร Transmission Line	47
3.16 บล็อกไดอะแกรมการทำงานของโปรแกรมควบคุมการบันทึกสัญญาณบีคอน และควบคุมการคำนวณค่าเฟส	49
3.17 บล็อกการรับและบันทึกสัญญาณความถี่ 150 MHz	50
3.18 บล็อกการรับและบันทึกสัญญาณ 400 MHz	52
3.19 บล็อกการทำงานของ WBX Board	52
3.20 บล็อกการทำงานของ WBX Board ในการ Demodulation IQ Signal	53
3.21 โฟลว์ชาร์ตภาพรวมการทำงานของโปรแกรม	55
3.22 โฟลว์ชาร์ตการทำงานของโปรแกรมหลัก	56
3.23 โฟลว์ชาร์ตแสดงการทำงานฟังก์ชัน save_date(enter,exit,satname)	57
3.24 โฟลว์ชาร์ตแสดงการทำงานของคลาส wfm_rx_block()	58
3.25 โฟลว์ชาร์ตแสดงการทำงานผ่านGUI	59
3.26 บล็อกไดอะแกรมขั้นตอนการคำนวณผลต่างเฟสของสัญญาณบีคอน	60
3.27 โฟลว์ชาร์ตโปรแกรมคำนวณค่าผลต่างเฟสของสัญญาณบีคอน	62

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ (ต่อ)

รูปที่	หน้า
4.1	65
4.2	66
4.3	66
4.4	67
4.5	67
4.6	68
4.7	68
4.8	69
4.9	69
4.10	70
4.11	70
4.12	71
4.13	71
4.14	72
4.15	72
4.16	73
4.17	73
4.18	74
4.19	74

สารบัญรูปภาพ (ต่อ)

รูปที่	หน้า	
4.20	ค่า SWR ของวงจรรองช่วงความถี่ผ่าน 400 MHz ด้าน Input	75
4.21	ค่า SWR ของวงจรรองช่วงความถี่ผ่าน 400 MHz ด้าน Output	75
4.22	ผลตอบแทนทางความถี่ของวงจรรองความถี่ผ่าน 400 MHz	76
4.23	วงจรรองความถี่ 150 MHz (ปรับปรุง)	76
4.24	ค่าอัตราการขยายในการส่งผ่าน (S_{21}) ของวงจรรองความถี่ผ่าน 150 MHz	77
4.25	ค่าอัตราการสูญเสียเนื่องจากการสะท้อน (Log Mag) ของวงจรรองความถี่ผ่าน 150 MHz ด้าน Input	77
4.26	ค่าอัตราการสูญเสียเนื่องจากการสะท้อน (Log Mag) ของวงจรรองความถี่ผ่าน 150 MHz ด้าน Output	78
4.27	ค่า SWR ของวงจรรองช่วงความถี่ผ่าน 150 MHz ด้าน Input	78
4.28	ค่า SWR ของวงจรรองช่วงความถี่ผ่าน 150 MHz ด้าน Output	79
4.29	ผลตอบแทนทางความถี่ของวงจรรองความถี่ผ่าน 150 MHz (ปรับปรุง)	79
4.30	วงจรรองความถี่ 400 MHz (ปรับปรุง)	80
4.31	ค่าอัตราการขยายในการส่งผ่าน (S_{21}) ของวงจรรองความถี่ผ่าน 400 MHz	80
4.32	ค่าอัตราการสูญเสียเนื่องจากการสะท้อน (Log Mag) ของวงจรรองความถี่ผ่าน 400 MHz ด้าน Input	81
4.33	ค่าอัตราการสูญเสียเนื่องจากการสะท้อน (Log Mag) ของวงจรรองความถี่ผ่าน 400 MHz ด้าน Output	81
4.34	ค่า SWR ของวงจรรองช่วงความถี่ผ่าน 400 MHz ด้าน Input	82
4.35	ค่า SWR ของวงจรรองช่วงความถี่ผ่าน 400 MHz ด้าน Output	82
4.36	ผลตอบแทนทางความถี่ของวงจรรองความถี่ผ่าน 400 MHz (ปรับปรุง)	83
4.37	ค่า TLE ที่ได้จากเว็บไซต์	84
4.38	(ก) คอมพิวเตอร์เชื่อมต่ออินเทอร์เน็ต (ข) คอมพิวเตอร์ไม่เชื่อมต่ออินเทอร์เน็ต	84
4.39	แสดงข้อมูลการโคจรผ่านของดาวเทียม	85
4.40	ค่าเวลาที่ได้จากโปรแกรมก่อนและหลังเรียงลำดับ	86
4.41	แสดงค่าเวลาที่เริ่มทำการบันทึกสัญญาณจริง	86
4.42	สถานะเครื่องบันทึกสัญญาณ	87

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปลภาพ (ต่อ)

รูปที่	หน้า
4.43 สัญญาณที่บันทึกได้เมื่อทำการป้อนสัญญาณความถี่ 150 MHz	87
4.44 สัญญาณที่บันทึกได้เมื่อทำการป้อนสัญญาณความถี่ 400 MHz	88
4.45 อินเตอร์เฟส โปรแกรมคำนวณการโคจรผ่านของดาวเทียม	88
4.46 กำหนดค่าให้กับ โปรแกรมคำนวณการโคจรผ่านของดาวเทียม	89
4.47 กล่องข้อความเตือนการกรองข้อมูลเกินขีด	90
4.48 กล่องข้อความเตือนการใส่ข้อมูลวัน เวลา ที่ไม่เป็นจริง	90
4.49 (ก)เพิ่มข้อมูลตำแหน่งที่เราสนใจ (ข) เพิ่มวัน เวลา ช่วงเวลาที่ทำการสังเกต	91
4.50 ไฟล์ที่ทำการคำนวณการโคจรผ่านของดาวเทียม	91
4.51 รูปสัญญาณที่พล็อตจากไฟล์ข้อมูลบางส่วนของสัญญาณบีคอนความถี่ 400 MHz	93
4.52 แสดงรูปสัญญาณที่พล็อตจากไฟล์ข้อมูลบางส่วนของสัญญาณบีคอนความถี่ 150 MHz	93
4.53 สเปกตรัมกำลังสัญญาณบีคอนที่การแปลง FFT ครั้งที่ 50000 ความถี่ 150MHz	94
4.54 สเปกตรัมทางเฟสของสัญญาณบีคอนที่การแปลง FFT ครั้งที่ 50000 ความถี่ 150MHz	94
4.55 สเปกตรัมทางเฟสของสัญญาณบีคอนที่การแปลง FFT ครั้งที่ 50000 ความถี่ 400MHz	95
4.56 สเปกตรัมทางเฟสของสัญญาณบีคอนที่การแปลง FFT ครั้งที่ 50000 ความถี่ 400MHz	95
4.57 สเปกตรัมกำลังสัญญาณบีคอนที่การแปลง FFT ครั้งที่ 125000 ความถี่ 150MHz	96
4.58 สเปกตรัมทางเฟสของสัญญาณบีคอนที่การแปลง FFT ครั้งที่ 125000 ความถี่ 150MHz	96
4.59 สเปกตรัมกำลังสัญญาณบีคอนที่การแปลง FFT ครั้งที่ 125000 ความถี่ 400MHz	97
4.60 สเปกตรัมทางเฟสของสัญญาณบีคอนที่การแปลง FFT ครั้งที่ 125000 ความถี่ 400MHz	97

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ (ต่อ)

รูปที่		หน้า
4.61	สเปกตรัมกำลังสัญญาณบิตอนที่การแปลง FFT ครั้งที่ 200000 ความถี่ 150MHz	98
4.62	สเปกตรัมทางเฟสของสัญญาณบิตอนที่การแปลง FFT ครั้งที่ 200000 ความถี่ 150MHz	98
4.63	สเปกตรัมกำลังสัญญาณบิตอนที่การแปลง FFT ครั้งที่ 200000 ความถี่ 400MHz	99
4.64	สเปกตรัมทางเฟสของสัญญาณบิตอนที่การแปลง FFT ครั้งที่ 200000 ความถี่ 400MHz	99
4.65	ค่าผลต่างเฟสของสัญญาณบิตอน 150MHz และ 400MHz เปลี่ยนแปลงตามเวลา	100
4.66	กราฟเปรียบเทียบค่า TEC ที่คำนวณได้ระหว่าง KMITL และ Kyoto University	101



สารบัญตาราง

ตารางที่	หน้า
2.1 Butterworth Capacitive Coupled Resonators	20
2.2 0.01-dB Chebyshev Capacitive Couple Resonators	20
2.3 0.1-dB Chebyshev Capacitive Couple Resonators	21
2.4 0.5-dB Chebyshev Capacitive Couple Resonators	21
2.5 1-dB Chebyshev Capacitive Couple Resonators	21
2.6 Bessel Capacitive Couple Resonators	22
2.7 Linear Phase With Equiripple Error of 0.05° Capacitive Coupled Resonators	22
2.8 Linear Phase With Equiripple Error of 0.5° Capacitive Coupled Resonators	22
2.9 Transitional Gaussian to 6-dB Capacitive Coupled Resonators	23
2.10 Transitional Gaussian to 6-dB Capacitive Coupled Resonators	23
2.11 ตัวอย่าง Two Line Element ปรรที่คที่ 1	30
2.12 ตัวอย่าง Two Line Element ปรรที่คที่ 2	31
3.1 Butterworth Capacitive Coupled Resonators	37
3.2 Butterworth Capacitive Coupled Resonators	41

บทที่ 1

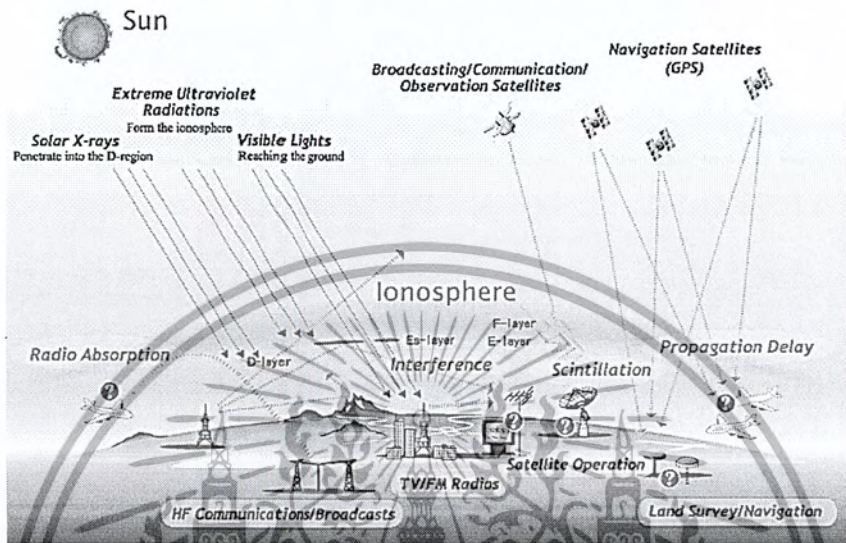
บทนำ

1.1 หลักการและเหตุผลของโครงการวิจัย

ชั้นบรรยากาศไอโอโนสเฟียร์ (Ionosphere) เป็นชั้นบรรยากาศที่มีความสูงประมาณ 80 กิโลเมตรถึง 1200 กิโลเมตรจากพื้นโลกเป็นชั้นบรรยากาศที่เกิดการไอออไนส์ (Ionized) มีปริมาณไอออนและอิเล็กตรอนจำนวนมากซึ่งความแปรปรวนของประจุเหล่านี้ก่อให้เกิดผลกระทบต่อระบบสื่อสาร เมื่อคลื่นสัญญาณเดินทางผ่านชั้นบรรยากาศไอโอโนสเฟียร์ ทำให้สัญญาณดาวเทียมมีสมรรถนะต่ำลงหรือในกรณีที่เลวร้ายที่สุดสัญญาณอาจขาดหายได้ (Loss-of-lock) ดังรูปที่ 1.1 ตัวแปรหนึ่งที่น่าสนใจในการศึกษาการเปลี่ยนแปลงของดัชนีหักเหในชั้นบรรยากาศไอโอโนสเฟียร์ได้แก่ ปริมาณอิเล็กตรอนสุทธิ (Total Electron Content) หรือ TEC ซึ่งการจะหาค่า TEC ได้นั้นจำเป็นต้องหาค่าผลต่างเฟสขึ้นมาก่อน ซึ่งสามารถหาได้โดยใช้ประโยชน์จากระบบเครื่องรับสัญญาณดาวเทียม โดยในปัจจุบันการวัดค่า TEC ทำได้โดยการติดตั้งเครื่องรับสัญญาณบีคอนแอนะล็อก (Analog beacon receiver) โดยส่วนใหญ่พัฒนาสำหรับความถี่เท่ากับ 150 MHz และ 400 MHz ซึ่งบางรุ่นก็มีจำหน่ายในท้องตลาด เครื่องรับแอนะล็อกจะรับสัญญาณจากดาวเทียมวงจรรต่ำ (Low-Earth Orbit Satellite, LEO) จำนวน 2 ถึง 3 ความถี่และทำการวิเคราะห์ผลต่างของเฟส แต่ระบบการวัด TEC ด้วยเครื่องมือทั้งสองนี้มีราคาสูงมากในระดับ 1 ล้านบาทต่อระบบ (ฮาร์ดแวร์และซอฟต์แวร์)

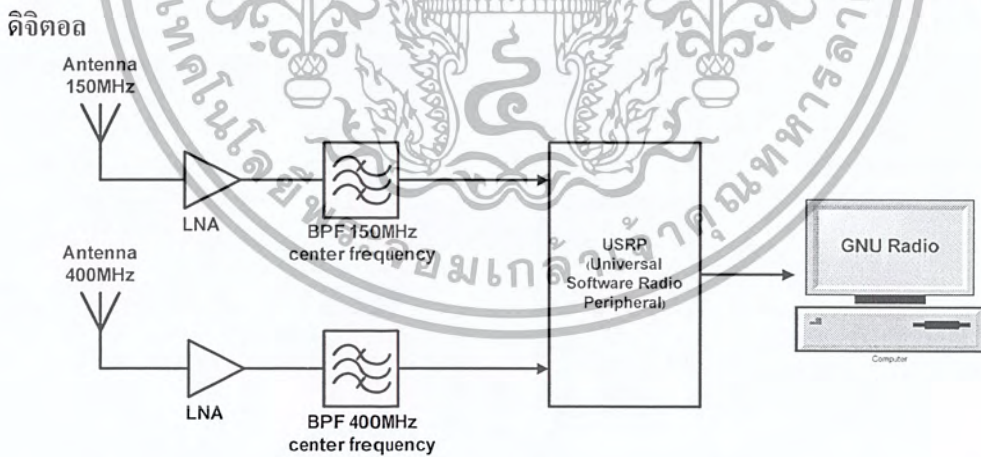
ดังนั้นผู้จัดทำจึงมีแนวคิดในการพัฒนาระบบคำนวณค่าความต่างเฟสและ TEC ของสัญญาณดาวเทียมอันเนื่องมาจากผลกระทบของชั้นบรรยากาศ โดยระบบดังกล่าวจะประกอบไปด้วย ส่วน RF Front-End วงจรกรองความถี่แถบผ่าน เครื่องรับสัญญาณดาวเทียมแบบดิจิทัล และซอฟต์แวร์ที่ใช้ในการคำนวณค่า TEC ในชั้นบรรยากาศในราคาที่ประหยัดกว่า

Ionospheric Effects on Radio Applications



รูปที่ 1.1 ผลกระทบจากชั้นบรรยากาศไอโอโนสเฟียร์ต่อคลื่นวิทยุ

1.1.1 โครงสร้างพื้นฐานการทำงานของเครื่องรับสัญญาณดาวเทียมบีคอนแบบ



รูปที่ 1.2 บล็อกไดอะแกรมแสดงระบบเครื่องรับสัญญาณดาวเทียมบีคอนแบบดิจิทัล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 1.2 บล็อกไดอะแกรมแสดงระบบเครื่องรับสัญญาณดาวเทียมบีคอนแบบดิจิทัล โดยสัญญาณดาวเทียมบีคอนความถี่ 150MHz และ 400MHz จากดาวเทียมจะถูกรับโดยสายอากาศ จากนั้นสัญญาณจะผ่านภาคขยาย ไปยัง วงจรกรองความถี่แถบผ่านย่าน 150MHz และ 400 MHz จากนั้นสัญญาณที่ผ่านวงจรกรองแล้วจะถูกแปลงจากสัญญาณแอนะล็อก เป็น ดิจิตอล และ ถูกย้ายย่านความถี่เป็นเบสแบนด์ โดยชุดเครื่องมือที่เรียกว่า USRP และส่งข้อมูลผ่านสาย USB2.0 ไปยังคอมพิวเตอร์เพื่อทำการการประมวลผลสัญญาณ ในการควบคุมการทำงานของ USRP จะใช้ GNU Radio ส่วนการคำนวณหาผลต่างเฟสจะใช้ Python

1.2 วัตถุประสงค์

1. ออกแบบวงจรกรองแถบความถี่ผ่าน
2. ออกแบบโปรแกรมควบคุมอุปกรณ์ USRP ในการรับและบันทึกสัญญาณบีคอน
3. ออกแบบโปรแกรมคำนวณความต่างเฟสของสัญญาณบีคอน ด้วย Software Defined Radio
4. ออกแบบระบบการทำนายการโคจรของดาวเทียมที่โคจรผ่านตำแหน่งที่สนใจในแต่ละวัน

1.3 ขอบเขตของการทำโครงการ

1. ออกแบบและสร้างวงจรกรองแถบความถี่ผ่าน 150 MHz และ 400 MHz โดยมีแบนวิดท์ 10 MHz และ 27 MHz ตามลำดับ
2. ออกแบบโปรแกรมที่ใช้ในการคำนวณหาค่าความต่างเฟส โดยประยุกต์ใช้กับชุดพัฒนา USRP ร่วมกับ ซอฟต์แวร์ GNU Radio

บทที่ 2

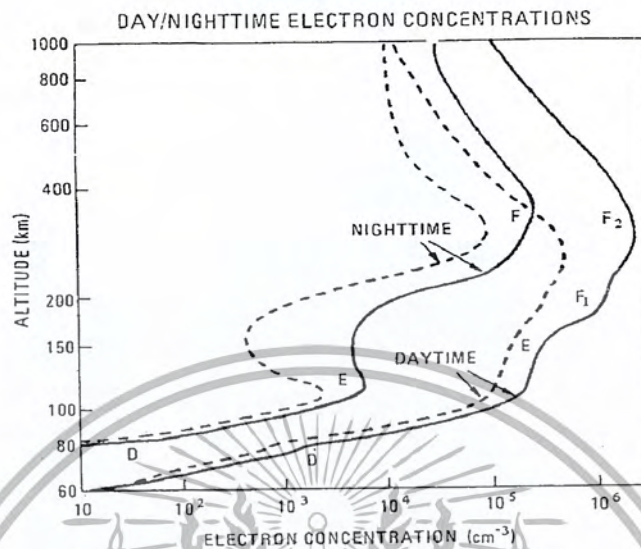
ทฤษฎีและหลักการที่เกี่ยวข้อง

2.1 ปริมาณอิเล็กตรอนในชั้นบรรยากาศไอโอโนสเฟียร์

ชั้นบรรยากาศไอโอโนสเฟียร์ (Ionosphere) ดังรูปที่ 2.1 เป็นชั้นบรรยากาศที่อยู่สูงกว่าชั้นมีโซสเฟียร์ (Mesosphere) ที่ชั้นบรรยากาศไอโอโนสเฟียร์นี้มีอิเล็กตรอนอิสระอยู่เป็นจำนวนมาก เป็นอิเล็กตรอนที่หลุดออกมาจากอะตอมของโมเลกุลและไอออนที่เป็นประจุบวกแสงอัลตราไวโอเล็ตจากดวงอาทิตย์คือตัวการที่ทำให้อิเล็กตรอนแตกตัวจากอะตอมของออกซิเจนโดยบริเวณขอบเขตของชั้นบรรยากาศไอโอโนสเฟียร์เริ่มต้นที่ความสูงประมาณ 60 กิโลเมตร บริเวณที่เกิดปรากฏการณ์ไอออไนเซชัน (Ionization) มากที่สุดคือที่ระดับความสูงประมาณ 300 กิโลเมตร

เนื่องจากองค์ประกอบของชั้นบรรยากาศมีการเปลี่ยนแปลงตามความสูง อัตราการเกิดไอออนจึงเปลี่ยนแปลงไปทำให้เกิดช่วงความแรงของการไอออไนเซชันเป็นช่วงๆ เรียกว่าเลเยอร์ "D", "E", "F1", และ "F2" โดยเฉพาะ เลเยอร์ F อยู่ที่ความสูงตั้งแต่ 150 กิโลเมตรขึ้นไป เป็นเลเยอร์ที่มีผลกระทบต่อการหักเหคลื่นวิทยุที่ใช้ในการสื่อสาร เลเยอร์ในชั้นนี้แบ่งออกได้เป็น F1 และ F2 ไอออนในบริเวณนี้ส่วนใหญ่เป็น NO^+ และ O^+

ปัจจุบันการสื่อสารดาวเทียมเติบโตอย่างรวดเร็ว การแพร่สัญญาณภาพโทรทัศน์และวิทยุ โทรศัพททางไกล โทรศัพทเซลลูลาร์ ได้อาศัยสัญญาณดาวเทียมเป็นสถานีฐานในการส่งสัญญาณ ระบบการสื่อสารหลายระบบต้องอาศัยชั้นบรรยากาศไอโอโนสเฟียร์เพื่อสะท้อนคลื่นวิทยุให้เดินทางไปได้ไกล พายุไอโอโนสเฟียร์ทำให้การสื่อสารวิทยุบางย่านความถี่เสียหายได้ และทำให้เกิดการเปลี่ยนแปลงระดับสัญญาณอย่างกะทันหัน ทำให้เกิดการเดินทางของคลื่นที่เปลี่ยนแปลงไป



รูปที่ 2.1 ชั้นบรรยากาศไอโอโนสเฟียร์

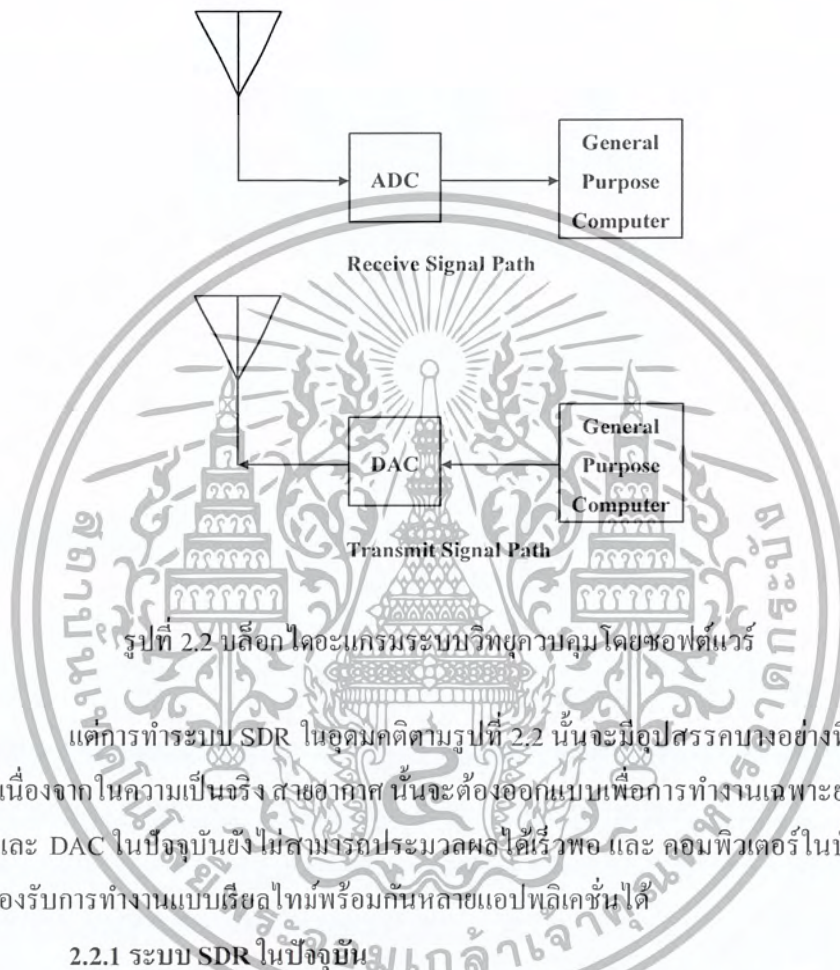
2.2 วิทยุควบคุมด้วยซอฟต์แวร์ (Software-Defined Radio)

วิทยุควบคุมด้วยซอฟต์แวร์ (Software-Defined Radio) หรือเรียกโดยย่อว่า SDR คือระบบสื่อสารทางวิทยุที่สามารถทำการมอดูเลต (Modulation) และการดีมอดูเลต (Demodulation) สัญญาณวิทยุได้ด้วยซอฟต์แวร์ (Software) โดยมีแนวคิดที่ว่า ซอฟต์แวร์ควรทำงานเข้าใกล้กับสายอากาศให้มากที่สุด และซอฟต์แวร์นั้นจะต้องทำงานบนคอมพิวเตอร์ทั่วไปได้ด้วย ซึ่งแสดงได้ดังบล็อกไดอะแกรมดังรูปที่ 2.2

จากแนวคิดข้างต้นนี้ตามอุดมคติ สายอากาศ, ADC/DAC (Analog to Digital Converter/Digital to Analog Converter) และคอมพิวเตอร์นั้นต้องสามารถที่จะประมวลผลสัญญาณวิทยุที่แบบเรียลไทม์ (Real time) ได้

โดยงานวิจัยทางด้าน SDR นั้นต้องการที่จะเปลี่ยนปัญหาด้านวิศวกรรมวิทยุทางด้านฮาร์ดแวร์ (Hardware) ไปยังด้านของซอฟต์แวร์ (Software) ซึ่งข้อดีของการทำแบบนี้คือซอฟต์แวร์นั้นมีกระบวนการทำงานที่มีความยืดหยุ่น คาดคะเนผลได้ ทำซ้ำได้ และเข้าถึงได้มากกว่าทางด้านฮาร์ดแวร์ ด้านซอฟต์แวร์นั้น ระบบวิทยุนั้นจะถูกสร้างขึ้นโดยซอฟต์แวร์ เพราะฉะนั้น ระบบ SDR โดยลำพังนั้นสามารถเรียกได้ว่าเป็นหนึ่งในอุปกรณ์ รับ-ส่ง สัญญาณ

คลื่นวิทยุได้ (RF Transceiver) เพียงแค่สั่งให้โปรแกรมที่สร้างขึ้นมาจากบล็อกของโปรแกรมต่างๆในหน่วยความจำทำงาน



แต่การทำระบบ SDR ในอุดมคติตามรูปที่ 2.2 นั้นจะมีอุปสรรคบางอย่างที่จัดการได้ยากซึ่งเนื่องจากในความเป็นจริง สัญญาณที่รับมาจะต้องถูกแปลงเป็นสัญญาณความถี่ต่ำก่อน ADC และ DAC ในปัจจุบันยังไม่สามารถประมวลผลได้เร็วพอ และ คอมพิวเตอร์ในปัจจุบันนั้นยังไม่รองรับการทำงานแบบเรียลไทม์พร้อมกันหลายแอปพลิเคชันได้

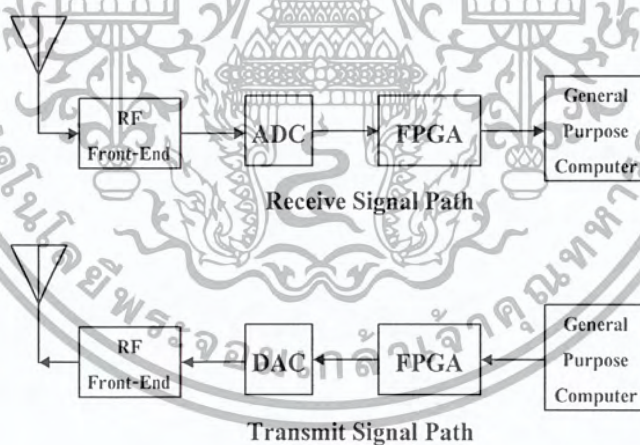
2.2.1 ระบบ SDR ในปัจจุบัน

ระบบ SDR จากรูปที่ 2.2 นั้นแสดงถึงระบบทั่วไปที่เป็นการประยุกต์ใช้งานอย่างง่ายและความถี่ไม่สูงมากนัก เช่น FM และ AM เท่านั้น อย่างไรก็ตามการประยุกต์ใช้งานที่นอกเหนือจากนี้จะมีองค์ประกอบอื่นเพิ่มเติมขึ้นมาคือ FPGAs (Field Programmable Gate Arrays) และ Super Heterodyne Mixing ที่เรียกว่า RF Front End เพื่อเพิ่มความสามารถของระบบให้รองรับความต้องการที่สูงขึ้นได้ แสดงดังรูปที่ 2.3

ในส่วนของภาครับนั้น ส่วน RF Front End จะทำการแปลงสัญญาณที่รับได้ที่ความถี่คลื่นพาห้ (Carrier) แปลงไปสู่ความถี่ย่าน IF (Intermediate Frequency) หรือ เบสแบนด์

(Baseband) ส่วนภาคส่งนั้น RF Front End ทำการย้ายความถี่ย่าน IF หรือ เบสแบนด์ ไปยังความถี่คลื่นพาห้ที่ต้องการในงานที่ใช้อัตราการส่งข้อมูลต่ำ ADC/DAC จะช่วยในการลดภาระการคำนวณของคอมพิวเตอร์ อย่างไรก็ตาม โดยปกติแล้วการประมวลผลยังต้องการประสิทธิภาพที่สูง ซึ่งการจะลดภาระของการคำนวณของคอมพิวเตอร์นั้นยังสามารถทำได้อีก โดยการให้ FPGA นั้นทำงานอยู่ระหว่าง ADC/DAC และ คอมพิวเตอร์ ซึ่งชิพ FPGA จะทำงานในส่วนการประมวลผลสัญญาณในระดับสูง ซึ่งใช้ปกติในระบบ SDR เช่น ในการทำ DDC/DUC (Digital Up/Down Conversion) และ วงจรกรอง (Filter)

บล็อกไดอะแกรมที่เพิ่มขึ้นมาในรูปที่ 2.3 แสดงถึงระบบ SDR ในปัจจุบัน ซึ่งระบบนี้มีความยืดหยุ่นไม่มากนักเหมือนระบบ SDR ในอุดมคติแต่ FPGA ที่ใช้กันทั่วไปนั้นไม่ได้มีผลกระทบต่อความยืดหยุ่นของระบบโดยรวมเนื่องจากมีฟังก์ชันที่เอาไว้อยู่ทั่วไปในระบบวิทยุแต่อย่างไรก็ตาม RF Front End ทั้งหมด (รวมถึงสายอากาศด้วย) นั้นมีข้อจำกัดในการทำงานภายในย่านความถี่ที่ออกแบบเท่านั้น ดังนั้นรูปแบบของวิทยุ SDR ที่เพิ่มในส่วนของ RF Front End นั้นจะถูกจำกัดในส่วนของสัญญาณเฉพาะย่านความถี่ที่ RF Front End ทำงานได้เท่านั้น



รูปที่ 2.3 บล็อกไดอะแกรมแสดงการใช้งานระบบการสื่อสารแบบ SDR ในปัจจุบัน

2.2.2 ระบบวิทยุควบคุมซอฟต์แวร์แบบโอเพนซอร์ส (Open source SDR)

GNU Radio คือซอฟต์แวร์ที่เป็นโอเพนซอร์ส ซึ่งคิดค้นโดย Mr.Eric Blossom ในปัจจุบันงานทางด้าน GNU Radio นั้นได้มีการพัฒนาขึ้นเป็นรูปแบบของสถาปัตยกรรม โดยใน

การสร้างระบบ SDR รูปแบบต่างๆ ซอฟต์แวร์นั้นจะประกอบด้วย ส่วนของฮาร์ดแวร์ที่ไม่ขึ้นกับการโปรแกรมการประมวลผลสัญญาณ และ ฮาร์ดแวร์ที่ขึ้นกับการโปรแกรม ซึ่งรวมไปถึงการเชื่อมต่อระหว่างโปรแกรมการประมวลผลสัญญาณที่ทำงานบนเครื่องคอมพิวเตอร์ทั่วไป กับ ส่วนของฮาร์ดแวร์ (ADC/DAC หรือ FPGA)

ในจำนวนของฮาร์ดแวร์ในปัจจุบันที่ใช้เชื่อมต่อกับ GNU Radio นั้นมีทั้ง การ์ดเสียง, TV Tuner และ ADC ประสิทธิภาพสูง ของเครื่องคอมพิวเตอร์ที่ใช้งานนั้นๆด้วย อย่างไรก็ตาม อุปกรณ์ฮาร์ดแวร์ที่นิยมใช้ในการเชื่อมต่อกับ GNU Radio นั้น คือ Universal Software Radio Peripheral (USRP) ซึ่ง USRP มีการทำงานตามบล็อกไดอะแกรมที่แสดงในรูปที่ 2.3 คือ บล็อกไดอะแกรมที่เป็นรูปแบบของ SDR เมื่อซอฟต์แวร์ GNU Radio ทำงานบนเครื่องคอมพิวเตอร์และทำงานเชื่อมต่อกับ USRP

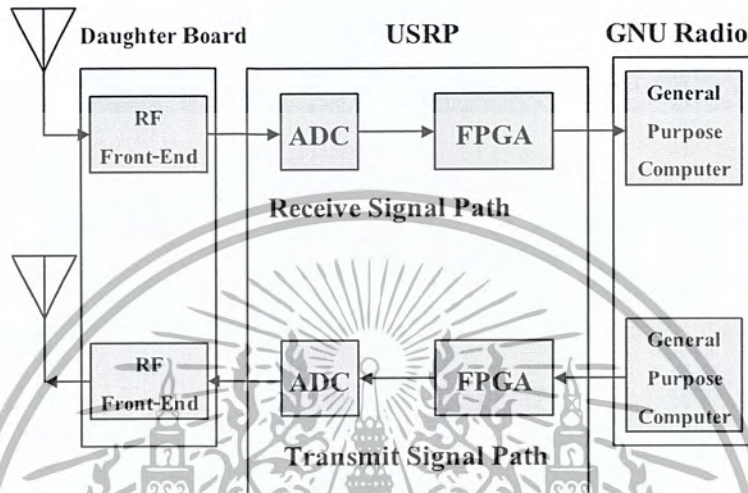
2.3 The Universal Software Radio Peripheral (USRP)

USRP (Universal Software Radio Peripheral) ถูกออกแบบและสร้างโดย Mr. Matt Ettus โดยในปัจจุบันนี้ USRP ได้มีการปรับปรุง มาถึง สี่ครั้งและสามารถสั่งซื้อได้ที่ Ettus Reserch , LLC

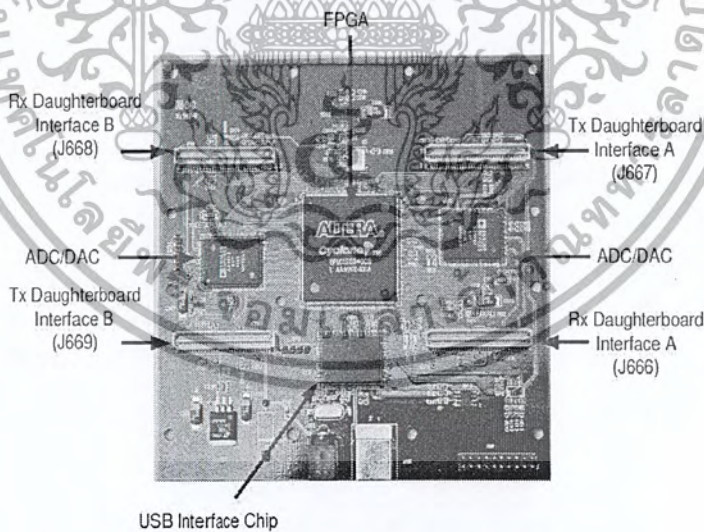
โดยที่ USRP มีส่วนการเชื่อมต่อทางอิเล็กทรอนิกส์และทางกลกับ RF Front End (Daughter board) ซึ่งจะแปลงความถี่เป็นย่าน IF หรือ Baseband จากความถี่ย่าน RF ที่สนใจ จากรูปที่ 2.4 แสดงถึงโครงสร้างในรูปที่ 2.3 ที่แทนได้โดยโครงสร้างของ GNU Radio และ USRP ที่ทำงานร่วมกับ Daughter Board ซึ่ง USRP จะเป็นตัวกำหนดฟังก์ชัน ADC/DAC และ FPGA โดยที่ Daughter Board นั้นทำหน้าที่ในส่วนของ RF Front-End แสดงดังรูปที่ 2.4 ซึ่งจุดเชื่อมต่อ Daughter Board นั้นแสดงดังรูปที่ 2.5

การออกแบบ USRP ร่วมกับ Daughter Board นั้น พัฒนาขึ้นภายใต้โครงการเปิด โดยที่สามารถดาวน์โหลดข้อมูลทุกอย่างเกี่ยวกับอุปกรณ์นี้ได้และสามารถดัดแปลงได้โดยไม่ผิดกฎหมายหลักการออกแบบ USRP คือ การประมวลผลสัญญาณ โดยเฉพาะ เช่น การมอดูเลชันและดีมอดูเลชันนั้นจะกระทำที่ CPU ของเครื่องคอมพิวเตอร์ ส่วนการประมวลผลความเร็วสูงทั่วไป เช่น Digital Up/Down Conversion (DDC/DUC), Decimation, Interpolation นั้นจะกระทำบน

ชิพ FPGA ในรูปที่ 2.7 แสดงเส้นทางการรับ-ส่งสัญญาณของ USRP โดยเปรียบเทียบกับรูปที่ 2.7 กับ USRP ของจริงที่แสดงดังรูปที่ 2.6

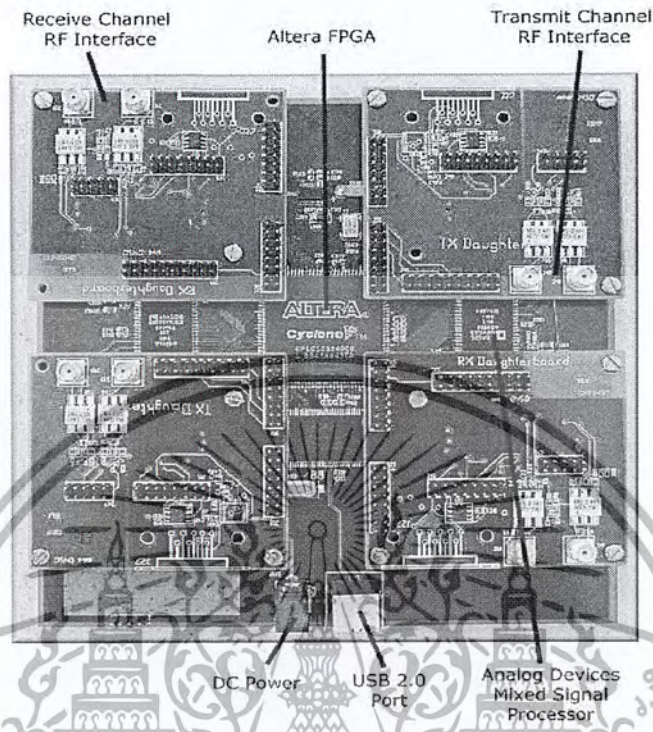


รูปที่ 2.4 บล็อกไดอะแกรมระบบ SDR โดยใช้ GNU Radio และ USRP



รูปที่ 2.5 USRP Motherboard โดยไม่รวม Daughter Board

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.6 USRP Motherboard กับ Basic Transmit และ Receive Daughter Board

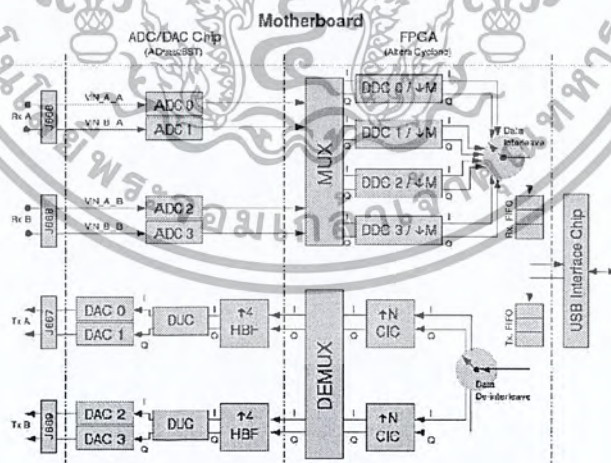
2.3.1 เส้นทางารรับข้อมูล (Signal receive path)

USRP นั้นมีสล็อต 2 สล็อตที่ใช้รับสัญญาณที่ Daughter Board โดยสล็อตนี้มีชื่อเรียกเป็น RXA และ RXB ซึ่งก็คือพอร์ต (Port) เบอร์ J666 และ J668 ตามลำดับ โดยในแต่ละพอร์ตนั้นรับค่าจริง 2 ค่าที่เป็นค่าสัญญาณแรงดันเข้ามาจาก Daughter Board ซึ่งสัญญาณนี้จะกำหนดเป็น VIN_A_X และ VIN_B_X โดยที่ X จะถูกแทนด้วย A หรือ B ขึ้นอยู่กับ สล็อตฝั่งใดที่รับสัญญาณมา (เช่น RxA หรือ RxB) จากนั้นสัญญาณแอนะล็อก VIN_A และ VIN_B จะถูกส่งไปยัง ADC สองตัวที่มีค่าความละเอียด 12 Bits โดยมีอัตราแซมปลิงเท่ากับ 64 MegaSamples/s หลังจากนั้นค่า แซมเปิล ความละเอียด 12 Bits จะถูกส่งไปยัง FPGA เพื่อประมวลผล ในการเข้าไปยัง FPGA สัญญาณที่ถูกแปลงเป็นดิจิทัล จะมีการสร้างเส้นทางโดยตัว Multiplexer หรือ MUX เพื่อจัดสรรสัญญาณให้ DDC ที่มีทั้งหมดสี่ตัว

DDC นั้นจำเป็นต้องใช้ในส่วนของ Complex Mixing โดยจะทำการกำหนดค่า In-Phase และ Quadrature อินพุตซึ่งจะขึ้นอยู่กับผู้ใช้ในการกำหนด VIN_A_A , VIN_B_A ,

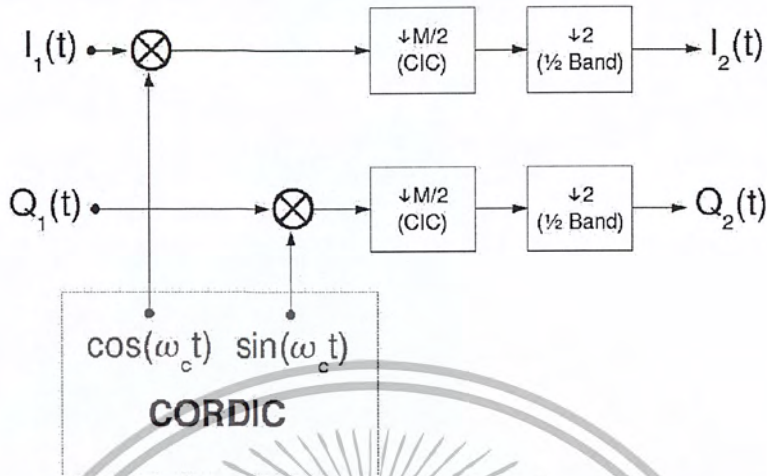
VIN_A_B , VIN_B_B ในการสร้างเส้นทางไปยัง In-Phase และ Quadrature พอร์ตของ DDC ทั้งสี่ตัวแต่ละ DDC จะแปลงสัญญาณค่าอินพุตไปเป็นสัญญาณ Complex Baseband ซึ่งเมื่อทำการ Down-Convert แล้วสัญญาณจะถูก Decimated ด้วยค่าแฟคเตอร์ที่กำหนดโดยผู้ใช้โดยการ Decimation จะกระทำในสองส่วน โดยให้ M คือค่า Decimation กำหนดโดยผู้ใช้สัญญาณนั้นในตอนแรกจะถูก Decimated ด้วยแฟคเตอร์ $M/2$ โดยใช้วงจรกรอง CIC และ Decimation ครั้งสุดท้ายด้วยค่า 2 ซึ่งจะทำโดย วงจรกรอง Half-Band ในส่วนของ DDC และ Decimation นั้นแสดงดังรูปที่ 2.8

ทั้ง In-Phase และ Quadrature เอาท์พุต ของแต่ละตัวกรอง Half-Band นั้นถ้ากำหนดโดยผู้ใช้นั้นจะแทรกสลับกันและส่งไปยัง FIFO Buffer ซึ่งค่าแชนเนลของข้อมูลเหล่านี้จะส่งไปยังชิพเชื่อมต่อ USB2.0 กับคอมพิวเตอร์ผู้ใช้เพื่อการประมวลผลต่อไป ในรูปที่ 2.8 แสดง DDC/Decimation ทั้ง 4 ส่วน ซึ่งใช้งานแค่สองตัวจากทั้งหมด นั้นหมายความว่าผู้ใช้สามารถที่จะทำการรับข้อมูลจากทั้ง RXA และ RXB (2 Channels) ของแต่ละค่า Complex แชนเนล โดยใช้ 32 Bits (16 Bits สำหรับ In-Phase และ 16 Bits สำหรับ Quadrature Values) ซึ่ง USB 2.0 ชิปนั้นสามารถที่รับส่งข้อมูลได้สูงสุด 32MBytes/s นั้นทำให้จำกัดค่าแบนด์วิดท์ของสัญญาณที่ส่งไปยังเครื่องคอมพิวเตอร์



รูปที่ 2.7 เส้นทางการรับ-ส่ง สัญญาณ โดยการทำงานของ DDC และ DUC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.8 Digital Down-Conversion (DDC) และ Decimation ในการรับสัญญาณของ USRP



รูปที่ 2.9 Digital Up-Conversion (DUC) ในเส้นทางส่งสัญญาณของ USRP

2.3.2 อุปกรณ์เสริมในการรับสัญญาณ (Daughter Boards)

Daughter Boards ที่มีอยู่ในปัจจุบันจากบริษัท Ettus Research นั้น ถ้าเป็น Basic Boards นั้นจะไม่มีส่วนของ Tuning หรือส่วนขยาย (Amplifier) ส่วนบอร์ดอื่นๆ นั้น จะมีส่วนของ Tuning และส่วนขยายรวมอยู่ด้วยซึ่ง Daughter Board ที่ทางบริษัท Ettus ได้จัดทำและจำหน่ายในปัจจุบันจะมีทั้งสี่ชิ้นนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- BasicTX : 1 MHz – 250 MHz Transmitter
- BasicRX : 1 MHz – 250 MHz Receiver
- LFTX : DC – 30 MHz Transmitter
- LFRX : DC – 30 MHz Receiver
- TVRX : 50 MHz – 860 MHz Receiver
- DBSRX : 800 MHz – 2.4 GHz Receiver
- RFX400: 400 – 500 MHz Transceiver, Transmit Power: 200 mW (23dBm)
- RFX900: 750 – 1050 MHz Transceiver, Transmit Power: 200 mW (23dBm)
- RFX1200: 1150 – 1450 MHz Transceiver, Transmit Power: 200 mW (23dBm)
- RFX1800: 1.5 – 2.1 GHz Transceiver, Transmit Power: 100 mW (20dBm)
- RFX2400: 2.3 – 2.9 GHz Transceiver, Transmit Power: 50 mW (17dBm)
- XCVR2450: 2.4 – 2.5 GHz and 4.9 – 5.9 GHz Transceiver, Transmit Power: 100 mW (20dBm)
- WBX : 50 MHz – 2.2 GHz Transceiver, Transmit Power: 100 mW (20dBm)

2.4 GNU Radio

GNU Radio คือ Open Source Software Framework สำหรับสร้างงานทางด้าน SDR สามารถใช้งานได้กับระบบปฏิบัติการ Linux , Unix , Windows หรือ Mac OS X โดย GNU Radio เตรียมการประมวลผลสัญญาณของคอมพิวเตอร์เชื่อมต่อกับฮาร์ดแวร์และ Graphical User Interface และ Library ที่เป็นประโยชน์ โดย GNU Radio นั้น Flow graph ของข้อมูลจะถูกสร้างขึ้นจากโหนดเรียกว่า Block ซึ่งจะทำให้การเชื่อมต่อกับด้านที่เรียกว่า พอร์ต (Ports) ซึ่งแต่ละบล็อกจะแทนการประมวลผลสัญญาณดิจิทัลที่ต่างกันและการส่งผ่านข้อมูลระหว่างบล็อกนั้นทำได้โดยการเชื่อมต่อพอร์ตอินพุตของบล็อกหนึ่งกับพอร์ตเอาต์พุตของอีกบล็อกหนึ่ง ภายใต้โครงสร้างนี้สามารถสร้างบล็อกและเชื่อมต่อด้วยกราฟได้และเมื่อทุกบล็อกเชื่อมต่อกันหมด GNU Radio ก็จะเป็นตัวจัดการไหลของข้อมูลจากซอร์สบล็อก (Source Block) ของข้อมูลไปยังซิงค์บล็อก (Sink Block)

GNU Radio เป็นระบบไฮบริดจ์ โดยงานที่เป็นบล็อกของการประมวลผลสัญญาณดิจิทัลจะเขียนโดยภาษา C++ และส่วนที่เป็นการเชื่อมต่อกราฟและการจัดการทั่วไปนั้นจะใช้โปรแกรมภาษาไพธอน (Python) ซึ่งทำให้ผู้พัฒนาสามารถเขียนโปรแกรมที่เอาไปประมวลผลสัญญาณคุณภาพสูงด้วยภาษา C++ แต่ใช้ภาษาที่ง่ายในการสร้างแอปพลิเคชันด้วยไพธอน ใน GNU Radio นั้นจะใช้ภาษาไพธอนร่วมกับบล็อกการประมวลผลสัญญาณโดยภาษา C++ ไปยังจุดเชื่อมต่อจะถูกสร้างอัตโนมัติโดย SWIG (The Simple Wrapper Interface Generator) สำหรับไพธอนข้อมูลของ SWIG นั้นสามารถค้นหาได้ใน www.swig.org ข้อจำกัดของ GNU Radio นั้นไม่สามารถนำมาใช้ในทางด้านระบบสื่อสารสองทิศทางแบบอะซิงโครนัส (Two-way asynchronous communication systems) เช่น TDMA และงานด้าน Packet - Based System

2.5 วงจรกรองความถี่ (Filter)

วงจรกรองความถี่ (Filter) คืออุปกรณ์ที่ทำหน้าที่เป็นตัวเลือกผ่านสัญญาณ โดยวงจรกรองจะยอมให้สัญญาณเฉพาะย่านความถี่ผ่านไปได้เท่านั้น เราจะเรียกย่านความถี่ที่วงจรกรองยอมให้ผ่านว่าย่านความถี่ผ่าน (Passband) และย่านความถี่ที่วงจรกรองไม่ยอมให้ผ่านเรียกว่า ย่านความถี่หยุด (Stopband) เราสามารถแบ่งวงจรกรองออกเป็นสี่ชนิดได้ตามลักษณะของอุปกรณ์ที่ใช้ในการสร้างวงจรดังนี้

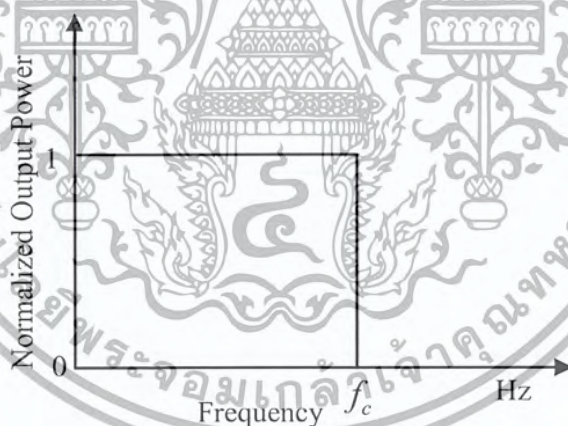
1. วงจรกรองพาสซีฟ (Passive Filter) เป็นวงจรกรองสัญญาณแบบต่อเนื่องที่ประกอบไปด้วยความต้านทาน ตัวเก็บประจุ และขดลวดเหนี่ยวนำ วงจรกรองประเภทนี้สามารถใช้ในการกรองสัญญาณที่มีความถี่ตั้งแต่สัญญาณไฟตรง (DC) ไปจนถึงย่านความถี่สูงๆ อย่างไรก็ตามเราไม่นิยมใช้วงจรกรองพาสซีฟในย่านความถี่ต่ำ เนื่องจากจะต้องใช้ขดลวดเหนี่ยวนำซึ่งมีขนาดใหญ่ทำให้มีความสูญเสียมากดังนั้นโดยทั่วไปวงจรกรองพาสซีฟจะถูกใช้งานตั้งแต่ย่านความถี่เสียง (Audio Frequency) ความถี่สัญญาณภาพ (Video Frequency) ไปจนถึงย่านความถี่สูงมาก (VHF)

2. วงจรกรองแบบแอคทีฟ (Active Filter) เป็นวงจรกรองสัญญาณแบบต่อเนื่องที่ใช้ อุปกรณ์ประเภทแอคทีฟร่วมกับตัวต้านทานและตัวเก็บประจุ วงจรกรองที่นิยมใช้ในปัจุบันคือ วงจรกรอง Active RC และวงจรกรองตัวเก็บประจุสวิทช์ (ซึ่งวงจรทั้งสองแบบสามารถใช้งานในย่านความถี่ไฟตรง – 500 kHz)

2.6 ประเภทของวงจรกรอง

2.6.1 วงจรกรองความถี่ต่ำผ่าน (Lowpass Filter)

วงจรกรองความถี่ต่ำผ่านเป็นวงจรที่ยอมให้สัญญาณความถี่ต่ำผ่านแต่จะกั้นความถี่สูงไว้รูปที่ 2.10 แสดงผลตอบสนองทางความถี่ของวงจรกรองความถี่ต่ำผ่านในอุดมคติ โดยที่วงจรกรองความถี่ต่ำผ่านจะสามารถกรองสัญญาณได้อย่างสมบูรณ์คือ ไม่ยอมให้สัญญาณที่มีความถี่มากกว่าความถี่ตัดผ่านออกมาได้เลยแต่ในทางปฏิบัติจะกรองสัญญาณได้ไม่สมบูรณ์คือจะมีสัญญาณที่มีความถี่มากกว่าความถี่ตัดผ่านออกมาได้แต่จะเป็นสัญญาณที่ถูกลดทอน โดยนิยามเราจะเรียกย่านความถี่ที่วงจรกรองยอมให้ผ่านว่าย่านความถี่ผ่าน (Pass Band) หรือ แบนด์วิดท์ (Bandwidth : BW) จะเป็นช่วงที่ความถี่ถูกลดทอนลงมา 0.707 เท่าของแอมพลิจูดสูงสุดหรือลดลงมา -3 dB จากแอมพลิจูดสูงสุดนั่นเอง ส่วนย่านความถี่ที่วงจรไม่ยอมให้ผ่านไปจะเรียกว่าย่านความถี่หยุด (Stop Band) จากรูปที่ 2.10 จะเห็นได้ว่าย่านความถี่ผ่านจะอยู่ระหว่าง 0 ถึง f_c (Cut-off Frequency)

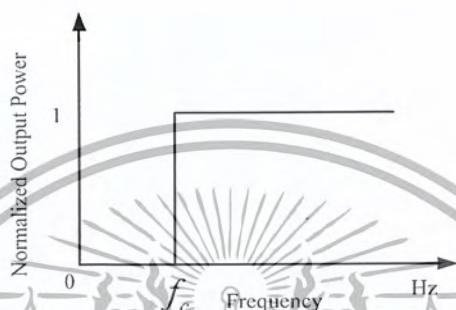


รูปที่ 2.10 ผลตอบสนองทางความถี่ของวงจรกรองความถี่ต่ำผ่าน

2.6.2 วงจรกรองความถี่สูงผ่าน (Highpass Filter)

จากรูปที่ 2.11 แสดงผลตอบสนองทางความถี่ของวงจรกรองความถี่สูงผ่านในอุดมคติ โดยที่วงจรกรองความถี่สูงผ่านในอุดมคติจะสามารถกรองสัญญาณได้อย่างสมบูรณ์แบบคิจะ ไม่ยอมให้สัญญาณที่มีความถี่น้อยกว่าความถี่ตัด (Cut-off Frequency) ผ่านออกมาได้เลย แต่

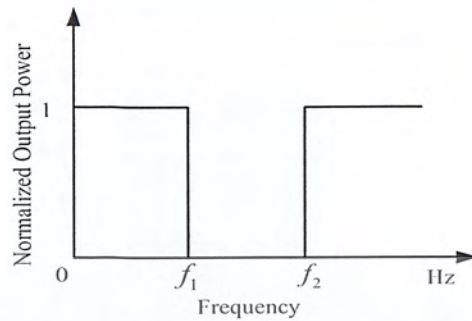
ในทางปฏิบัติแล้วจะไม่สามารถกรองสัญญาณได้อย่างสมบูรณ์ คือจะมีสัญญาณที่มีความถี่น้อยกว่าความถี่ตัดผ่านออกมาได้เล็กน้อย โดยช่วงที่มีความถี่สูงกว่าความถี่ตัดเรียกว่าย่านความถี่ผ่าน (Pass Band) ส่วนช่วงที่มีความถี่ต่ำกว่าความถี่ตัดเรียกว่าย่านความถี่หยุด (Stop Band)



รูปที่ 2.11 ผลตอบสนองทางความถี่ของวงจรกรองความถี่สูงผ่าน

2.6.3 วงจรกรองแถบความถี่หยุด (Band-Reject Filter หรือ Notch Filter)

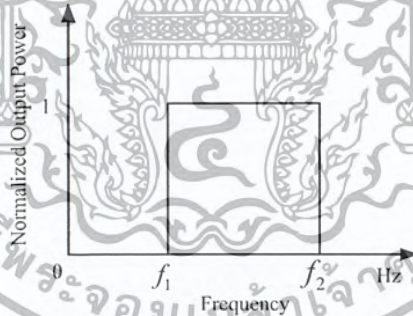
วงจรกรองแถบความถี่หยุดจะทำหน้าที่ตรงกันข้ามกับวงจรกรองแถบความถี่ผ่าน กล่าวคือวงจรกรองแถบความถี่หยุดจะกั้นไม่ให้สัญญาณ ณ ช่วงความถี่ใดๆ ผ่านไปได้ แต่จะยอมให้สัญญาณที่มีความถี่ต่ำหรือสูงกว่าช่วงความถี่นั้นผ่านได้ จากรูปที่ 2.12 เป็นผลตอบสนองทางความถี่ของวงจรกรองแถบความถี่หยุดในอุดมคติ โดยในอุดมคตินั้นจะสามารถกรองสัญญาณได้อย่างสมบูรณ์คือจะให้สัญญาณที่มีความถี่อยู่ในช่วงมากกว่าและน้อยกว่าความถี่ f_1 และ f_2 ผ่านออกมาได้ความถี่ที่อยู่ในช่วง f_1 และ f_2 จะไม่สามารถผ่านออกมาได้เลย แต่ในทางปฏิบัติสัญญาณที่อยู่ในช่วงความถี่ f_1 และ f_2 จะผ่านได้เล็กน้อย นิยามช่วงความถี่ระหว่าง f_1 และ f_2 เรียกว่าช่วงความถี่หยุด (Stop Band) และช่วงความถี่ที่น้อยกว่า f_1 และช่วงความถี่ที่มากกว่า f_2 เรียกว่าช่วงความถี่ผ่าน (Pass Band)



รูปที่ 2.12 ผลตอบสนองทางความถี่ของวงจรกรองแถบความถี่สูงสุด

2.6.4 วงจรกรองแถบความถี่ผ่าน (Bandpass Filter)

วงจรกรองแถบความถี่ผ่านเป็นวงจรกรองที่ยอมให้สัญญาณในช่วงความถี่ใดๆ ผ่านไปได้ แต่จะไม่ยอมให้สัญญาณที่มีความถี่ต่ำกว่าหรือสูงกว่าช่วงความถี่นั้นผ่านไปได้ รูปที่ 2.13 แสดงผลตอบสนองทางความถี่ของวงจรกรองแถบความถี่ผ่าน โดย f_1 และ f_2 เป็นจุดปลายของย่านความถี่ผ่านของวงจร โดย $f_1 < f_2$ เราจะพบว่าแบนด์วิดท์ของวงจรคือ



รูปที่ 2.13 ผลตอบสนองทางความถี่ของวงจรกรองแถบความถี่ผ่าน

$$BW = f_2 - f_1 \quad (2.1)$$

และเราจะนิยามความถี่กึ่งกลาง (Center Frequency) : f_c ว่าเป็น

$$f_c = \sqrt{f_1 f_2} \quad (2.2)$$

ซึ่งโดยทั่วไปเราจะกำหนดให้ BW ของวงจรกรองแถบความถี่ผ่านคือแถบความถี่ที่

สัญญาณสามารถผ่านไปได้โดยมีการลดทอนไม่เกิน 3 dB นั่นคือ $BW = BW_{-3dB}$

ตัวประกอบคุณภาพ (Quality Factor: Q) เป็นตัวบอกแถบความถี่ผ่านของวงจร หมายถึง กรณีสที่ค่า Q ต่ำ แบนด์วิดท์จะกว้าง ในทางกลับกันถ้าค่า Q สูง แบนด์วิดท์จะแคบ

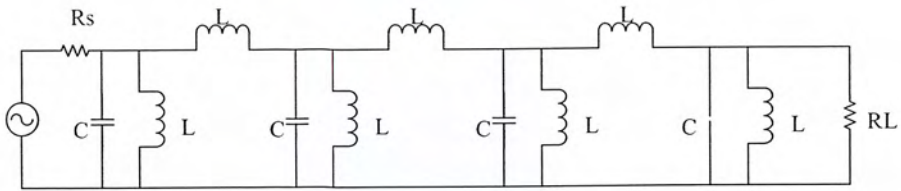
$$Q = \frac{\omega_0}{BW} = \frac{\omega_0}{f_2 - f_1} \quad (2.3)$$

2.7 วงจรกำทรเข้าคู่ (Coupled Resonator)

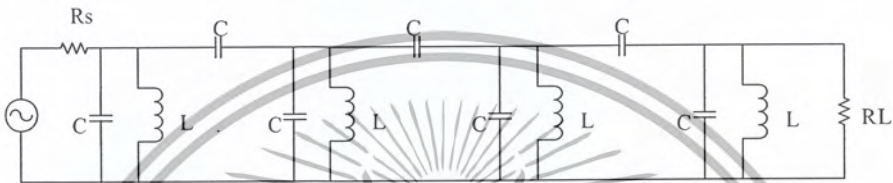
หากเราต้องการวงจรกรองแถบความถี่ผ่านที่มีความคมในการลดทอนสัญญาณนอกแถบความถี่ผ่านที่สูงขึ้น เราสามารถทำได้โดยการเชื่อมต่อ (Couple) วงจรกำทรเข้าด้วยกัน วงจรกรองความถี่ที่มีช่วงความถี่ผ่านแคบ ($\frac{f_H}{f_L} < 2$) สามารถออกแบบได้โดยใช้เทคนิคการคัปปลิง (Coupling) ซึ่งเป็นการนำตัวอุปกรณ์ เช่น ตัวเก็บประจุ (Capacitor) หรือ ตัวเหนี่ยวนำ (Inductor) มาต่อเชื่อมต่อระหว่างวงจรรزون (Tune) ที่ต่อแบบขนานโดยสามารถแสดงชนิดของโครงสร้างให้เห็นดังรูปที่ 2.14

โครงสร้างวงจรกรองความถี่ผ่านแบบคัปปลิงรีโซเนเตอร์ ออกแบบสำหรับวงจรกรองความถี่ผ่านที่มีค่า Q_c เท่ากับ 10 หรือมากกว่าค่าของตัวอุปกรณ์นั้นจะค่อนข้างมีความเป็นจริงมากกว่าค่าอุปกรณ์ที่ได้จากการสร้างวงจรกรองความถี่ผ่านแบบการแปลงจากวงจรกรองความถี่ต่ำผ่าน (Lowpass to Bandpass- Transformations) โดยเฉพาะอย่างยิ่งวงจรกรองความถี่ผ่านที่มีค่า Q_c สูงมาก ๆ โครงสร้างที่มีการคัปปลิงด้วยตัวเก็บประจุ (Capacitive Couple) เป็นวิธีที่ได้รับความนิยม เนื่องจากประหยัดค่าใช้จ่ายและในเรื่องของกระบวนการผลิตด้วย

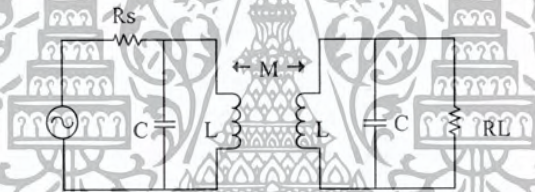
ตามทฤษฎีเหตุผลที่ใช้ในการอธิบายในเรื่องวิธีการออกแบบนั้นขึ้นอยู่กับสมมติฐานที่ว่า อุปกรณ์ที่ใช้ในการคัปปลิงมีค่าอิมพีแดนซ์ (Impedance) จะคงที่สำหรับโดเมนความถี่ สมมติฐานนี้มีการประมาณอย่างแม่นยำในช่วงที่มีแบนด์วิดท์แคบๆ ที่ระดับ DC นั้นตัวเก็บประจุที่ใช้ในการคัปปลิงนั้นจะทำให้ผลตอบสนองของค่าซีโร (Zero) มีมากขึ้นด้วยเหตุผลนี้จึงทำให้ผลตอบสนองทางความถี่มีค่าเพิ่มขึ้นแบบไม่สมมาตรทั้งในทางคณิตศาสตร์และเรขาคณิต ซึ่งค่าที่ได้นั้นจะเบี่ยงเบนออกจากความถี่กึ่งกลาง (Center Frequency) ลักษณะของผลตอบสนองในด้านความถี่ต่ำผ่านอาจมีความสูงชันมากกว่าก็อาจมีความเป็นไปได้อีก



(ก.)



(ข.)

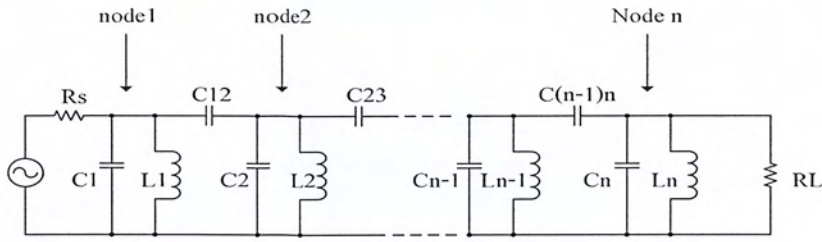


(ค.)

รูปที่ 2.14 (ก.) การคัปปลิงด้วยตัวเหนี่ยวนำ (ข.) การคัปปลิงด้วยตัวเก็บประจุ (ค.) การคัปปลิงด้วยตัวเหนี่ยวนำแม่เหล็ก

รูปแบบทั่วไปของการสร้างวงจรช่วงความถี่ผ่านนั้นอาจแสดงให้เห็นดังรูปที่ 2.15 จำนวนอันดับ (n-Order) นั้นจะกำหนดให้มีวงจร Tuning (Tune Circuit) ที่ต่อแบบขนานจำนวน n วงจรและมีจำนวนโหนด n โหนดจากตารางที่ 2.1 ถึงตารางที่ 2.10 นั้นเป็นตารางที่แสดงค่าพารามิเตอร์ q และ k พารามิเตอร์นี้ทำให้เกิดเป็นค่าประจำอุปกรณ์ที่ใช้สร้างวงจรกรองความถี่ผ่านโดยใช้ตัวเก็บประจุเป็นตัวคัปปลิงดังรูปที่ 2.14(ข.) สำหรับแต่ละโครงข่าย (Network) ค่าพารามิเตอร์ q_1 และ q_n จะสอดคล้องกับวงจรรีโซแนนซ์ (Resonance) ส่วนค่าพารามิเตอร์ k นั้นจะอยู่ในเทอมของ k_{12}, k_{23}, \dots ซึ่งจะสอดคล้องกับค่าคัปปลิงที่ต่ออนุกรมอยู่ในวงจรดังรูปที่ 2.15

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.15 รูปแบบทั่วไปของวงจรกรองช่วงความถี่ผ่านโดยใช้ตัวเก็บประจุเป็นตัวคัปปลิง

ตารางที่ 2.1 Butterworth Capacitive Coupled Resonators

N	q_1	q_2	k_{12}	k_{23}	k_{34}	k_{45}	k_{56}	k_{67}	k_{78}
2	1.414	1.414	0.707						
3	1.000	1.000	0.707	0.707					
4	0.765	0.765	0.841	0.541	0.841				
5	0.618	0.618	1.000	0.556	0.556	1.000			
6	0.518	0.518	1.169	0.605	0.518	0.605	1.169		
7	0.445	0.445	1.342	0.667	0.527	0.527	0.667	1.342	
8	0.390	0.390	1.519	0.736	0.554	0.510	0.554	0.736	1.519

ตารางที่ 2.2 0.01-dB Chebyshev Capacitive Couple Resonators

n	q_1	q_2	k_{12}	k_{23}	k_{34}	k_{45}	k_{56}	k_{67}	k_{78}
2	1.483	1.483	0.708						
3	1.181	1.181	0.682	0.682					
4	1.046	1.046	0.737	0.541	0.737				
5	0.977	0.977	0.780	0.540	0.540	0.780			
6	0.937	0.937	0.809	0.550	0.518	0.550	0.809		
7	0.913	0.913	0.829	0.560	0.517	0.517	0.560	0.829	
8	0.897	0.897	0.843	0.567	0.520	0.520	0.520	0.567	0.843

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.3 0.1-dB Chebyshev Capacitive Couple Resonators

n	q_1	q_2	k_{12}	k_{23}	k_{34}	k_{45}	k_{56}	k_{67}	k_{78}
2	1.638	1.638	0.711						
3	1.433	1.433	0.622	0.622					
4	1.345	1.345	0.685	0.542	0.685				
5	1.301	1.301	0.703	0.536	0.536	0.703			
6	1.277	1.277	0.715	0.539	0.518	0.539	0.715		
7	1.262	1.262	0.722	0.542	0.516	0.516	0.542	0.722	
8	1.251	1.251	0.728	0.545	0.516	0.510	0.516	0.545	0.728

ตารางที่ 2.4 0.5-dB Chebyshev Capacitive Couple Resonators

n	q_1	q_2	k_{12}	k_{23}	k_{34}	k_{45}	k_{56}	k_{67}	k_{78}
2	1.950	1.950	0.723						
3	1.864	1.864	0.647	0.647					
4	1.826	1.826	0.648	0.545	0.648				
5	1.807	1.807	0.652	0.534	0.534	0.652			
6	1.796	1.796	0.655	0.533	0.519	0.533	0.655		
7	1.790	1.790	0.657	0.533	0.516	0.516	0.533	0.657	
8	1.785	1.785	0.658	0.533	0.515	0.511	0.515	0.533	0.658

ตารางที่ 2.5 1-dB Chebyshev Capacitive Couple Resonators

n	q_1	q_2	k_{12}	k_{23}	k_{34}	k_{45}	k_{56}	k_{67}	k_{78}
2	2.210	2.210	0.739						
3	2.210	2.210	0.645	0.645					
4	2.210	2.210	0.638	0.546	0.638				
5	2.210	2.210	0.633	0.535	0.538	0.633			
6	2.250	2.250	0.631	0.531	0.510	0.531	0.531		
7	2.250	2.250	0.631	0.530	0.517	0.517	0.530	0.631	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.6 Bessel Capacitive Couple Resonators

n	q_1	q_2	k_{12}	k_{23}	k_{34}	k_{45}	k_{56}	k_{67}	k_{78}
2	0.5755	0.148	0.900						
3	0.337	2.203	1.748	0.684					
4	0.233	2.240	2.530	1.175	0.644				
5	0.394	0.275	1.910	0.750	0.650	1.987			
6	0.145	0.187	2.000	0.811	0.601	1.253	3.038		
7	0.187	0.242	3.325	1.660	1.293	0.695	0.674	2.203	
8	0.139	0.242	4.284	2.079	1.484	1.246	0.678	0.697	2.286

ตารางที่ 2.7 Linear Phase With Equiripple Error of 0.05 Capacitive Coupled Resonators

n	q_1	q_2	k_{12}	k_{23}	k_{34}	k_{45}	k_{56}	k_{67}	k_{78}
2	0.648	2.109	0.856						
3	0.433	2.254	1.489	0.652					
4	0.493	0.718	0.632	0.718	0.739				
5	0.547	0.446	0.800	0.848	0.584	1.372			
6	0.397	0.468	0.993	1.379	0.683	0.661	1.553		
7	0.316	0.484	2.490	1.442	1.446	0.927	0.579	1.206	
8	0.335	0.363	2.585	1.484	1.602	1.160	0.596	0.868	1.733

ตารางที่ 2.8 Linear Phase With Equiripple Error of 0.5 Capacitive Coupled Resonators

n	q_1	q_2	k_{12}	k_{23}	k_{34}	k_{45}	k_{56}	k_{67}	k_{78}
2	0.825	1.980	0.783						
3	0.553	2.425	1.330	0.635					
4	0.581	1.026	1.575	0.797	0.656				
5	0.664	0.611	1.779	0.919	0.576	1.162			
6	0.552	0.586	1.874	1.355	0.641	0.721	1.429		
7	0.401	0.688	2.324	1.394	1.500	1.079	0.590	1.045	
8	0.415	0.563	2.410	1.470	1.527	1.409	0.659	0.755	1.335

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.9 Transitional Gaussian to 6-dB Capacitive Coupled Resonators

n	q_1	q_2	k_{12}	k_{23}	k_{34}	k_{45}	k_{56}	k_{67}	k_{78}
2									
3	0.404	2.338	1.662	0.691					
4	0.570	0.914	1.623	0.798	0.682				
5	0.891	0.670	1.418	0.864	0.553	1.046			
6	0.883	0.752	1.172	1.029	0.595	0.605	1.094		
7	0.736	0.930	1.730	0.955	0.884	0.534	0.633	1.104	
8	0.738	0.948	1.724	0.866	0.922	0.708	0.501	0.752	1.089

ตารางที่ 2.10 Transitional Gaussian to 6-dB Capacitive Coupled Resonators

n	q_1	q_2	k_{12}	k_{23}	k_{34}	k_{45}	k_{56}	k_{67}	k_{78}
2									
3	0.415	2.345	1.631	0.686					
4	0.419	0.766	1.989	0.833	0.740				
5	0.534	0.503	2.085	0.976	0.605	1.333			
6	0.543	0.558	1.839	1.442	0.686	0.707	1.468		
7	0.492	0.665	1.708	1.440	1.181	0.611	0.781	1.541	
8	0.549	0.640	1.586	1.262	1.896	0.808	0.569	1.023	1.504

2.7.1 ขั้นตอนการออกแบบวงจรกรองช่วงความถี่ผ่านแบบ (Narrowband Coupled Resonators)

คำนวณหาค่า Q จากความสัมพันธ์

$$BW = f_2 - f_1$$

$$Q_{BPF} = \frac{f_c}{BW_{3dB}}$$

(2.4)

กำหนดค่า q 's และ k 's จากตารางที่ 2.1-2.10 โดยต้องเลือกฟังก์ชัน และอันดับที่ใช้ในการออกแบบวงจรกรองช่วงความถี่ผ่านโดยใช้ตัวเก็บประจุเป็นตัวคัปปลิง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$Q_1 = Q_{BPF} \times q_1 \quad (2.5)$$

$$Q_n = Q_{BPF} \times q_n \quad (2.6)$$

$$K_{xy} = \frac{k_{xy}}{Q_{BPF}} \quad (2.7)$$

หาค่าแหล่งกำเนิดและโหลด ได้จากการกำหนดค่าตัวเหนี่ยวนำ (Inductance) จากความสัมพันธ์

$$R_s = \omega_0 L Q_1 \quad (2.8)$$

$$R_L = \omega_0 L Q_n \quad (2.9)$$

ผลรวมของค่าตัวเก็บประจุในแต่ละโหนด หาได้จากความสัมพันธ์

$$C_{node} = \frac{1}{\omega_0^2 L} \quad (2.10)$$

ค่าตัวเก็บประจุที่ต่อคัปปลิงระหว่างวงจรจูน (Tune) หาได้จากความสัมพันธ์

$$C_{xy} = K_{xy} C_{node} \quad (2.11)$$

ผลรวมของตัวเก็บประจุที่ต่ออยู่ในแต่ละโหนดต้องมีค่าเท่ากับ C_{node} เพราะฉะนั้น ตัวเก็บประจุที่ต่อขนานอยู่ในวงจรจูน (Tune) ต้องมีค่าเท่ากับ C_{node} โดยด้วยค่าตัวเก็บประจุที่ต่อคัปปลิงอยู่ในโหนดนั้น เช่น

$$C_1 = C_{node} - C_{12} \quad (2.12)$$

$$C_2 = C_{node} - C_{12} - C_{23} \quad (2.13)$$

$$C_7 = C_{node} - C_{67} - C_{78} \quad (2.14)$$

วงจรกรองช่วงความถี่ผ่านที่เสร็จสมบูรณ์นั้นค่าอิมพีแดนซ์ระหว่างแหล่งกำเนิดและโหลดจะต้องมีค่าที่เหมาะสม

2.8 วงโคจรดาวเทียม

2.8.1 หลักการพื้นฐานของระบบวงโคจรของดาวเทียม

ระบบวงโคจรของดาวเทียมคือ การที่ดาวเทียมโคจรรอบๆ โลกด้วยความเร็วต่างๆ และคงสภาพการเคลื่อนที่จากการที่จะถูกต้านทานด้วยแรงโน้มถ่วงของโลกและแรงเหวี่ยงที่ออกจากจุดศูนย์กลางของโลก กล่าวคือแรงโน้มถ่วงของโลกจะพยายามดึงดาวเทียมตกสู่พื้นโลก แต่

แรงเหวี่ยงที่ออกจากจุดศูนย์กลางของโลกจะดูดให้ดาวเทียมออกห่างจากโลกและออกจากวงโคจร ความสัมพันธ์ของแรงโน้มถ่วงของโลกและแรงเหวี่ยงที่ออกจากจุดศูนย์กลางของโลก มีผลกระทบกับวงโคจรของดาวเทียมดังนี้คือ

1. แรงโน้มถ่วงของโลกจะขึ้นอยู่กับน้ำหนักของดาวเทียม และความเร็วในการเคลื่อนที่ของดาวเทียม
2. แรงเหวี่ยงที่ออกจากจุดศูนย์กลางของโลกจะขึ้นอยู่กับน้ำหนักของดาวเทียม และความเร็วในการเคลื่อนที่ของดาวเทียม
3. ถ้าแรงโน้มถ่วงของโลกมากกว่าแรงเหวี่ยงที่ออกจากจุดศูนย์กลางของโลกจะทำให้ดาวเทียมมีวงโคจรเคลื่อนที่เข้าหาโลก
4. ถ้าแรงโน้มถ่วงของโลกน้อยกว่าแรงเหวี่ยงที่ออกจากจุดศูนย์กลางของโลกจะทำให้ดาวเทียมมีวงโคจรเคลื่อนที่ออกจากโลก
5. ถ้าแรงโน้มถ่วงของโลกเท่ากับแรงเหวี่ยงที่ออกจากจุดศูนย์กลางของโลกจะทำให้ดาวเทียมเคลื่อนที่คงที่รอบๆโลก

รูปที่ 2.16 แรงโน้มถ่วงของโลก และแรงเหวี่ยงออกจากจุดศูนย์กลางของโลก

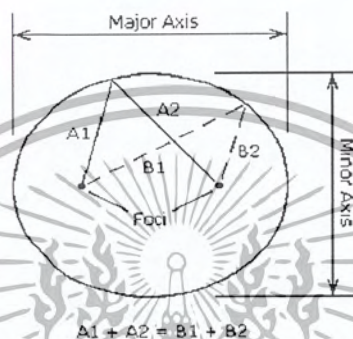
จากหลักการพื้นฐานดังกล่าวได้ทำให้เกิดทฤษฎีเกี่ยวกับการเคลื่อนที่ของดาวเทียม เพื่อกำหนดเป็นวงโคจรของดาวเทียมขึ้น ไม่ว่าจะเป็นกฎของ Kepler, Newton, Lagrange, Laplace, Gauss, และ Hamilton ก็เป็นบุคคลที่คิดค้นทฤษฎีและกฎต่างๆเกี่ยวกับระบบวงโคจรของดาวเทียม ในที่นี้จะทำการศึกษา 2 กฎที่มีความสำคัญโดยตรงกับระบบวงโคจรของดาวเทียม

2.8.1.1. Kepler's Laws (กฎของคีเพอร์)

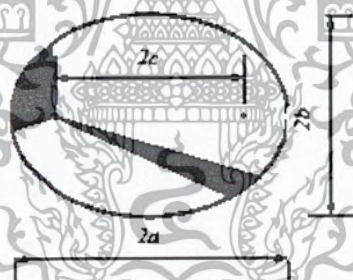
1. วงโคจรของดาวเทียมแต่ละดวงจะมีจุดโฟกัสเดียวเท่านั้นที่จะทำให้เป็นรูปวงรี

2. แนวการเคลื่อนที่ของวงโคจรของโลกถึงดาวเทียมจะมีวงกวาดของวงโคจรเท่ากัน และเวลาที่ใช้ในการเคลื่อนก็จะเท่ากันด้วยเส้นซึ่งลากระหว่างดวงอาทิตย์กับโลกจะกวาดพื้นที่ได้เท่ากัน เมื่อเวลาเท่ากัน ดังรูปที่ 2.18

3. สองเท่าของระยะทางการหมุนในวงโคจรจะเป็นสัดส่วนโดยตรงกับพื้นที่ครึ่งหนึ่งของวงโคจร



รูปที่ 2.17 ความสัมพันธ์ทางคณิตศาสตร์ของวงรี



รูปที่ 2.18 แสดงพื้นที่ที่กวาดได้

2.8.2 ประเภทวงโคจรดาวเทียม

สามารถแบ่งวงโคจรของดาวเทียมได้ตามระดับความสูง ประเภทการใช้งาน และตามระนาบวงโคจรดังนี้

2.8.2.1 ประเภทวงโคจรดาวเทียมตามระดับความสูง สามารถพิจารณาได้จากรูปที่ 2.19

1.) LEO (Low Earth Orbit) หมายถึงวงโคจรของดาวเทียมระดับต่ำ (400 – 2,000 กิโลเมตร) เหนือผิวโลกวงโคจรแบบนี้จะอยู่ระหว่างชั้นบรรยากาศกับ Van Allen radiation ซึ่งไม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มีการกำหนดความสูงที่แน่นอน แต่ดาวเทียมที่อยู่ในวงโคจรนี้จะอยู่สูงจากผิวโลกต่ำกว่า 2,000 กิโลเมตร ประโยชน์ของดาวเทียมในวงโคจรแบบนี้คือใช้ในการถ่ายภาพ สื่อสารเช่นดาวเทียม Iridium

ดาวเทียมที่มีวงโคจรเป็นวงกลมจะเป็น LEO จะโคจรอยู่ห่างจากผิวโลกไม่เกิน 1,000 กิโลเมตร ทำให้สามารถถ่ายภาพบนพื้นผิวโลกด้วยความละเอียดสูง การติดต่อสื่อสารกับดาวเทียมประเภทนี้จะใช้อุปกรณ์สายอากาศขนาดเล็กและเครื่องส่งวิทยุที่ต่ำกว่า จึงสะดวกที่จะผลิตอุปกรณ์สื่อสารที่มีขนาดเล็กซึ่งสะดวกในการใช้งานแต่มีข้อเสีย คือดาวเทียมประเภทนี้จะมี Footprint หรือพื้นที่ที่สามารถติดต่อสื่อสารได้มีขนาดเล็กและช่วงเวลาที่สามารถใช้งานได้ไม่เกิน 15 นาทีในการโคจรผ่านแต่ละครั้ง ซึ่งโดยทั่วไปจะโคจรผ่านเพียงวันละ 2 ครั้ง

2.) MEO (Medium Earth Orbit) หมายถึงวงโคจรของดาวเทียมระดับกลาง (8,000 – 12,000 กิโลเมตร) เหนือผิวโลกเป็นวงโคจรของดาวเทียมที่มีความสูงอยู่ระหว่าง LEO กับ GEO ประโยชน์ของดาวเทียมในวงโคจรนี้ได้แก่ ดาวเทียมสื่อสาร ดาวเทียมระบบนำร่อง เป็นต้น

3.) GEO (High Earth Orbit) หมายถึง วงโคจรของดาวเทียมระดับสูง (มากกว่า 20,000 กิโลเมตร) เหนือผิวโลกจะสังเกตเห็นว่าที่ระดับความสูง 2,000 – 8,000 กิโลเมตร และ 12,000 – 20,000 กิโลเมตร ไม่น่ามาใช้เป็นวงโคจรของดาวเทียมเพราะมีแถบรังสีความเข้มสูงที่เรียกว่า The Van Allan Belts ซึ่งมีโปรตอนและอิเล็กตรอนอยู่อย่างหนาแน่นโดยถูกกักไว้ด้วยสนามแม่เหล็กของโลก ดาวเทียมซึ่งเคลื่อนที่ผ่านแถบรังสีนี้อาจเกิดความเสียหายแก่อุปกรณ์อิเล็กทรอนิกส์ภายในได้ง่ายดังนั้นวงโคจรจึงถูกออกแบบให้ใช้เวลาเคลื่อนที่ผ่านแถบรังสีน้อยที่สุดหรือหลีกเลี่ยงไม่ให้โคจรผ่านแถบรังสีดังกล่าวเลย



รูปที่ 2.19 แสดงวงโคจรในระดับต่างๆ

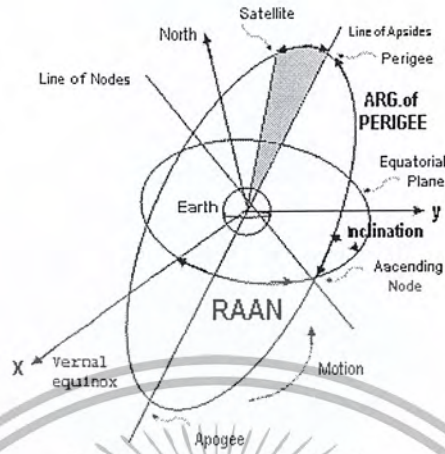
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.9 Keplerien Elements (Two line Element : TLE)

Keplerien Elements หรือ Satellite Orbital Elements คือ ชุดตัวเลขที่ใช้อธิบายวงโคจรของดาวเทียมแต่ละดวงค่า Keplerien Elements เปรียบได้กับการถ่ายภาพวัตถุที่กำลังเคลื่อนที่ ณ เวลาใดเวลาหนึ่ง แล้วอธิบายเป็นค่าความเร็วในการเคลื่อนที่และทิศทางในการเคลื่อนที่ของวัตถุนั้น ซึ่งค่า Keplerien Elements จะบอกถึงลักษณะวงโคจรของดาวเทียม แนวที่วงโคจรกระทำต่อโลก ตำแหน่งของดาวเทียมบนวงโคจร ณ เวลาที่ทำการวัดค่า โดยที่องค์ประกอบวงโคจรดาวเทียมมีหลายองค์ประกอบดังรูปที่ 2.20

ใน Keplerien Model ดาวเทียมมีวงโคจรเป็นวงรีที่มีรูปร่างและทิศทางคงที่ โลกอยู่ที่จุดโฟกัสจุดหนึ่งของวงรีไม่ใช่จุดศูนย์กลางของวงโคจร (ยกเว้นในกรณีที่วงโคจรเป็นวงกลมที่สมบูรณ์) ในโลกแห่งความเป็นจริงมีความซับซ้อนมากกว่า Keplerien Model ซึ่งโปรแกรมที่ใช้ติดตามดาวเทียมจะใช้ค่าเพิ่มเติมแก้ไขให้กับ Keplerien Model เช่นค่า Drag เป็นต้น

จากระนาบวงโคจรของดาวเทียมซึ่งเป็นรูปวงรี จุดที่ดาวเทียมโคจรเข้าใกล้ผิวโลกมากที่สุด เรียกว่า Perigee และจุดที่ดาวเทียมโคจรออกห่างจากโลกมากที่สุดเรียกว่า Apogee ส่วน Inclination คือมุมระหว่างเส้นตรงที่ลากผ่านจุดศูนย์กลางของโลกและขั้วโลกเหนือกับเส้นตรงที่ลากผ่านจุดศูนย์กลางของโลกและขั้วโลกเหนือกับเส้นตรงที่ลากผ่านจุดศูนย์กลางของโลกและตั้งฉากกับระนาบวงโคจรของดาวเทียม แนวที่ระนาบของเส้นศูนย์สูตรตัดกับระนาบวงโคจรของดาวเทียมจะเป็นเส้นตรงที่เรียกว่า Line of Nodes จุดที่ดาวเทียมโคจร จากซีกโลกเหนือไปยังซีกโลกใต้ ตัดกับแนวระนาบเส้นศูนย์สูตรเรียกว่า Descending node จุดที่ดาวเทียมโคจร จากซีกโลกใต้ไปยังซีกโลกเหนือตัดกับแนวระนาบเส้นศูนย์สูตรเรียกว่า Ascending node มุมที่วัดระหว่าง Line of Nodes (เส้นตรงที่ลากจากจุดศูนย์กลางของโลกไปยัง Ascending node) ไปยังเส้นผ่านศูนย์กลางหลักของวงโคจร (เส้นตรงที่ลากจากจุดศูนย์กลางของโลกไปยัง Perigee) เรียกว่า Argument of Perigee



รูปที่ 2.20 องค์ประกอบวงโคจร

มุมที่วัดได้ระหว่างเส้นตรงที่ลากจากจุดศูนย์กลางของโลกไปยังจุดตัดที่ดวงอาทิตย์ขึ้นผ่านเส้นศูนย์สูตรและเส้นตรงที่ลากจากจุดศูนย์กลางของโลกไปยังจุดตัดที่ดาวเทียมขึ้นผ่านเส้นศูนย์สูตร (Ascending node) คือ Right Ascension of Ascending Node (R.A.A.N.) โดยค่าตัวแปรบางตัวถูกนำมาแสดงใน Keplerian Elements หรือ Two Line Element นี้จะถูกนำมาใช้ในการคำนวณตามแบบจำลองที่มีการพิจารณาถึงแรงต้านจากภายนอกที่ทำให้การโคจรเปลี่ยนแปลงไปแบบจำลองที่ใช้คำนวณตำแหน่งและความเร็วของดาวเทียมที่โคจรรอบโลกที่มีความถูกต้องสูงสำหรับบุคคลทั่วไปใช้งานและดาวเทียมมีคาบเวลาการโคจรน้อยกว่า 255 นาที คือ SGP4 (Simplified General Perturbations No.4) ถ้าคาบเวลาการโคจรมากกว่านี้จะใช้แบบจำลอง SDP4 (Simplified Deep Space General Perturbations)

รูปแบบค่า Keplerian Elements

#NAME

```
1 NNNNNC NNNNAAA NNNNN.NNNNNNNN +.NNNNNNNN +NNNNN-N +NNNNN-N N NNNNN
2 NNNNN NNN.NNNN NNN.NNNN NNNNNNN NNN.NNNN NNN.NNNN NN.NNNNNNNNNNNNNNN
```

ตัวอย่าง Keplerian Elements

ISS (ZARYA)

```
1 25544U 98067A 05025.44039352 .00022267 00000-0 18298-3 0 6696
2 25544 51.6468 270.7801 0009320 29.2053 326.3814 15.70303393353239
```

ตารางที่ 2.11 ตัวอย่าง Two Line Element บรรทัดที่ 1

Field	Column	Description
1.1	01	Line Number of Element Data เลขบรรทัด=1
1.2	03-07	Satellite Number เลขประจำดาวเทียม = 25544
1.3	08	Classification / Unclassified = U (เปิดเผย)
1.4	10-11	International Designator (Last two digits of launch year) ปีที่ปล่อย=98
1.5	12-14	International Designator (Launch number of the year) ครั้งที่ปล่อยในรอบปี = 067
1.6	15-17	International Designator (Piece of the launch) ชิ้นที่ปล่อย = A
1.7	19-20	Epoch Year (Last two digits of year)
1.8	21-32	Epoch (Day of the year and fractional portion of the day)
1.9	34-43	First Time Derivative of the Mean Motion Rate of change of MM = .00022267
1.10	45-52	Second Time Derivative of Mean Motion (decimal point assumed)
1.11	54-61	BSTAR drag term (decimal point assumed) Drag = .0018298
1.12	63	Ephemeris type
1.13	65-68	Element number Element ลำดับที่ 669
1.14	69	Checksum (Modulo 10) Check sum = 6 (Letters, blanks, periods, plus signs = 0; minus signs = 1)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.12 ตัวอย่าง Two Line Element บรรทัดที่ 2

Field	Column	Description
2.1	01	Line Number of Element Data เลขบรรทัด = 2
2.2	03-07	Satellite Number เลขประจำดาวเทียม = 25544
2.3	09-16	Inclination [Degrees] Inclination (degrees) = 51.6468
2.4	18-25	Right Ascension of the Ascending Node [Degrees] RAAN (degrees) = 270.7801
2.5	27-33	Eccentricity (decimal point assumed) Eccentricity = 0009320
2.6	35-42	Argument of Perigee [Degrees] Argument of perigee (degrees) = 29.2053
2.7	44-51	Mean Anomaly [Degrees] Mean anomaly (degrees) = 326.3814, ให้คูณด้วย 256/360 สำหรับ phase units
2.8	53-63	Mean Motion [Revs per day] Mean motion (rev./day) = 15.70303393
2.9	64-68	Revolution number at epoch [Revs] จำนวนรอบที่เวลา epoch = 35323
2.10	69	Checksum (Modulo 10) Check sum = 9

2.9.1 ส่วนประกอบที่สำคัญของ Keplerien Elements

1. Epoch หรือ Epoch Time หมายถึง ช่วงเวลาที่วัดค่า Keplerien Element ซึ่งมีรูปแบบของเวลาดังนี้ YYDOY.FOD โดยที่ YY เป็นเลขสองหลักท้ายของปี ค.ศ. DOY เป็นเลขสามหลักบอกวันที่ของปีนั้นโดยนับวันที่ 1 มกราคม เป็นวันที่ 1 FOD เรียกว่า Fraction of day เป็นเศษส่วนของเวลาเทียบกับ Solar day โดยการแปลงเวลาให้เป็น FOD จะทำได้ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$FOD = \frac{1}{24} \left\{ Hr + \frac{\text{min}}{60} + \frac{\text{sec}}{3600} \right\} \quad (2.23)$$

ตัวอย่างเช่น เวลา 15:35:04 ของวันที่ 245 ของปี ค.ศ. 2010 เขียนได้เป็น 10245.64935185

2. Orbital Inclination หรือ Inclination หมายถึง มุมระหว่างระนาบวงโคจรของดาวเทียมกับระนาบของเส้นศูนย์สูตร Inclination มีค่าอยู่ระหว่าง 0 – 180 องศา ถ้า Inclination มีค่าใกล้ 0 จะเรียกว่า Equatorial Orbit เพราะดาวเทียมจะโคจรเหนือแนวเส้นศูนย์สูตร ถ้า Inclination มีค่าใกล้กับ 90 องศา จะเรียกว่า Polar Orbit เพราะดาวเทียมจะโคจรผ่านเหนือขั้วโลกเหนือและใต้ ถ้า Inclination มีค่าระหว่าง 0 – 90 องศา จะเรียกว่า Prograde Orbit ถ้า Inclination มีค่าระหว่าง 90 – 180 องศา

3. Right Ascension of Ascending Node (R.A.A.N.) หมายถึง มุมที่วัดระหว่างเส้นตรงที่ลากจากจุดศูนย์กลางของโลกไปยังจุดตัดที่ดวงอาทิตย์ขึ้นบนเส้นศูนย์สูตรและเส้นตรงที่ลากจากจุดศูนย์กลางของโลกไปยังจุดตัดที่ดาวเทียมขึ้นบนเส้นศูนย์สูตร (ascending node)

4. Argument of Perigee หมายถึง มุมที่วัดระหว่าง Line of Nodes (เส้นตรงที่ลากจากจุดศูนย์กลางของโลกไปยัง Ascending Node) ไปยังเส้นผ่าศูนย์กลางหลักของวงโคจร (เส้นตรงที่ลากจากจุดศูนย์กลางของโลกไปยัง Perigee)

5. Eccentricity (e) หมายถึง ค่าเยื้องศูนย์กลางของวงโคจร มีค่ามากกว่าหรือเท่ากับ 0 และน้อยกว่า 1 ถ้า e = 0 วงโคจรจะเป็นวงกลม 1 ถ้า e มีค่าใกล้ 1 วงโคจรจะเป็นวงรีที่ยาวและแคบ

6. Mean Motion หมายถึง จำนวนรอบของวงโคจร (จากจุด perigee ถึงจุด perigee) ที่ดาวเทียมโคจรภายในหนึ่งวัน (Solar day = 1,440 นาที) หรือ $MM = 1,440/T$

7. Mean Anomaly หมายถึง ตำแหน่งของดาวเทียมบนวงโคจรมีค่าเท่ากับ 0 ถึง 256 ที่จุด Perigee, MA มีค่าเท่ากับ 0 ถึง 256 และที่จุด Apogee MA มีค่าเท่ากับ 128 การหาค่า MA จะคิดจากค่าเวลาที่ใช้ในการโคจร

2.10 กราฟฟิคอล ยูเซอร์ อินเตอร์เฟซ (Graphical user interface (GUI))

GUI (Graphical User Interface) เป็นวิธีการให้ความสะดวกแก่ผู้ใช้คอมพิวเตอร์ ให้ติดต่อสื่อสารกับเครื่องคอมพิวเตอร์โดยผ่านทางภาพ เช่น ใช้เมาส์กดเลือกสัญลักษณ์รูป (Icon)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แทนการพิมพ์คำสั่งดังแต่ก่อน หรือการเลือกคำสั่งตามรายการเลือกที่เรียกว่า ระบบเมนู

GUI เป็นอินเตอร์เฟซด้วยกราฟฟิกของผู้ใช้กับเครื่องคอมพิวเตอร์ เช่น เว็บเบราว์เซอร์ คำนี้เกิดขึ้น เนื่องจากการอินเตอร์เฟซกับคอมพิวเตอร์ในรุ่นแรกไม่ได้ใช้กราฟฟิก แต่เป็นการใช้ตัวอักษรและเป็นพิมพ์ ปกติจะเป็นคำสั่ง เช่น ระบบปฏิบัติการ DOS ในขั้นกลางการอินเตอร์เฟซของผู้ใช้เป็นการอ่านอินเตอร์เฟซแบบเมนู (Menu-based Interface) ซึ่งยอมให้ใช้เมาส์คลิกคำสั่งได้ นอกจากการพิมพ์เป็นพิมพ์ ระบบปฏิบัติการส่วนใหญ่จะเป็นแบบ GUI ในส่วนโปรแกรมประยุกต์จะใช้องค์ประกอบของ GUI ที่มากับระบบปฏิบัติการและเพิ่มการอินเตอร์เฟซของตัวเองเข้าไป บางครั้ง GUI ใช้ชื่อมากกว่าหนึ่งในการทำงานจริง

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, ระบบปฏิบัติการUbuntu และโปรแกรมประยุกต์ต่างๆ ได้มีจุดเริ่มต้นที่ Xerox Palo Alto Research Laboratory ต่อมาในทศวรรษที่ 70 Apple ได้เริ่มใช้ครั้งแรกในเครื่องคอมพิวเตอร์แมคอินทอช ต่อมา Microsoft ได้นำเอาแนวคิดมาพัฒนาเป็นระบบปฏิบัติการ Windows กับเครื่อง IBM-Compatible เมื่อมีการสร้างโปรแกรมประยุกต์ เครื่องมือแบบ Object-Oriented จะเขียนการอินเตอร์เฟซด้วยกราฟฟิก ในแต่ละสมาชิกของ GUI จะเรียกว่า Class Form เมื่อสร้างอ็อบเจกต์ขึ้นมาแล้ว ซึ่งสามารถเขียนคำสั่งหรือปรับปรุงด้วยเมธอด (Method) เพื่อให้อ็อบเจกต์เหล่านั้นตอบสนองต่อการใช้งานของผู้ใช้

2.11 ภาษาไพธอน

ภาษาไพธอนเป็นเครื่องมือที่มีการอินเตอร์เฟซคล้ายกับ Matlab กล่าวคือประมวลผลคำสั่งทีละบรรทัดหรือจะเป็นโปรแกรมแบบซอร์ส (Source) ไฟล์จากนั้นสั่งให้ประมวลผลในภายหลังได้

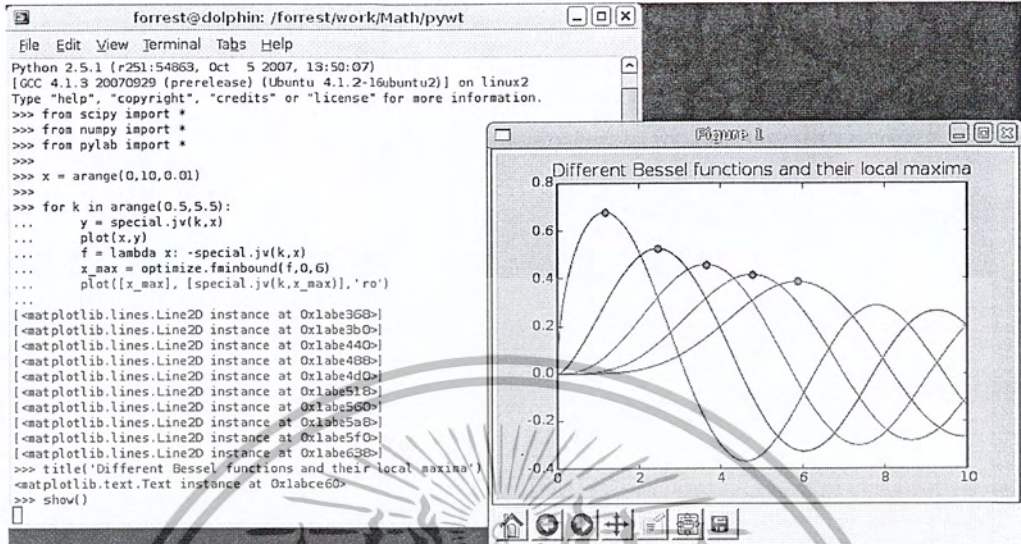
ช่วงต้นปี ค.ศ. 1990 (พ.ศ. 2553) Guido Van Rossum ได้ออกแบบและสร้างภาษาไพธอน โดยโค้ดไพธอนทั้งหมดถูกสร้างขึ้นมาจากภาษา C โดยที่ภาษาไพธอนเป็นโอเพนซอร์ส

กล่าวคือ ผู้สร้างไพธอนไม่คิดมูลค่ากับสิ่งที่สร้างสรรค์ขึ้นมาสามารถใช้งานได้อย่างอิสระ (Freedom) ไม่มีค่าลิขสิทธิ์ อีกทั้งอนุญาตให้นักพัฒนาสามารถแก้ไขโปรแกรมให้มีความสามารถสูงขึ้นได้อีกด้วย ดังนั้นเป็นเหตุผลให้ผู้เชี่ยวชาญทั้งหลายในหลากหลายสาขา ต่างร่วมกันสร้างเครื่องมือเพื่อประกอบให้ไพธอนเป็นโปรแกรมที่ดีและนำไปใช้งานในด้านต่างๆ ได้อย่างกว้างขวาง ภายในไพธอนประกอบด้วยมอดูล (Module) ต่างๆ มากมายและในแต่ละมอดูลประกอบด้วยคำสั่งหรือฟังก์ชันอีกจำนวนมาก ดังนั้น โปรแกรมเมอร์สามารถเรียกใช้ชุดคำสั่งและมอดูลเหล่านั้นได้อย่างสะดวกรวดเร็ว ง่ายดาย และมีประสิทธิภาพนอกจากนั้นถ้าผู้ที่มีความชำนาญทางด้านภาษา C หรือ C++ ท่านสามารถสร้างโปรแกรมให้สามารถทำงานร่วมกับไพธอนได้ หรือใช้เป็นส่วนหนึ่งของไพธอนได้อีกด้วย ปัจจุบันนี้มีผู้เชี่ยวชาญหลายท่านได้สร้างเครื่องมือเพื่อสนับสนุนการใช้งานภาษาไพธอน ได้แก่ ด้านปัญญาประดิษฐ์ ด้านคณิตศาสตร์ ด้านกราฟิก ด้านเน็ตเวิร์ก และด้านอื่นๆ อีกมากมาย

2.11.1 Library/Module ที่นำมาใช้งาน

Module Scipy คือ เครื่องมือที่ใช้ในการคำนวณทางวิทยาศาสตร์ที่ใช้ร่วมกับโปรแกรมภาษาไพธอน โดยเป็นเครื่องมือที่เป็น Open Source Software โดยการใช้งานของ Scipy จะใช้งานร่วมกับ โมดูล (Module) ที่ชื่อ Numpy ที่เป็นเครื่องมือที่ใช้ในการสร้างอาร์เรย์ (Array) ในการคำนวณ Scipy จะมีฟังก์ชันที่เป็นประโยชน์ มากมาย เช่น เครื่องมือในการแปลง FFT (Fast Fourier Transform) , การคำนวณทางเมทริกซ์ (Matrix) และ การประมวลผลสัญญาณ

Module Matplotlib คือ เครื่องมือที่ใช้ในการพลอตกราฟทางคณิตศาสตร์ ในภาษาไพธอน และ ยังใช้ร่วมกับโมดูล Numpy ซึ่งลักษณะการใช้งานจะใกล้เคียงกับ โปรแกรม MATLAB



รูปที่ 2.21 ตัวอย่างการใช้โมดูล SciPy และ Matplotlib

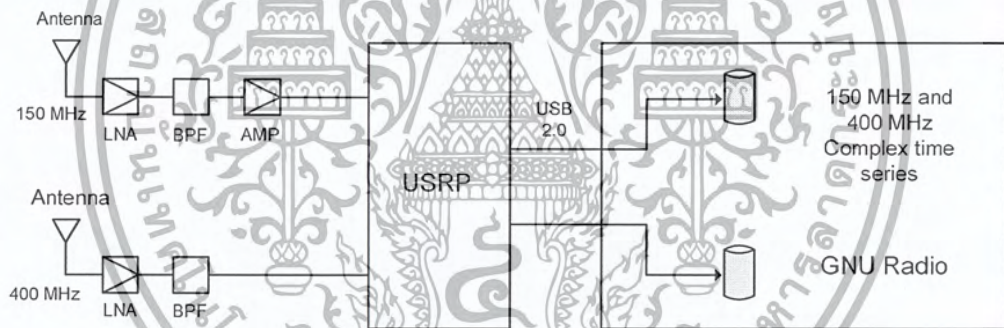
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

การออกแบบและการจัดทำปริญญาานิพนธ์

3.1 หลักการทำงานโดยรวมของระบบ

จากรูปที่ 3.1 สัญญาณบีคอนความถี่ 150 MHz และ 400 MHz จะถูกส่งลงมาจกดาวเทียมโดยทำการรับสัญญาณผ่านสายอากาศเข้าสู่วงจรกรองแถบความถี่ผ่านย่านความถี่ 150 MHz และ 400 MHz จากนั้นสัญญาณจะถูกส่งผ่านไปยังชุดพัฒนา USRP ซึ่งอุปกรณ์นี้จะทำการแปลงสัญญาณแอนะล็อกความถี่สูงให้กลายเป็นสัญญาณดิจิทัลเบสแบนด์และส่งข้อมูลไปยังคอมพิวเตอร์ผ่านทางพอร์ต USB 2.0 จากนั้นจะทำการบันทึกสัญญาณดิจิทัลเบสแบนด์เก็บไว้เป็นไฟล์ข้อมูลทั้งสัญญาณ 150 MHz และ 400 MHz เพื่อเตรียมไว้สำหรับการประมวลผลค่าความต่างเฟสในภายหลัง



รูปที่ 3.1 บล็อกไดอะแกรมการทำงานโดยรวมของระบบ

3.2 การออกแบบวงจรกรองความถี่ผ่านแบบเข้าคู่ (Coupled Resonator)

การออกแบบวงจรกรองความถี่ผ่าน 150 MHz

ความถี่กึ่งกลาง (Center Frequency) 150 MHz

แบนด์วิดท์ (BW) 10 MHz

ฟังก์ชันที่ใช้ในการออกแบบ Butterworth Function order 3

จากข้อกำหนดในการออกแบบวงจรกรองความถี่ที่ใช้ฟังก์ชัน Butterworth order 3,

$$BW=10 \text{ MHz}, f_c=150 \text{ MHz}$$

ตารางที่ 3.1 Butterworth Capacitive Coupled Resonators

n	q ₁	q _n	k ₁₂	k ₂₃	k ₃₄	k ₄₅	k ₅₆	k ₆₇	k ₇₈
2	1.414	1.414	0.707						
3	1.000	1.000	0.707	0.707					
4	0.765	0.765	0.841	0.541	0.841				
5	0.618	0.618	1.000	0.556	0.556	1.000			
6	0.518	0.518	1.169	0.605	0.518	0.605	1.169		
7	0.445	0.445	1.342	0.667	0.527	0.527	0.667	1.342	
8	0.390	0.390	1.519	0.736	0.554	0.510	0.554	0.736	1.519

วิธีทำ จากตารางที่ 3.1 เราสามารถหาค่าพารามิเตอร์ q และ k (order $n=3$) ได้ค่าต่างๆ ดังนี้

$$q_1 = 1.000$$

$$q_2 = 1.000$$

$$k_{12} = 0.707$$

$$k_{23} = 0.707$$

จากข้อกำหนดในการออกแบบสามารถหาค่า Q ได้จากความสัมพันธ์

$$Q_{bp} = \frac{f_0}{BW_{3dB}} = \frac{150 \text{ MHz}}{10 \text{ MHz}} = 15$$

ดังนั้น

$$Q_1 = Q_{BPF} \times q_1 = 15 \times 1.000 = 15$$

$$Q_3 = Q_{BPF} \times q_3 = 15 \times 1.000 = 15$$

$$K_{12} = \frac{k_{12}}{Q_{BPF}} = \frac{0.707}{15} = 0.04713$$

$$K_{23} = \frac{k_{23}}{Q_{BPF}} = \frac{0.707}{15} = 0.04713$$

กำหนดค่า $R_s = 50 \Omega$ และ $R_L = 50 \Omega$ ดังนั้นสามารถคำนวณหาค่า L ได้จากความสัมพันธ์

$$L = \frac{R_{L,S}}{\omega_0 Q_{1,3}} = \frac{50}{2\pi(150 \times 10^6)(15)} = 3.537 \times 10^{-9} = 3.537 \text{ nH}$$

ดังนั้นเราสามารถคำนวณหาค่า C_{node} ได้จากความสัมพันธ์

$$C_{node} = \frac{1}{\omega_0^2 L} = \frac{1}{[(2\pi)(150 \times 10^6)]^2 (3.537 \times 10^{-9})} = 318.29 \text{ pF}$$

จากค่า $C_{Coupling}$ เราสามารถหาค่า $C_{Coupling}$ ได้จากความสัมพันธ์

$$C_{12} = K_{12} C_{node} = (0.04713)(318.29 \times 10^{-12}) = 15 \text{ pF}$$

$$C_{23} = K_{23} C_{node} = (0.04713)(318.29 \times 10^{-12}) = 15 \text{ pF}$$

ส่วนค่าตัวเก็บประจุที่ต่อขนาน สามารถหาได้จากความสัมพันธ์

$$C_1 = C_{node} - C_{12} = 318.29 - 15 = 303.29 \text{ pF}$$

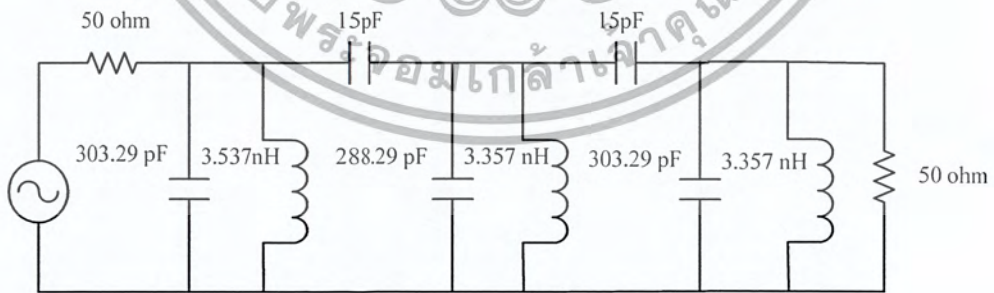
$$C_2 = C_{node} - C_{12} - C_{23} = 318.29 - 15 - 15 = 288.29 \text{ pF}$$

$$C_3 = C_{node} - C_{23} = 318.29 - 15 = 303.29 \text{ pF}$$

ดังนั้นจากรูปที่ 3.2 แบบทั่วไปของวงจรกรองความถี่ผ่านเมื่อเรานำค่าอุปกรณ์ที่คำนวณได้มาใส่ลงในวงจรจะได้วงจรกรองห้วงความถี่ 150 MHz ดังรูปที่ 3.3



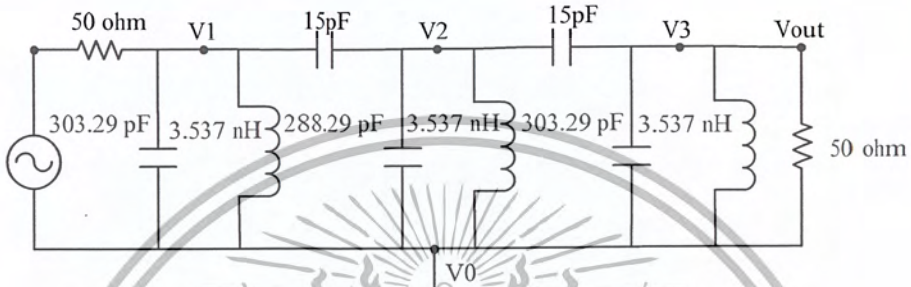
รูปที่ 3.2 รูปแบบทั่วไปของวงจรกรองความถี่ผ่านแบบ Coupled Resonator



รูปที่ 3.3 วงจรกรองความถี่ผ่าน 150 MHz

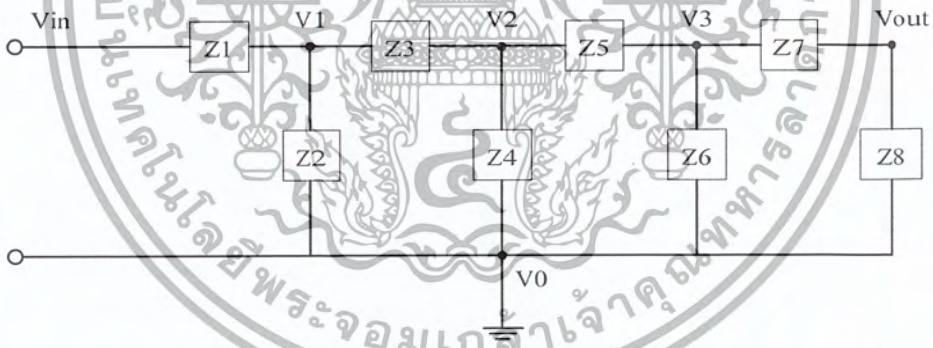
การหา ฟังก์ชันส่งผ่านของวงจรกรองช่วงความถี่ 150 MHz สามารถหาได้ดังนี้
 วิธีทำ จากรูปวงจรกรองช่วงความถี่ 150 MHz ที่ได้จากการคำนวณข้างต้นเรา
 สามารถหา ฟังก์ชันส่งผ่านได้ตามขั้นตอนดังนี้

1) ทำการกำหนดโหนดลงในวงจร



รูปที่ 3.4 วงจรกรองความถี่ผ่านหลังจากการกำหนดโหนด

2) ให้มองเป็นบล็อกของ Impedance เพื่อการวิเคราะห์ที่ง่ายขึ้นจะได้ดังภาพด้านล่าง



รูปที่ 3.5 วงจร Transmission Line

3) ใช้ Node Analysis ในการวิเคราะห์หา ซึ่งจะได้สมการดังนี้

$$\frac{V_1 - V_{in}}{Z_1} + \frac{V_1}{Z_2} + \frac{V_1 - V_2}{Z_3} = 0 \tag{3.1}$$

$$\frac{V_2 - V_1}{Z_3} + \frac{V_2}{Z_4} + \frac{V_2 - V_3}{Z_5} = 0 \tag{3.2}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\frac{V_3 - V_2}{Z_5} + \frac{V_3}{Z_6} + \frac{V_3 - V_{out}}{Z_7} = 0 \quad (3.3)$$

$$\frac{V_{out} - V_3}{Z_7} + \frac{V_{out}}{Z_8} = 0 \quad (3.4)$$

โดย

$$Z_1 = R_s \quad (3.5)$$

$$Z_2 = \frac{S^2 C_1 L_1 + 1}{S L_1} \quad (3.6)$$

$$Z_3 = \frac{1}{S C_{12}} \quad (3.7)$$

$$Z_4 = \frac{S^2 C_2 L_2 + 1}{S L_2} \quad (3.8)$$

$$Z_5 = \frac{1}{S C_{23}} \quad (3.9)$$

$$Z_6 = \frac{S^2 C_3 L_3 + 1}{S L_3} \quad (3.10)$$

$$Z_7 = \frac{1}{S C_{34}} \quad (3.11)$$

$$Z_8 = R_L \quad (3.12)$$

4) เมื่อทำการนำสมการทั้งหมดมาหาความสัมพันธ์ จะได้ ฟังก์ชันส่งผ่านดังนี้

$$H(z) = \frac{V_{out}}{V_{in}} = \frac{(2.481 \times 10^{23}) S^3}{S^6 + (1.257 \times 10^8) S^5 + (2.673 \times 10^{18}) S^4 + (2.235 \times 10^{26}) S^3 + (2.374 \times 10^{36}) S^2 + (9.915 \times 10^{43}) S + (7.009 \times 10^{53})}$$

การออกแบบวงจรกรองความถี่ผ่าน 400 MHz

ความถี่กึ่งกลาง (Center Frequency) 400 MHz

BW ที่ 3 dB 27 MHz

ฟังก์ชันที่ใช้ในการออกแบบ Butterworth Function order 3

จากข้อกำหนดในการออกแบบวงจรกรองความถี่ที่ใช้ฟังก์ชัน Butterworth order 3,

$$BW = 20 \text{ MHz}, f_c = 400 \text{ MHz}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.2 Butterworth Capacitive Coupled Resonators

n	q_1	q_n	k_{12}	k_{23}	k_{34}	k_{45}	k_{56}	k_{67}	k_{78}
2	1.414	1.414	0.707						
3	1.000	1.000	0.707	0.707					
4	0.765	0.765	0.841	0.541	0.841				
5	0.618	0.618	1.000	0.556	0.556	1.000			
6	0.518	0.518	1.169	0.605	0.518	0.605	1.169		
7	0.445	0.445	1.342	0.667	0.527	0.527	0.667	1.342	
8	0.390	0.390	1.519	0.736	0.554	0.510	0.554	0.736	1.519

วิธีทำ จากตารางที่ 3.2 เราสามารถหาค่าพารามิเตอร์ q และ k (order $n=3$) ได้ค่าต่างๆ ดังนี้

$$q_1 = 1.000$$

$$q_2 = 1.000$$

$$k_{12} = 0.707$$

$$k_{23} = 0.707$$

จากข้อกำหนดในการออกแบบสามารถหาค่า Q ได้จากความสัมพันธ์

$$Q_{BPF} = \frac{f_0}{BW_{3dB}} = \frac{400 \text{ MHz}}{20 \text{ MHz}} = 20$$

ดังนั้น $Q_1 = Q_{BPF} \times q_1 = 20 \times 1.000 = 20$

$$Q_3 = Q_{BPF} \times q_3 = 20 \times 1.000 = 20$$

$$K_{12} = \frac{k_{12}}{Q_{BPF}} = \frac{0.707}{20} = 0.03535$$

$$K_{23} = \frac{k_{23}}{Q_{BPF}} = \frac{0.707}{20} = 0.03535$$

กำหนดค่า $R_s = 50 \Omega$ และ $R_L = 50 \Omega$ ดังนั้นสามารถคำนวณหาค่า L ได้จากความสัมพันธ์

$$L = \frac{R_{L,S}}{\omega_0 Q_{1,3}} = \frac{50}{2\pi(400 \times 10^6)(20)} = 0.995 \times 10^{-9} \approx 1 \text{ nH}$$

ดังนั้นเราสามารถคำนวณหาค่า C_{node} ได้จากความสัมพันธ์

$$C_{node} = \frac{1}{\omega_0^2 L} = \frac{1}{[(2\pi)(400 \times 10^6)]^2 (0.995 \times 10^{-9})} = 159.11 \text{ pF}$$

จากค่า C_{node} เราสามารถหาค่า $C_{Coupling}$ ได้จากความสัมพันธ์

$$C_{12} = K_{12}C_{node} = (0.03535)(159.11 \times 10^{-12}) = 5.62 \text{ pF}$$

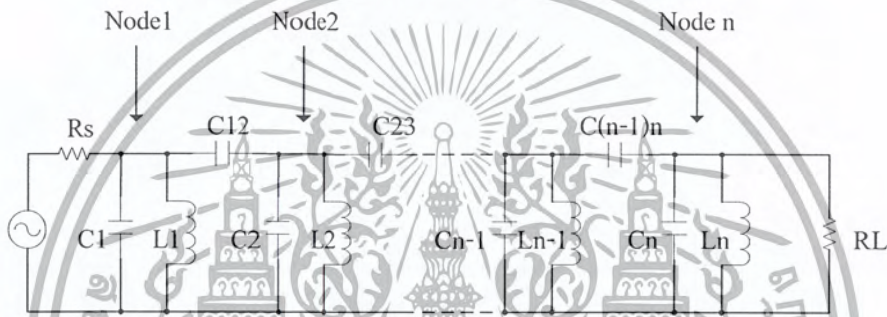
$$C_{23} = K_{23}C_{node} = (0.03535)(159.11 \times 10^{-12}) = 5.62 \text{ pF}$$

ส่วนค่าตัวเก็บประจุที่ต่อขนาน สามารถหาได้จากความสัมพันธ์

$$C_1 = C_{node} - C_{12} = 159.11 \text{ pF} - 5.62 \text{ pF} = 153.49 \text{ pF}$$

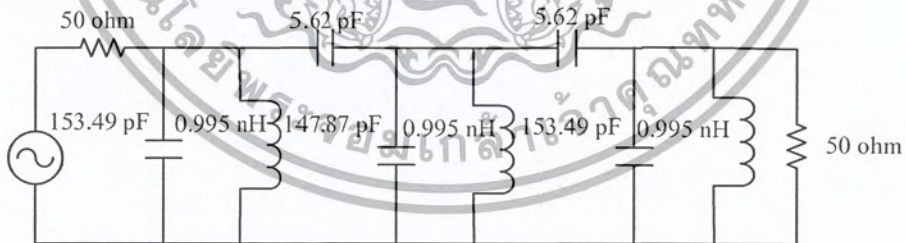
$$C_2 = C_{node} - C_{12} - C_{23} = 159.11 \text{ pF} - 5.62 \text{ pF} - 5.62 \text{ pF} = 147.87 \text{ pF}$$

$$C_3 = C_{node} - C_{23} = 159.11 \text{ pF} - 5.62 \text{ pF} = 153.49 \text{ pF}$$



รูปที่ 3.6 รูปแบบทั่วไปของวงจรกรองความถี่ผ่านแบบ Coupled Resonator

ดังนั้นจากรูปแบบทั่วไปของวงจรกรองความถี่ผ่านเมื่อเรานำค่าอุปกรณ์ที่คำนวณได้มาใส่ลงในวงจรจะได้วงจรกรองช่วงความถี่ 400 MHz ดังรูปที่ 3.7



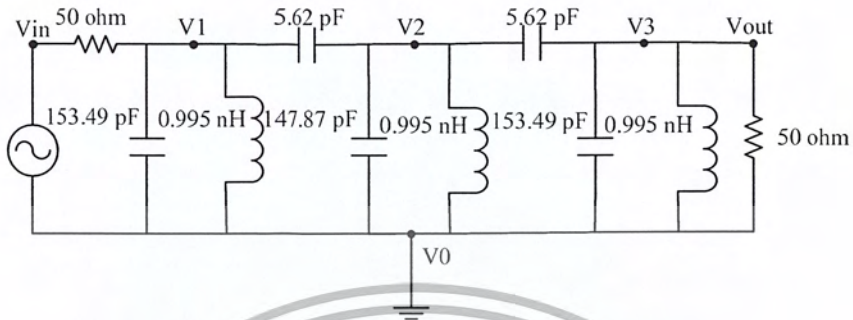
รูปที่ 3.7 วงจรกรองความถี่ผ่าน 400 MHz

การหา ฟังก์ชันส่งผ่านของวงจรกรองช่วงความถี่ 400 MHz สามารถหาได้ดังนี้

วิธีทำ จากรูปวงจรกรองช่วงความถี่ 400 MHz ที่ได้จากการคำนวณข้างต้นเราสามารถหา ฟังก์ชันส่งผ่านได้ตามขั้นตอนดังนี้

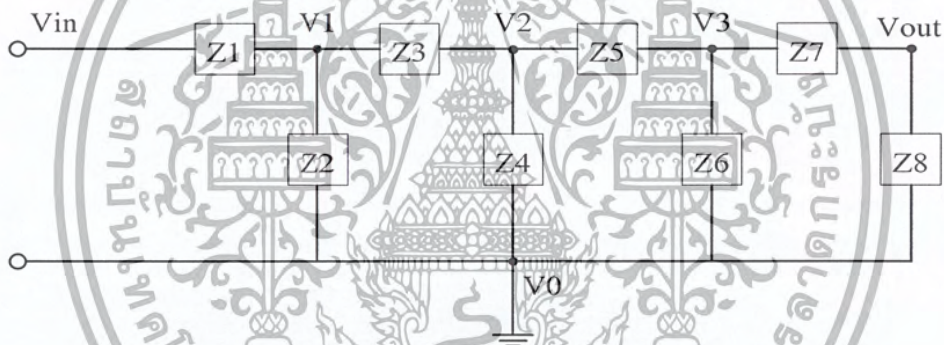
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1) ทำการกำหนดโหนดลงในวงจร



รูปที่ 3.8 วงจรกรองความถี่ผ่านหลังจากการกำหนดโหนด

2) ให้มองเป็นบล็อกของ Impedance เพื่อการวิเคราะห์ที่ง่ายขึ้นจะได้ดังภาพด้านล่าง



รูปที่ 3.9 วงจร Transmission Line

3) ใช้ Node Analysis ในการวิเคราะห์หา ซึ่งจะได้สมการดังนี้

$$\frac{V_1 - V_{in}}{Z_1} + \frac{V_1}{Z_2} + \frac{V_1 - V_2}{Z_3} = 0 \tag{3.13}$$

$$\frac{V_2 - V_1}{Z_3} + \frac{V_2}{Z_4} + \frac{V_2 - V_3}{Z_5} = 0 \tag{3.14}$$

$$\frac{V_3 - V_2}{Z_5} + \frac{V_3}{Z_6} + \frac{V_3 - V_{out}}{Z_7} = 0 \tag{3.15}$$

$$\frac{V_{out} - V_3}{Z_7} + \frac{V_{out}}{Z_8} = 0 \tag{3.16}$$

โดย

$$Z_1 = R_s \tag{3.17}$$

$$Z_2 = \frac{S^2 C_1 L_1 + 1}{S L_1} \tag{3.18}$$

$$Z_3 = \frac{1}{S C_{12}} \tag{3.19}$$

$$Z_4 = \frac{S^2 C_2 L_2 + 1}{S L_2} \tag{3.20}$$

$$Z_5 = \frac{1}{S C_{23}} \tag{3.21}$$

$$Z_6 = \frac{S^2 C_3 L_3 + 1}{S L_3} \tag{3.22}$$

$$Z_7 = \frac{1}{S C_{34}} \tag{3.23}$$

$$Z_8 = R_L \tag{3.24}$$

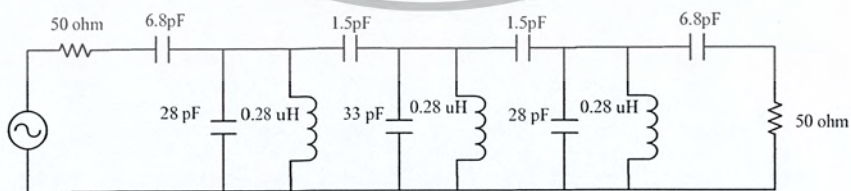
4) เมื่อทำการนำสมการทั้งหมดมาหาความสัมพันธ์ จะได้ ฟังก์ชันส่งผ่านดังนี้

$$H(z) = \frac{V_{out}}{V_{in}} = \frac{(1.984 \times 10^{24}) S^3}{S^6 + (2.513 \times 10^8) S^5 + (1.898 \times 10^{19}) S^4 + (3.177 \times 10^{27}) S^3 + (1.199 \times 10^{38}) S^2 + (1.003 \times 10^{46}) S + (2.52 \times 10^{56})}$$

วงจรกรองช่วงความถี่ 150 MHz (ปรับปรุ้ง)

วงจรกรองช่วงความถี่ 150 MHz ที่ได้จากการออกแบบด้วยโปรแกรม AAE Filter

Design จะมีรูปวงจรที่ได้จากการออกแบบ ดังรูปที่ 3.10

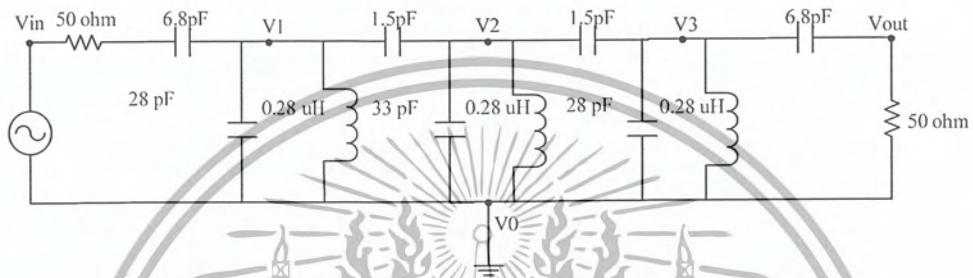


รูปที่ 3.10 วงจรกรองความถี่ผ่าน 150 MHz

การหา พังก์ชันส่งผ่านของวงจรกรองช่วงความถี่ 150 MHz (ปรับปรุง) สามารถหาได้ดังนี้

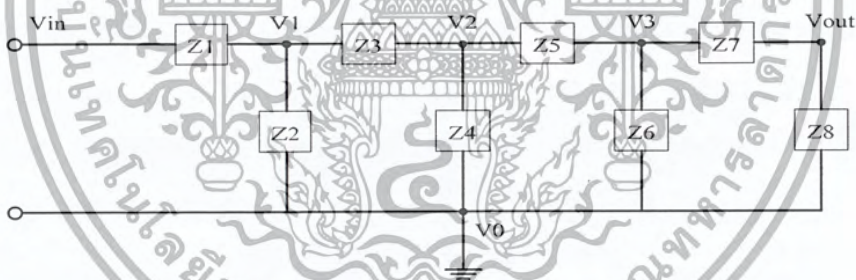
วิธีทำ จากรูปวงจกรองช่วงความถี่ 150 MHz ที่ได้จากการออกแบบด้วยโปรแกรม AAE Filter Design สามารถหา พังก์ชันส่งผ่านได้ตามขั้นตอนดังนี้

1) ทำการกำหนดโหนดลงในวงจร



รูปที่ 3.11 วงจกรองความถี่ผ่านหลังจากการกำหนดโหนด

2) ให้มองเป็นบล็อกของ Impedance เพื่อการวิเคราะห์ที่ง่ายขึ้นจะได้ดังภาพด้านล่าง



รูปที่ 3.12 วงจร Transmission Line.

3) ใช้ Node Analysis ในการวิเคราะห์หา ซึ่งจะได้สมการดังนี้

$$\frac{V_1 - V_{in}}{Z_1} + \frac{V_1}{Z_2} + \frac{V_1 - V_2}{Z_3} = 0 \tag{3.25}$$

$$\frac{V_2 - V_1}{Z_3} + \frac{V_2}{Z_4} + \frac{V_2 - V_3}{Z_5} = 0 \tag{3.26}$$

$$\frac{V_3 - V_2}{Z_5} + \frac{V_3}{Z_6} + \frac{V_3 - V_{out}}{Z_7} = 0 \tag{3.27}$$

$$\frac{V_{out} - V_3}{Z_7} + \frac{V_{out}}{Z_8} = 0 \tag{3.28}$$

โดย

$$Z_1 = \frac{SR_s C + 1}{SC} \tag{3.29}$$

$$Z_2 = \frac{S^2 C_1 L_1 + 1}{SL_1} \tag{3.30}$$

$$Z_3 = \frac{1}{SC_{12}} \tag{3.31}$$

$$Z_4 = \frac{S^2 C_2 L_2 + 1}{SL_2} \tag{3.32}$$

$$Z_5 = \frac{1}{SC_{23}} \tag{3.33}$$

$$Z_6 = \frac{S^2 C_3 L_3 + 1}{SL_3} \tag{3.34}$$

$$Z_7 = \frac{1}{SC_{34}} \tag{3.35}$$

$$Z_8 = R_L \tag{3.36}$$

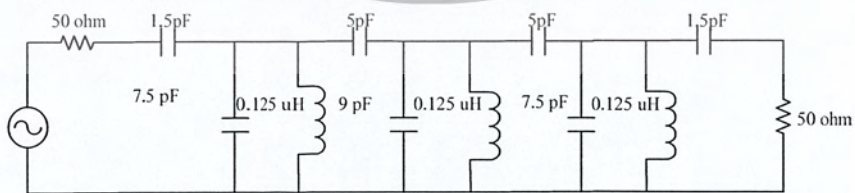
4) เมื่อทำการนำสมการทั้งหมดมาหาความสัมพันธ์ จะได้ ฟังก์ชันส่งผ่านดังนี้

$$H(z) = \frac{V_{out}}{V_m} = \frac{(8.372 \times 10^{23}) S^3}{S^6 + (1.885 \times 10^8) S^5 + (2.693 \times 10^{18}) S^4 + (3.357 \times 10^{26}) S^3 + (2.383 \times 10^{36}) S^2 + (1.487 \times 10^{46}) S + (7.009 \times 10^{53})}$$

วงจรกรองช่วงความถี่ 400 MHz (ปรับปรุง)

วงจรกรองช่วงความถี่ 400 MHz ที่ได้จากการออกแบบด้วยโปรแกรม AAE Filter

Design จะมีรูปวงจรที่ได้จากการออกแบบ ดังรูปที่ 3.13

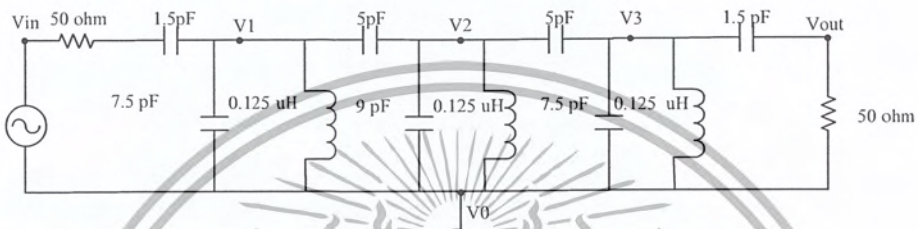


รูปที่ 3.13 วงจรกรองความถี่ผ่าน 400 MHz

การหา ฟังก์ชันส่งผ่านของวงจรกรองช่วงความถี่ 400 MHz (ปรับปรุง) สามารถหาได้ดังนี้

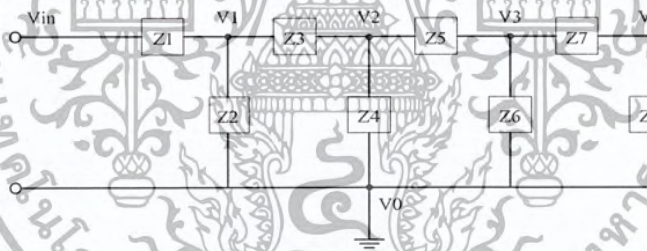
วิธีทำ จากรูปวงจกรองช่วงความถี่ 400 MHz ที่ได้จากการออกแบบด้วยโปรแกรม AADE Filter Design สามารถหา ฟังก์ชันส่งผ่านได้ตามขั้นตอนดังนี้

1) ทำการกำหนดโหนดลงในวงจร



รูปที่ 3.14 วงจกรองความถี่ผ่านหลังจากการกำหนดโหนด

2) ให้องค์ประกอบของ Impedance เพื่อการวิเคราะห์ที่ง่ายขึ้นจะได้ดังภาพด้านล่าง



รูปที่ 3.15 วงจร Transmission Line

3) ใช้ Node Analysis ในการวิเคราะห์หา ซึ่งจะได้สมการดังนี้

$$\frac{V_1 - V_{in}}{Z_1} + \frac{V_1}{Z_2} + \frac{V_1 - V_2}{Z_3} = 0 \tag{3.37}$$

$$\frac{V_2 - V_1}{Z_3} + \frac{V_2}{Z_4} + \frac{V_2 - V_3}{Z_5} = 0 \tag{3.38}$$

$$\frac{V_3 - V_2}{Z_5} + \frac{V_3}{Z_6} + \frac{V_3 - V_{out}}{Z_7} = 0 \tag{3.39}$$

$$\frac{V_{out} - V_3}{Z_7} + \frac{V_{out}}{Z_8} = 0 \tag{3.40}$$

โดย

$$Z_1 = \frac{SR_s C + 1}{SC} \quad (3.41)$$

$$Z_2 = \frac{S^2 C_1 L_1 + 1}{SL_1} \quad (3.42)$$

$$Z_3 = \frac{1}{SC_{12}} \quad (3.43)$$

$$Z_4 = \frac{S^2 C_2 L_2 + 1}{SL_2} \quad (3.44)$$

$$Z_5 = \frac{1}{SC_{23}} \quad (3.45)$$

$$Z_6 = \frac{S^2 C_3 L_3 + 1}{SL_3} \quad (3.46)$$

$$Z_7 = \frac{1}{SC_{34}} \quad (3.47)$$

$$Z_8 = R_L \quad (3.48)$$

4) เมื่อทำการนำสมการทั้งหมดมาหาความถี่พ้อง จะได้ ฟังก์ชันส่งผ่านดังนี้

$$H(z) = \frac{V_{out}}{V_{in}}$$

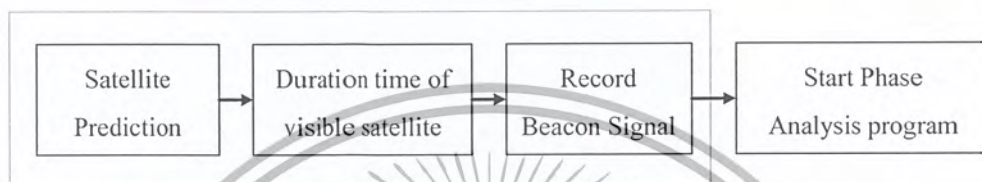
$$H(z) = \frac{V_{out}}{V_{in}} = \frac{(3.876 \times 10^{24}) S^3}{S^6 + (3.142 \times 10^8) S^5 + (1.9 \times 10^9) S^4 + (3.973 \times 10^{27}) S^3 + (1.2 \times 10^{38}) S^2 + (1.253 \times 10^{46}) S + (2.52 \times 10^{56})}$$

3.3 โปรแกรมควบคุมการทำงานของระบบ

จากรูปที่ 3.16 โปรแกรมนี้ใช้ในการควบคุมการทำงานของระบบด้วยภาษาไพธอน เนื่องจากดาวเทียมไม่ได้โคจรผ่านบริเวณที่เราสนใจตลอดเวลา เราจึงอาศัยสคริปของไพธอนชื่อ tle2azel ซึ่งเป็นโปรแกรมที่ใช้ในการคำนวณเวลาการโคจรของดาวเทียมในการคำนวณช่วงเวลาการโคจรผ่านของดาวเทียมในบริเวณที่สนใจหรือบริเวณสถานีฐาน จากนั้นโปรแกรมจะคอยเช็คเวลาปัจจุบันกับเวลาที่ดาวเทียมโคจรผ่าน เพื่อทำการบันทึกสัญญาณบีคอนที่รับได้จากดาวเทียม เมื่อโปรแกรมทำการบันทึกสัญญาณเสร็จเรียบร้อยแล้วนั้น โปรแกรมหาค่าเฟสจะเริ่มทำงานต่อไป

โดยไพธอนสคริป tle2azel นี้อาศัยหลักการทำงานของโมดูล Ephem โมดูลนี้เป็นโมดูลเกี่ยวกับการคำนวณทางดาราศาสตร์ โดยสามารถรับค่า TLE ไปคำนวณหาค่าตำแหน่งของ

ดาวเทียมในช่วงเวลาต่างๆ รวมทั้งข้อมูลเกี่ยวกับวงโคจรของดาวเทียม อาทิ มุมกวาด (Azimuth) มุมเงย (Elevation) เป็นต้น ในการคำนวณหาตำแหน่งวงโคจรของดาวเทียมนั้นจะอ้างอิงจากแบบจำลอง SGP4 (Simplified General Perturbation No.4) และ SDP4 (Simplified Deep Space General Perturbation No.4)



รูปที่ 3.16 บล็อกโคจรและแผนการทำงานของโปรแกรมควบคุมการบันทึกสัญญาณบีกอนและควบคุมการคำนวณค่าเฟส

3.3.1 รายละเอียดการทำงานในแต่ละบล็อก

1) คำนวณการโคจรผ่านของดาวเทียม (Satellite Prediction)

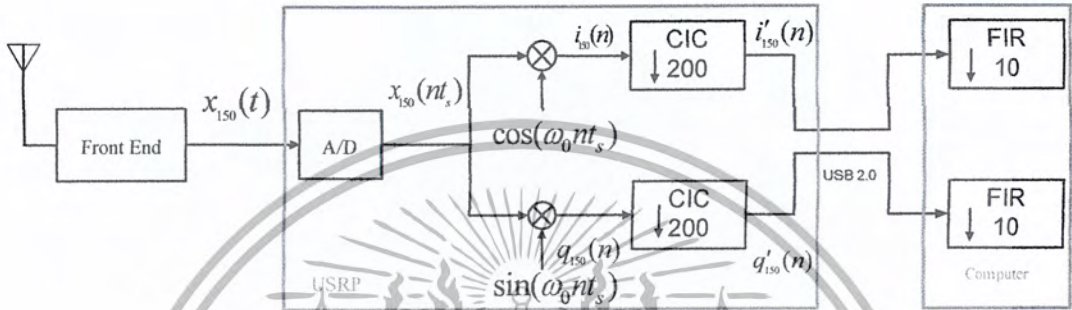
โปรแกรมเริ่มทำการคำนวณช่วงเวลาของการโคจรผ่านของดาวเทียมในบริเวณที่สนใจในช่วงเวลาหนึ่งเก็บเป็นไฟล์ข้อความ (Textfile) โดยอาศัยไพธอนสคริปชื่อ tle2azel.py ใช้ร่วมกับค่า TLE ซึ่งเป็นค่าชุดตัวเลขอธิบายวงโคจรของดาวเทียมแต่ละดวง โปรแกรมจะทำการอ่านค่า TLE จากเว็บไซต์ (<http://www.rish.kyoto-u.ac.jp/digitalbeacon/information/cosmic.tle>) และเก็บเป็นเท็กซ์ไฟล์

2) พิจารณาการโคจรผ่านของดาวเทียมในแต่ละรอบ

โปรแกรมจะทำการอ่านค่าช่วงเวลาการโคจรผ่านของดาวเทียมที่คำนวณได้โดยเลือกเฉพาะช่วงเวลาที่มีความมุมเงยของการโคจรผ่านมากกว่า 20 องศา แล้วทำการเรียงลำดับเวลาในการโคจรผ่านเข้ามายังบริเวณที่เราสนใจเก็บไว้ในตัวแปรลิสต์ จากนั้น โปรแกรมจะทำการตรวจเช็คเวลาปัจจุบันว่าอยู่ในช่วงเวลาที่ดาวเทียมโคจรผ่านบริเวณที่เราสนใจหรือไม่ ในกรณีที่อยู่ในช่วงเวลาที่ดาวเทียมโคจรผ่านในบริเวณที่เราสนใจโปรแกรมจะเริ่มทำการบันทึกสัญญาณดาวเทียม

3) บันทึกลับสัญญาณความถี่ม (สัญญาณบีคอน)

เมื่อช่วงเวลาปัจจุบันมีการ โครจรผ่านของดาวเทียมที่เราสนใจ โปรแกรมจะสั่งให้ USRP เริ่มทำการบันทึกค่าสัญญาณความถี่มที่รับได้ผ่านส่วนของ Front End โดยเริ่มบันทึกสัญญาณที่รับได้ทั้ง 2 ความถี่ 150 MHz และ 400 MHz ดังรูปที่ 3.17 และ รูปที่3.18 ตามลำดับ



รูปที่ 3.17 บล็อกการรับและบันทึกสัญญาณความถี่ 150 MHz

จากรูปที่ 3.17 สัญญาณความถี่ 150 MHz ผ่านส่วนของ Front End แล้วจะเข้าสู่การประมวลผลข้อมูลทางดิจิทัล สมมติให้สัญญาณที่รับได้เป็น $x_{150MHz}(t)$ ถูกแปลงสัญญาณจากแอนะล็อกเป็นดิจิทัลโดยใช้การซีกสัญญาณแบบ Under Sampling ซึ่งมีค่าอัตราการซีกสัญญาณ (f_s) เท่ากับ 64 MSamples/s จะได้เป็นสัญญาณ $x_{150MHz}(nt_s)$ ดังสมการที่ 3.49

$$x_{150MHz}(t) = A \cos(\omega_0 t + \phi) + A \sin(\omega_0 t + \phi) \tag{3.49}$$

โดยที่ $\omega_0 = 2\pi f_0$ และ $f_0 = 150MHz$ จะได้

$$x(nt_s) = A \cos(\omega_0 nt_s + \phi) + A \sin(\omega_0 nt_s + \phi) \tag{3.50}$$

โดยที่ $t_s = \frac{1}{f_s}$

จะได้เอาต์พุตเป็นสัญญาณความถี่ 22 MHz เป็นสัญญาณไม่ต่อเนื่อง (Discrete Time) จากนั้นสัญญาณจะถูกนำไปคูณกับสัญญาณดิสครีต (Discrete) $\sin(\omega_0 nt_s)$ และ $\cos(\omega_0 nt_s)$ ที่สร้างจาก NCO (Numerically Controlled Oscillator) โดยผลของการคูณจะได้สัญญาณเบสแบนด์ I Q ดังสมการที่ 3.51

$$i(n) = x(nt_s) \times \cos(\omega_0 nt_s) \tag{3.51}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\begin{aligned}
&= [A \cos(\omega_0 n t_s + \phi) + A \sin(\omega_0 n t_s + \phi)] \times \cos(\omega_0 n t_s) \\
&= A \cos(\omega_0 n t_s + \phi) \times \cos(\omega_0 n t_s) + A \sin(\omega_0 n t_s + \phi) \cos(\omega_0 n t_s) \\
&= \frac{A}{2} [\cos(2\omega_0 n t_s + \phi) + \cos(\phi)] + \frac{A}{2} [\sin(2\omega_0 n t_s + \phi) + \sin(\phi)] \\
&= \frac{A}{2} [\cos(\phi) + \sin(\phi)] + \frac{A}{2} [\cos(2\omega_0 n t_s + \phi) + \sin(2\omega_0 n t_s + \phi)]
\end{aligned}$$

ในทำนองเดียวกันสัญญาณเราจะได้สัญญาณ $q(n)$ ดังสมการ 3.52

$$q(n) = x(n t_s) \times \sin(\omega_0 n t_s) \quad (3.52)$$

$$\begin{aligned}
&= A \cos(\omega_0 n t_s + \phi) \times \sin(\omega_0 n t_s) + A \sin(\omega_0 n t_s + \phi) \sin(\omega_0 n t_s) \\
&= \frac{A}{2} [\sin(2\omega_0 n t_s + \phi) - \sin(\phi)] + \frac{A}{2} [\cos(\phi) - \cos(2\omega_0 n t_s + \phi)] \\
&= \frac{A}{2} [\cos(\phi) - \sin(\phi)] + \frac{A}{2} [\sin(2\omega_0 n t_s + \phi) - \cos(2\omega_0 n t_s + \phi)]
\end{aligned}$$

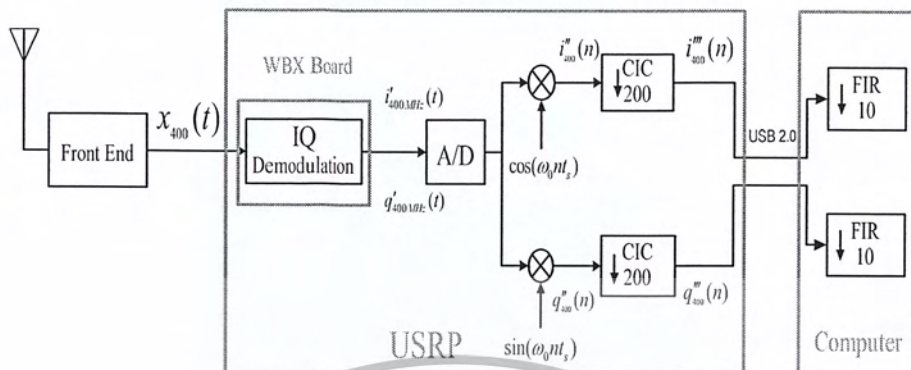
จากนั้นสัญญาณจะผ่าน CIC Filter ที่ทำหน้าที่เปรียบเสมือนวงจรกรองแถบความถี่ต่ำผ่าน (Low Pass filter) โดยมีการทำดาวยน์แซมปลิงด้วยเฟคเตอร์ที่มีค่า 200 ดังนั้นทำให้สัญญาณมีอัตราความถี่แซมปลิงเท่ากับ 320 kS/s จะได้สัญญาณที่มีองค์ประกอบความถี่ย่านเบสแบนด์ดังสมการที่ 3.53 และ 3.54

$$i'_{150\text{MHz}}(n) = \frac{A}{2} [\cos(\phi) + \sin(\phi)] \quad (3.53)$$

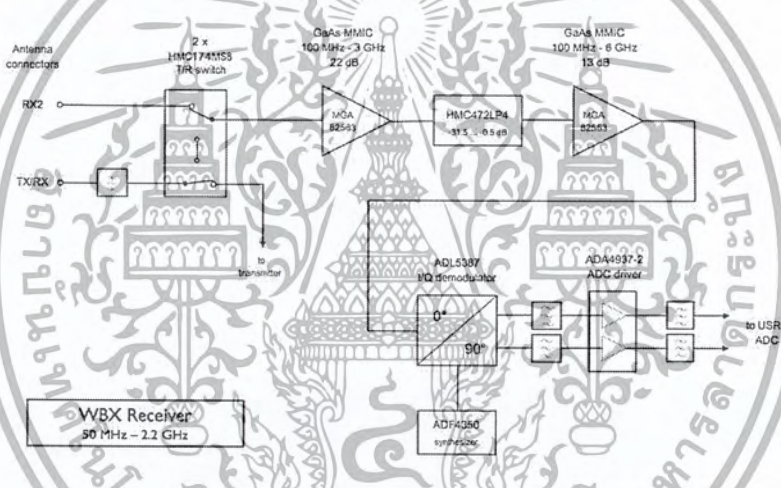
$$q'_{150\text{MHz}}(n) = \frac{A}{2} [\cos(\phi) - \sin(\phi)] \quad (3.54)$$

ในทำนองเดียวกันการรับและบันทึกสัญญาณความถี่ 400 MHz สามารถทำได้ดังบล็อกไดอะแกรมรูปที่ 3.18

เมื่อสัญญาณผ่านส่วน Front End จะเข้าสู่ USRP ซึ่งในกรณีนี้เราจะรับสัญญาณผ่าน Daughter Board ซึ่งในส่วนนี้จะทำหน้าที่คล้ายกับส่วน Front End ของ USRP โดยที่ WBX Board นี้ มีโครงสร้างดังรูปที่ 3.19



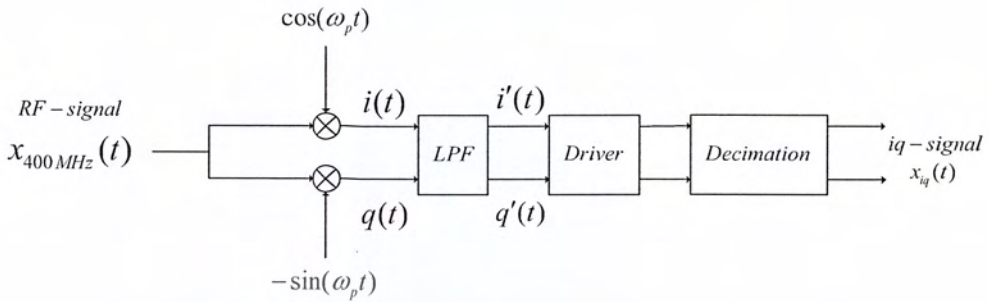
รูปที่ 3.18 บล็อกการรับและบันทึกสัญญาณ 400 MHz



รูปที่ 3.19 บล็อกการทำงานของ WBX Board

จะเห็นได้ว่าเมื่อสัญญาณผ่านเข้ามายังบอร์ดนั้น Daughter Board จะทำ IQ-Demodulation ดังรูปที่3.20 จะทำการสร้างสัญญาณ $\cos(\omega_p t)$ และ $-\sin(\omega_p t)$ ที่มีความถี่ใกล้เคียงกับความถี่เป้าหมาย (400MHz) จากนั้นนำสัญญาณที่ได้ไปคูณกับสัญญาณที่รับได้ โดยเราให้สัญญาณที่รับได้เป็น $x_{400MHz}(t)$ จะได้ว่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.20 บล็อกการทำงานของ WBX Board ในการ Demodulation IQ Signal

$$x_{400MHz}(t) = A \cos(\omega_c t + \phi) + A \sin(\omega_c t + \phi) \tag{3.55}$$

จะได้สัญญาณ $i(t)$ เป็น

$$\begin{aligned} i(t) &= x_{400} \cos(\omega_p t) \\ &= A \cos(\omega_c t + \phi) \cos(\omega_p t) + A \sin(\omega_c t + \phi) \cos(\omega_p t + \phi) \\ &= \frac{A}{2} \{ \cos[(\omega_c + \omega_p)t + \phi] + \cos[(\omega_c - \omega_p)t + \phi] \} \\ &\quad + \frac{A}{2} \{ \sin[(\omega_c + \omega_p)t + \phi] + \sin[(\omega_c - \omega_p)t + \phi] \} \end{aligned} \tag{3.56}$$

และ $q(t)$

$$\begin{aligned} q(t) &= x_{400} [-\sin(\omega_p t)] \\ &= A \cos(\omega_c t + \phi) [-\sin(\omega_p t)] + A \sin(\omega_c t + \phi) [-\sin(\omega_p t + \phi)] \\ &= -\frac{A}{2} \sin[(\omega_c + \omega_p)t + \phi] + \frac{A}{2} \sin[(\omega_c - \omega_p)t + \phi] \\ &\quad + \frac{A}{2} \cos[(\omega_c + \omega_p)t + \phi] - \frac{A}{2} \cos[(\omega_c - \omega_p)t + \phi] \end{aligned} \tag{3.57}$$

จากนั้นสัญญาณ $i(t)$ และ $q(t)$ จะผ่านเข้าสู่วงจรกรองความถี่ต่ำผ่านเพื่อตัดองค์ประกอบความถี่สูงออกจะได้สัญญาณดังสมการที่ 3.58

$$i'(t) = i'_{400MHz}(t) = \frac{A}{2} \{ \cos[(\omega_c - \omega_p)t + \phi] + \sin[(\omega_c - \omega_p)t + \phi] \} \tag{3.58}$$

ในที่นี้เราให้ $\omega_c - \omega_p = \Delta\omega$ จะได้

$$= \frac{A}{2} \{ \cos[\Delta\omega t + \phi] + \sin[\Delta\omega t + \phi] \}$$

และ

$$\begin{aligned}
 q'(t) &= q'_{400\text{MHz}}(t) = \frac{A}{2} \{ \sin[(\omega_c - \omega_p)t + \phi] - \cos[(\omega_c - \omega_p)t + \phi] \} \\
 &= \frac{A}{2} \{ \sin[\Delta\omega t + \phi] + \cos[\Delta\omega t + \phi] \}
 \end{aligned} \quad (3.59)$$

จากนั้นสัญญาณจะถูกส่งผ่านไปยังการแปลงสัญญาณจากสัญญาณแอนะล็อกเป็นสัญญาณดิจิทัล และนำสัญญาณที่ได้ไปคูณกับสัญญาณ $\sin \Delta\omega t$ และ $\cos \Delta\omega t$ จาก NCO ดังสมการที่ 3.60 และ 3.61

$$i''_{400\text{MHz}}(nt_s) = i'(nt_s) \cos(\Delta\omega nt_s) \quad (3.60)$$

$$\begin{aligned}
 &= \frac{A}{2} \{ \cos[\Delta\omega nt_s + \phi] + \sin[\Delta\omega nt_s + \phi] \} \cos(\Delta\omega nt_s) \\
 &= \frac{A}{2} [\cos(\Delta\omega nt_s + \phi) \cos(\Delta\omega nt_s) + \sin(\Delta\omega nt_s + \phi) \cos(\Delta\omega nt_s)] \\
 &= \frac{A}{4} \{ \cos[\Delta\omega nt_s + \phi] + \cos \phi \} + \frac{A}{4} \{ \sin[2\Delta\omega nt_s + \phi] + \sin \phi \}
 \end{aligned}$$

และ

$$q''_{400\text{MHz}}(nt_s) = \frac{A}{2} [\sin(\Delta\omega nt_s + \phi) - \cos(\Delta\omega nt_s + \phi)] \sin(\Delta\omega nt_s) \quad (3.61)$$

$$\begin{aligned}
 &= \frac{A}{2} [\sin(\Delta\omega nt_s + \phi) \sin(\Delta\omega nt_s)] - \frac{A}{2} [\cos(\Delta\omega nt_s + \phi) \sin(\Delta\omega nt_s)] \\
 &= \frac{A}{4} [-\cos(\Delta 2\omega nt_s + \phi) + \cos \phi] - \frac{A}{4} [\sin(\Delta 2\omega nt_s + \phi) + \sin \phi]
 \end{aligned}$$

จากนั้นสัญญาณจะถูกกรองและลดอัตราข้อมูล (Decimator) ลงอีกครั้งด้วย CIC Filter คล้ายกับกรณีสัญญาณความถี่ 150 MHz จะได้สัญญาณสัญญาณที่ส่งผ่านสาย USB เข้าสู่เครื่องคอมพิวเตอร์ดังสมการที่ 3.62 และ 3.63

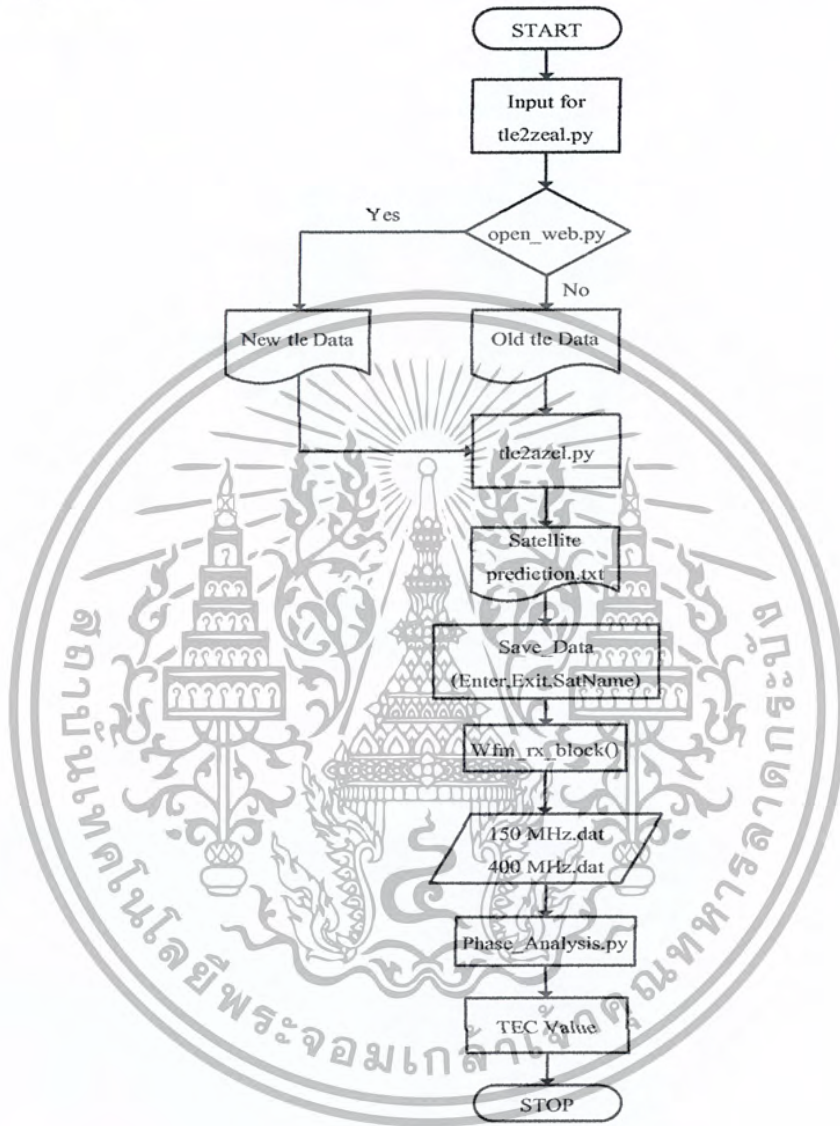
$$i'''_{400\text{MHz}}(nt_s) = \frac{A}{4} (\cos \phi + \sin \phi) \quad (3.62)$$

$$q'''_{400\text{MHz}}(nt_s) = \frac{A}{4} (\cos \phi + \sin \phi) \quad (3.63)$$

จากนั้นโปรแกรมจะทำการกรองสัญญาณด้วย FIR Filter ซึ่งในที่นี้เราทำการกรองเพื่อเลือกช่วงของสัญญาณที่รับได้เนื่องจากสัญญาณที่ส่งมาจากดาวเทียมนั้นมีค่าความคลาดเคลื่อนของความถี่ต่างกัน วงจรกรองนี้จะทำการคัดกรองเฉพาะช่วงบริเวณนั้น

โปรแกรมจะทำการบันทึกค่าสัญญาณที่ได้เป็นไบนารีไฟล์ (.dat) จากนั้นจะทำการสั่งให้โปรแกรมคำนวณหาค่าเฟสเริ่มทำงาน

ภาพรวมของโปรแกรมเมื่อมีการโคจรผ่านของดาวเทียมสามารถแสดงได้ดัง
 โฟลว์ชาร์ทรูปที่ 3.21



รูปที่ 3.21 โฟลว์ชาร์ทภาพรวมการทำงานของโปรแกรม

จากรูปที่ 3.21 เมื่อโปรแกรมเริ่มทำงาน โปรแกรมจะทำการสร้างและเก็บข้อมูลที่ใช้ในการคำนวณการโคจรผ่านของสัญญาณดาวเทียมก่อน จากนั้นจะไปสั่งให้เริ่มคำนวณการโคจรผ่านของดาวเทียม และจะเก็บข้อมูลการโคจรผ่านของดาวเทียมออกมาเป็นไฟล์ข้อมูล (Text File) จากนั้นโปรแกรมจะนำข้อมูลที่คำนวณได้มาทำการคัดเลือก ว่าช่วงเวลาการโคจรผ่านของ

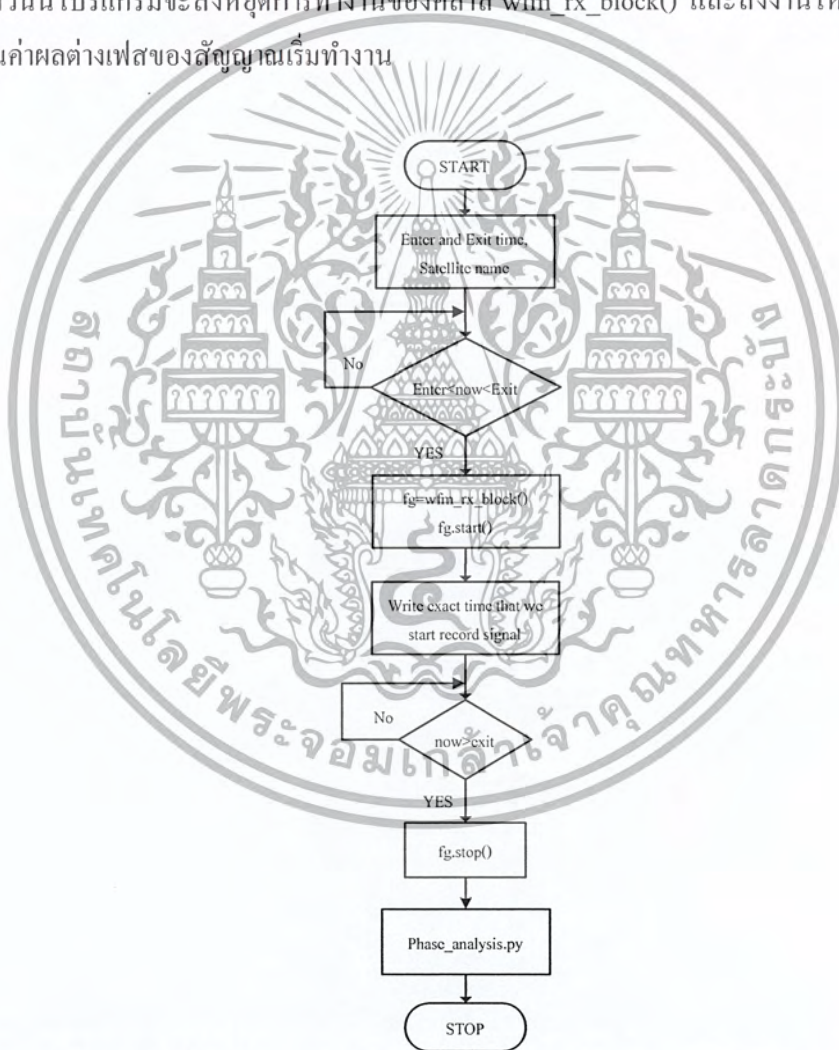
ดาวเทียมช่วงโคจรที่เหมาะสม โดยในที่นี้เราทำการคัดเลือกเฉพาะการโคจรผ่านของดาวเทียมที่มีค่าของมุมเงยที่มากกว่า 20 องศาเท่านั้น จากนั้น โปรแกรมจะทำการจัดเรียงเวลาในการโคจรผ่านของดาวเทียมในแต่ละวัน จากนั้นจะคอยตรวจสอบช่วงเวลาปัจจุบันว่ามีการโคจรผ่านของดาวเทียมเข้ามาหรือไม่ ในกรณีที่มีการโคจรผ่านของดาวเทียมนั้น โปรแกรมจะสั่งงานให้ฟังก์ชัน `save_date(enter,exit,satname)` นี้ทำงาน



รูปที่ 3.22 โฟลว์ชาร์ทการทำงานหลักของโปรแกรมหลัก

โดยฟังก์ชัน `save_date(enter,exit,satname)` นี้จะทำการรับอาร์กิวเมนต์ (Argument) 3 ค่าในการทำงาน นั่นคือ `enter` (เวลาที่ดาวเทียมโคจรเข้ามายังบริเวณที่เราสนใจ), `exit` (เวลาที่ดาวเทียมโคจรออกจากบริเวณที่เราสนใจ), `satname` (ชื่อดาวเทียมที่โคจรผ่านเข้ามายังบริเวณที่เรา

สนใจ) ดังไฟล์ชาร์ทรูปที่ 3.23 การทำงานโดยรวมของฟังก์ชันนี้จะเป็นการตรวจสอบช่วงเวลาการโคจรของดาวเทียมว่าอยู่ในบริเวณสถานีฐานและสั่งให้คลาส wfim_rx_block() เริ่มทำงานเมื่อรับสัญญาณเสร็จเรียบร้อยแล้วจะส่งคำสั่งให้โปรแกรมคำนวณค่าเฟสเริ่มทำงานจากไฟล์ชาร์ทรูปที่ 3.23 จะเห็นว่าฟังก์ชัน save_data() นี้จะเริ่มจากการเช็คเวลาปัจจุบันอีกครั้งว่ามีการโคจรของดาวเทียมอยู่บริเวณสถานีฐานหรือไม่ จากนั้นจะสั่งให้คลาส wfim_rx_block() เริ่มทำงาน จากนั้นจะคอยเช็คและรอเวลาการโคจรออกจากสถานีฐานของดาวเทียม เมื่อดาวเทียมโคจรออกจากบริเวณสถานีฐานแล้วนั้นโปรแกรมจะสั่งหยุดการทำงานของคลาส wfim_rx_block() และสั่งงานให้โปรแกรมคำนวณค่าผลต่างเฟสของสัญญาณเริ่มทำงาน



รูปที่ 3.23 ไฟล์ชาร์ทแสดงการทำงานของฟังก์ชัน save_data(enter,exit,satname)

คลาส `wfm_rx_block()` นี้จะทำหน้าที่ควบคุม USRP เพื่อประมวลผลสัญญาณที่รับได้จากสัญญาณแอนะล็อกความถี่สูงเป็นสัญญาณเบสแบนด์ดิจิทัลดั่งที่ได้อธิบายไปข้างต้น จากนั้นจะถูกกรองด้วย FIR Filter ภายในคลาสนี้ จากนั้นจะเก็บบันทึกสัญญาณเพื่อใช้ในการคำนวณหาค่าความต่างเฟสต่อไป คลาส `wfm_rx_block()` สามารถแสดงการทำงานโดยรวมได้ดังโพลีชาร์ทรูปที่ 3.24

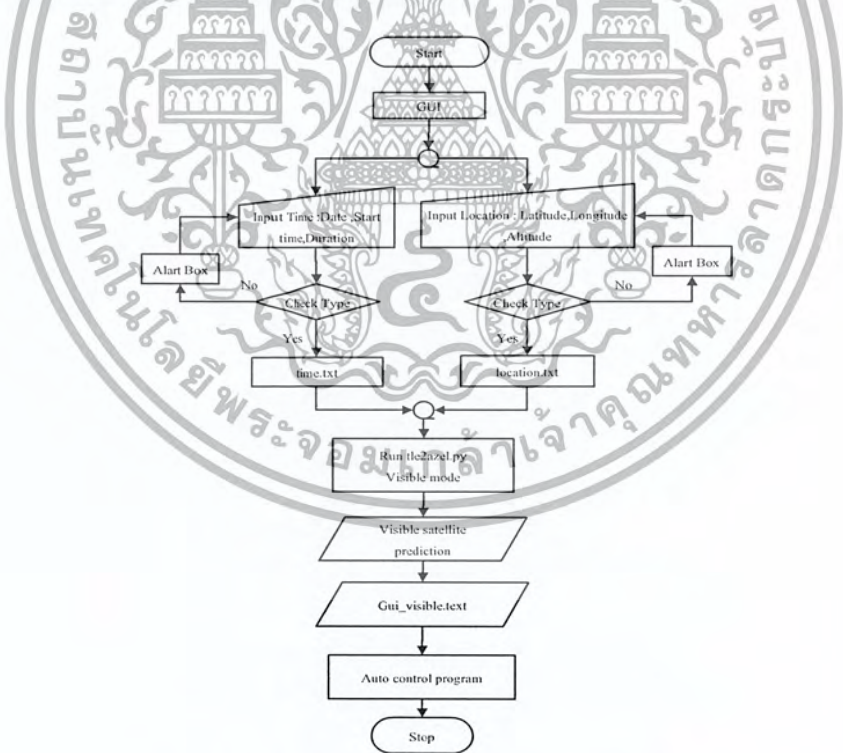


รูปที่ 3.24 โพลีชาร์ทแสดงการทำงานของคลาส `wfm_rx_block()`

3.2.2 อินเทอร์เน็ตโปรแกรมคำนวณการเดินทางผ่านของดาวเทียม และ บันทึกสัญญาณดาวเทียม

ในที่นี้เราทำการศึกษางานของโปรแกรม tle2azel.py ในการคำนวณหาการโคจรผ่านของดาวเทียมในบริเวณที่เราสนใจ ทางกลุ่มจึงจัดทำ Graphical user interface (GUI) เพื่อนำไปประยุกต์ใช้กับโปรแกรมนี้ได้สะดวกยิ่งขึ้น และ นำไปใช้เป็นส่วนหนึ่งในการส่งโปรแกรมการหาค่าเฟสประมวลผลต่อไป โดยการทำให้สร้าง GUI รับค่าตำแหน่งในบริเวณที่เราสนใจ และ ช่วงเวลาที่เราจะทำการคำนวณหาการโคจรผ่านตามไฟล์ชาร์ตดังรูปที่ 3.25

จากไฟล์ชาร์ตจะเห็นว่าเราทำการสร้าง GUI เพื่อให้ผู้ใช้นั้นสามารถระบุบริเวณที่สนใจ และ ช่วงเวลาที่จะทำการคำนวณ จากนั้นนำค่าที่ได้ไปเก็บไว้เป็นไฟล์ข้อมูล แล้วนำค่าไฟล์ที่ได้นี้ไปเป็นอินพุตให้แก่โปรแกรมที่ใช้คำนวณการโคจรผ่านของดาวเทียม และเก็บข้อมูลการโคจรผ่านในรูปแบบของไฟล์ข้อมูลเช่นกัน จากนั้นโปรแกรมจะเข้าสู่การเช็คเวลาและรอการบันทึกสัญญาณดาวเทียมในการโคจรผ่านในช่วงเวลาที่สนใจ

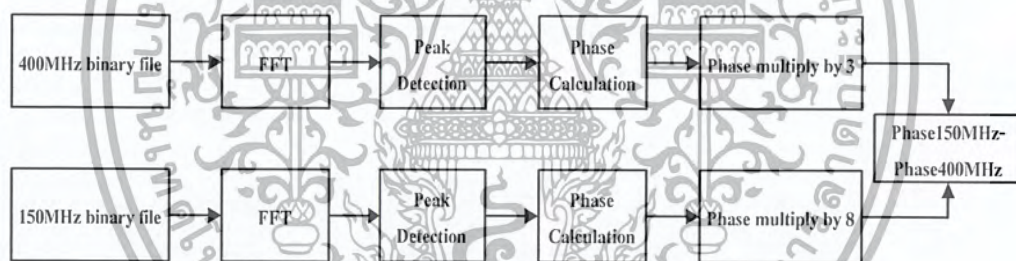


รูปที่ 3.25 ไฟล์ชาร์ตแสดงการทำงานผ่าน GUI

3.4 ภาคการประมวลผลข้อมูล / คำนวณหาค่าเฟส

การประมวลผลข้อมูลนั้น ใช้ข้อมูลที่เก็บไว้ในคอมพิวเตอร์หลังจากสัญญาณบีคอนได้ผ่าน USRP และแปลงเป็น สัญญาณเชิงซ้อน (Complex Signal) ที่เป็นดิจิทัลเบสแบนด์ เก็บไว้ในรูปไฟล์ไบนารี (Binary) สองไฟล์ คือ ไฟล์ของสัญญาณบีคอนความถี่ 150 MHz และไฟล์ของสัญญาณบีคอนความถี่ 400 MHz

จากบล็อกไดอะแกรม จะทำการอ่านข้อมูลทั้งสองไฟล์ที่อยู่ในรูปแบบของไบนารี ซึ่งทำการอ่านข้อมูลออกมาเป็นค่า เลขเชิงซ้อน (Complex Number) จากนั้นทำการ แปลง FFT (Fast Fourier Transform) โดยแปลงข้อมูลครั้งละ 4096 จุดข้อมูล ไปจนครบจำนวนข้อมูลทั้งหมด หลังจากนั้นจะทำการหาค่ากำลังสูงสุดในแต่ละครั้งที่แปลง FFT ซึ่งจะถือว่ากำลังของสัญญาณบีคอนควรมีค่ามากกว่าสัญญาณรบกวน จากนั้น จึงนำตำแหน่งที่ค่ากำลังของสัญญาณมีค่าสูงสุดมา คำนวณค่าเฟสของสัญญาณ และหาผลต่างเฟสต่อไป แสดงได้ดังรูปที่ 3.26



รูปที่ 3.26 บล็อกไดอะแกรมขั้นตอนการคำนวณผลต่างเฟสของสัญญาณบีคอน

การคำนวณผลต่างเฟสของสัญญาณบีคอน

ให้ขนาดข้อมูลของสัญญาณบีคอนมีค่าเท่ากับ

$$N_{total}$$

กำหนดให้จำนวนจุดของการแปลง FFT มีค่าเท่ากับ

$$4096$$

เพราะฉะนั้นค่า frequency resolution ของการแปลง FFT จะมีค่าเท่ากับ

$$\frac{32000}{4096} = 7.8125 \text{ Hz}$$

โดยจากไฟล์วาร์ทของโปรแกรมนั้นจะทำการแปลง FFT ต่อเนื่องไปเรื่อยๆ โดยทำการเลื่อนช่วงการแปลง FFT ไปทีละ 128 จะได้การแปลง FFT ทั้งหมด $\frac{N_{total}}{4096} \times 32$ ครั้ง โดยการ

ทำ FFT แต่ละครั้งจะให้ผลการแปลง FFT เป็นเลขเชิงซ้อนที่มีทั้งขนาด และเฟสแสดงได้จากสมการ

$$X(m) = X_{real}(m) + jX_{imag}(m) \quad (3.62)$$

โดยที่ $X(m)$ คือ ผลของการแปลง FFT ลำดับที่ m โดยค่า m มีค่าตั้งแต่ 0,1,2,3,.....,N-1

N คือจำนวนจุดของการแปลง FFT

ซึ่งผลของการแปลง FFT ที่ได้จะประกอบไปด้วยค่าขนาดและเฟสตามสมการที่

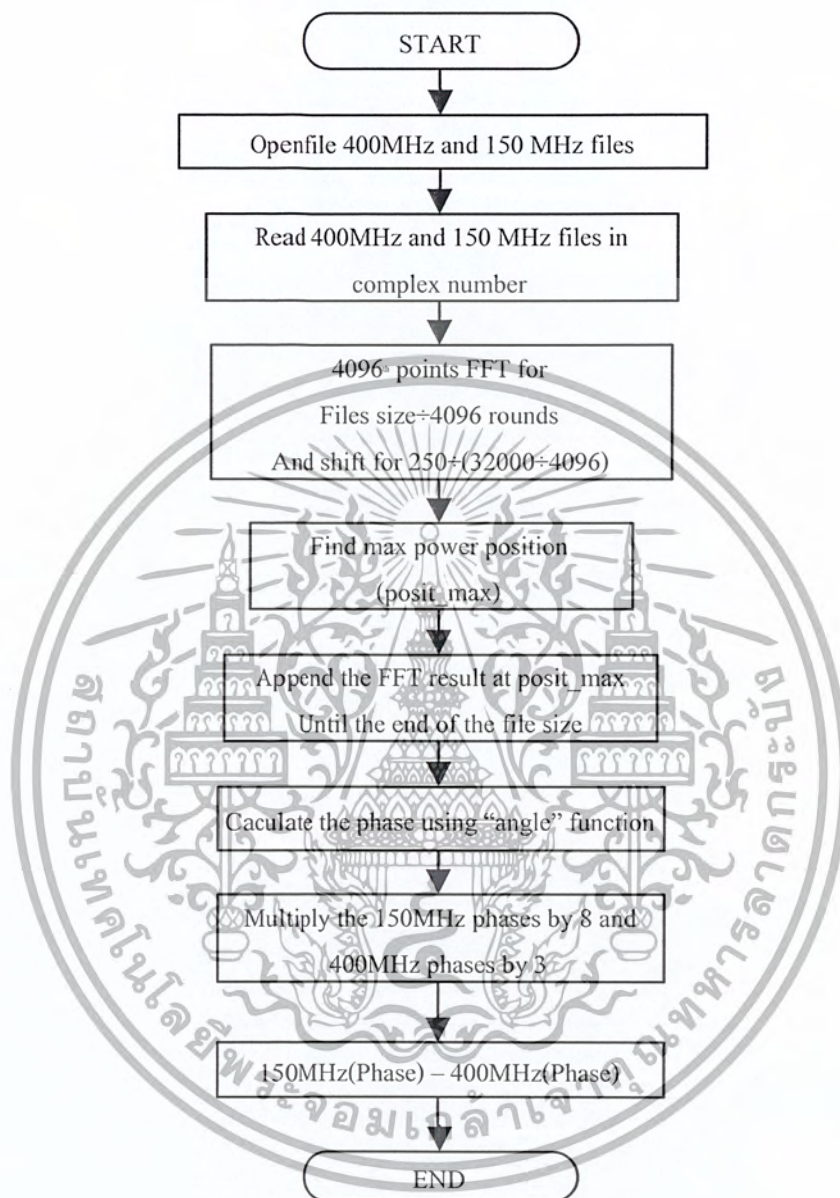
$$X_{mag}(m) = |X(m)| = \sqrt{X_{real}(m)^2 + X_{imag}(m)^2} \quad (3.63)$$

$$X_{\theta}(m) = \tan^{-1}\left(\frac{X_{imag}(m)}{X_{real}(m)}\right) \quad (3.64)$$

ซึ่งการคำนวณค่าเฟสจะใช้สมการที่ 3.64 ในการคำนวณแต่ก่อนที่จะคำนวณเฟสจะต้องทำการหาค่าสัญญาณบีคอนจากการคำนวณค่าสเปกตรัมทางกำลัง โดยจะถือว่าสัญญาณบีคอนนั้นควรจะมีกำลังสูงกว่าสัญญาณรบกวน โดยสามารถคำนวณค่าสเปกตรัมทางกำลังได้จากสมการ

$$X_{PS}(m) = X_{real}(m)^2 + jX_{imag}(m)^2 \quad (3.65)$$

โดยจะได้ค่าสูงสุดของสเปกตรัมของสัญญาณทั้งสิ้น $\frac{N_{total}}{4096} \times 32$ คิว ซึ่งจะได้ค่าเฟสจำนวน 32 ค่า เมื่อแปลง FFT ผ่านแต่ละบล็อกข้อมูลไปทำให้ค่าเฟสของสัญญาณนั้นมีอัตราเร็วของข้อมูลเท่ากับ $7.8125 \times 32 = 250\text{Hz}$ ซึ่งเพียงพอต่อการเปลี่ยนแปลงของค่าเฟสของสัญญาณที่รับได้จากนั้นนำมาคำนวณค่าเฟสตามสมการที่ 3.64 และนำค่าเฟสที่ได้มาหาความต่างเฟสซึ่งความต่างเฟสสามารถหาได้จากความสัมพันธ์ระหว่างอัตราส่วนความถี่ของสัญญาณบีคอน 150 MHz กับ 400 MHz ซึ่งความถี่อยู่ในอัตราส่วน 8:3 โดยการคำนวณเฟสก็จะต้องใช้อัตราส่วนนี้ในการคำนวณโดยการนำค่าเฟสของสัญญาณบีคอน 150MHz และ 400MHz มาคูณกับอัตราส่วน 8 และ 3 ตามลำดับ จึงจะนำมาทำการลบกันเพื่อหาความต่างเฟสได้โดยซึ่งจากการคำนวณที่กล่าวมาแล้วรวมกับบล็อกไคอะแกรมของการคำนวณความต่างเฟสจะสามารถเขียน โปรแกรมดังแสดงได้ตามไฟล์ชาร์ทรูปที่ 3.27



รูปที่ 3.27 โปรแกรมคำนวณค่าผลต่างเฟสของสัญญาณบีคอน

โดยการคำนวณค่าผลต่างเฟสนั้นมีประโยชน์ในการนำไปหาค่าความหนาแน่นอิเล็กตรอนในชั้นบรรยากาศซึ่งสามารถแสดงได้จากสมการ

$$\phi = \frac{\pi A}{f_r C} \left(\frac{1}{q^2} - \frac{1}{p^2} \right) \int N ds \quad (3.66)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยที่ ϕ คือความต่างเฟสของสัญญาณดาวเทียมบีคอน 150MHz และ 400MHz

$$A \text{ คือ ค่าคงที่มีค่าเท่ากับ } A = \frac{(2\pi e)^2}{m\epsilon_0} = 80.6m^3 / s^2$$

f_r คือค่าความถี่พื้นฐานของสัญญาณบีคอนมีค่าเท่ากับ 50 MHz

q, p คืออัตราส่วนระหว่างสัญญาณบีคอนสองความถี่โดยมีค่าเป็น 8 และ 3 ตามลำดับ

C คือค่าความเร็วแสงมีค่าประมาณ $3 \times 10^8 \text{ m/s}$

$\int Nds$ คือค่าความหนาแน่นอิเล็กตรอนในชั้นบรรยากาศ (Total Electron Content)

3.5 เครื่องมือที่ใช้ในการทดลอง

3.5.1 วงจรกรองแถบความถี่ผ่าน

เครื่องมือที่ใช้ในการทดลองมีดังนี้

Network analyzer

Spectrum Analyzer

Signal Generator

เครื่องคอมพิวเตอร์

3.5.2 โปรแกรมควบคุมการทำงาน และ ภาคการประมวลผลข้อมูล

รายการเครื่องมือที่ใช้ในการทดลอง แบ่งออกเป็นสองส่วนคือ เครื่องมือทางฮาร์ดแวร์ (Hardware) และ ซอฟต์แวร์ (Software) โดยในส่วนของฮาร์ดแวร์ คือ คอมพิวเตอร์ และ USRP ซึ่งจะใช้ในการทำงานทางด้านการประมวลผลต่างๆ ซึ่งต้องใช้ร่วมกับซอฟต์แวร์ที่ต้องการ โดยในส่วนของซอฟต์แวร์นั้น ใช้เครื่องมือภาษา ไพธอน (Python) ในการเขียนโปรแกรมเพื่อประมวลผลค่าความต่างเฟสของสัญญาณ

3.6 การจัดเก็บผลการทดลอง

3.5.1 วงจรกรองแถบความถี่ผ่าน

- 1) ทำการนำวงจรกรองความถี่ผ่านที่ได้ทำการออกแบบไปวัดอัตราการสูญเสียเนื่องจากการสะท้อน (Log Mag) กับเครื่อง Network Analyzer ทำการเก็บผลการทดลอง
- 2) ทำการนำวงจรกรองความถี่ผ่านที่ได้จากการออกแบบไปวัดค่าอัตราการขยายในการส่งผ่าน (S_{21}) ด้วยเครื่อง Network Analyzer ทำการเก็บผลการทดลอง

3) ทำการนำวงจรกรองความถี่ผ่านที่ได้จากการออกแบบไปวัด โดยจ่ายความถี่ต่างๆ เข้าไปในวงจรกรองความถี่ผ่านที่ออกแบบโดยเครื่อง Signal Generator แล้วใช้เครื่อง Spectrum Analyzer เพื่อเก็บผลค่า Characteristic ของวงจรกรองความถี่ผ่าน

4) นำค่าที่ได้จากการเก็บผลการทดลองด้วยเครื่อง Spectrum Analyzer มาพล็อตกราฟเพื่อคุณลักษณะของวงจรกรองความถี่ผ่านที่ออกแบบ

3.6.2 โปรแกรมควบคุมการทำงาน

1) ทดสอบการอ่านข้อมูลจากการเชื่อมต่ออินเทอร์เน็ต ว่าสามารถเก็บข้อมูลได้หรือไม่พร้อมทั้งในกรณีที่เครื่องคอมพิวเตอร์ไม่เชื่อมต่ออินเทอร์เน็ตอยู่นั้น โปรแกรมสามารถดำเนินการต่อไปได้หรือไม่

2) ทดลองการทำงานของโปรแกรมในการบันทึกสัญญาณจากเครื่องกำเนิดความถี่สูง และจัดเก็บเป็นแฟ้มข้อมูล

3.5.3 ภาคประมวลผลการทดลอง

รูปแบบการจัดเก็บข้อมูลนั้นจะทำการเก็บผลที่ได้จากการรันโปรแกรม โดยอ้างอิงจากไฟล์วาร์ท ของโปรแกรม โดยจะเก็บผลเรียงลำดับตามอัลกอริทึม (Algorithm) ที่ได้เขียนในไฟล์วาร์ท ไปเรื่อยๆ จนถึงค่าสุดท้ายที่ต้องการ

บทที่ 4

ผลการทดลอง

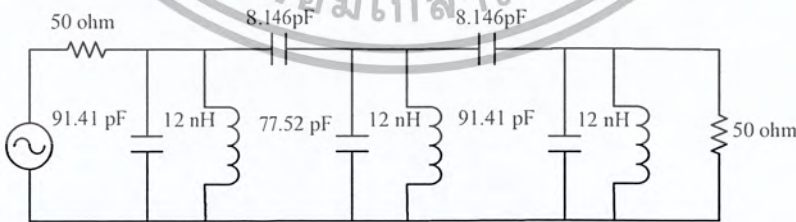
4.1 วงจรกรองแถบความถี่ผ่าน

การออกแบบวงจรกรองช่วงความถี่ผ่านนั้นเมื่อทำการออกแบบหาวงจรกรองช่วงความถี่ผ่านที่สำเร็จแล้วนั้น ในบางกรณีค่าอุปกรณ์ที่คำนวณได้กับในความเป็นจริงอาจไม่สอดคล้องกัน เนื่องจากค่าอุปกรณ์บางค่าไม่สามารถซื้อได้ตามท้องตลาดทั่วไป เช่น ค่าตัวเหนี่ยวนำ ซึ่งค่าตัวเหนี่ยวนำนี้นั้นจะหาได้เฉพาะบางค่าเท่านั้น นอกเหนือจากค่ามาตรฐานที่มีขายนั้นต้องสั่งทำโดยตรงจากโรงงานผู้ผลิต ซึ่งมีราคาแพง ดังนั้นจึงต้องมีการใช้โปรแกรมช่วยในการออกแบบให้ค่าอุปกรณ์นั้นมีความเหมาะสมกับความเป็นจริง โดยโปรแกรมที่ทางผู้จัดทำใช้ในการออกแบบนั้น คือ โปรแกรม Nuhertz Filter Solution 2009 ซึ่งสามารถใช้ช่วยในการออกแบบได้เป็นอย่างดี

4.1.1 วงจรกรองแถบความถี่ผ่าน 150 MHz

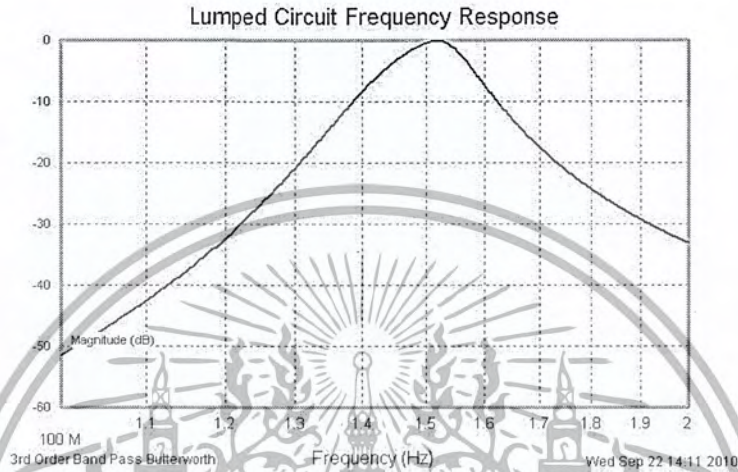
4.1.1.1 ผลการทดลองวงจรกรองความถี่ผ่าน 150 MHz ที่ได้จากการ Simulation ด้วยโปรแกรม Nuhertz Filter Solution 2009

เนื่องจากวงจรกรองความถี่ผ่านที่คำนวณได้จากรูปที่ 3 นั้น ค่าตัวเหนี่ยวนำ ($L=3.537\text{nH}$) ของวงจรกรองความถี่ผ่าน 150 MHz นั้น ไม่มีขายในท้องตลาดทั่วไป เมื่อทำการใช้โปรแกรมช่วยในการออกแบบแล้วนั้น โดยกำหนดค่าตัวเหนี่ยวนำ ($L=12\text{nH}$) ซึ่งเป็นค่ามาตรฐานที่หาซื้อได้ง่าย จะได้ว่าวงจรดังรูปที่ 4.1



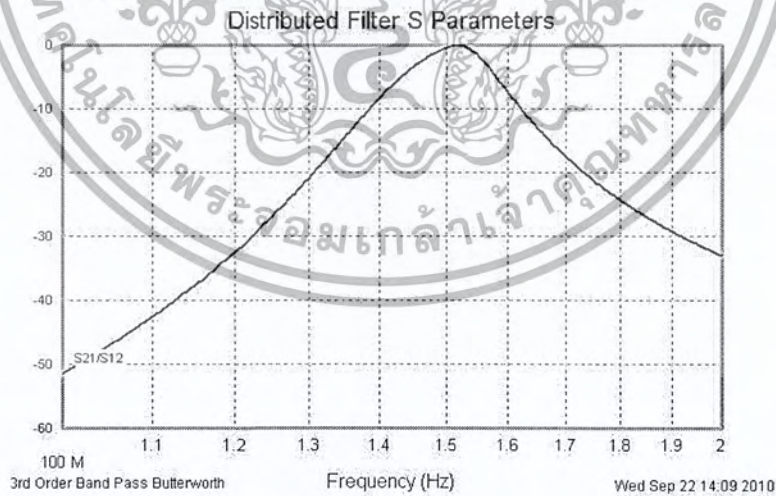
รูปที่ 4.1 วงจรกรองความถี่ผ่านที่ได้ทำการเปลี่ยนค่าตัวเหนี่ยวนำจาก 3.357 nH เป็น 12 nH

ซึ่งเมื่อสังเกตผลตอบสนองทางความถี่ที่ได้จะเห็นได้ว่าช่วงความถี่ผ่านนั้นเป็นไปตามลักษณะของวงจรที่ต้องการคือมีความถี่กึ่งกลาง 150 MHz และแบนด์วิดธ์ประมาณ 10 MHz ดังรูปที่ 4.2

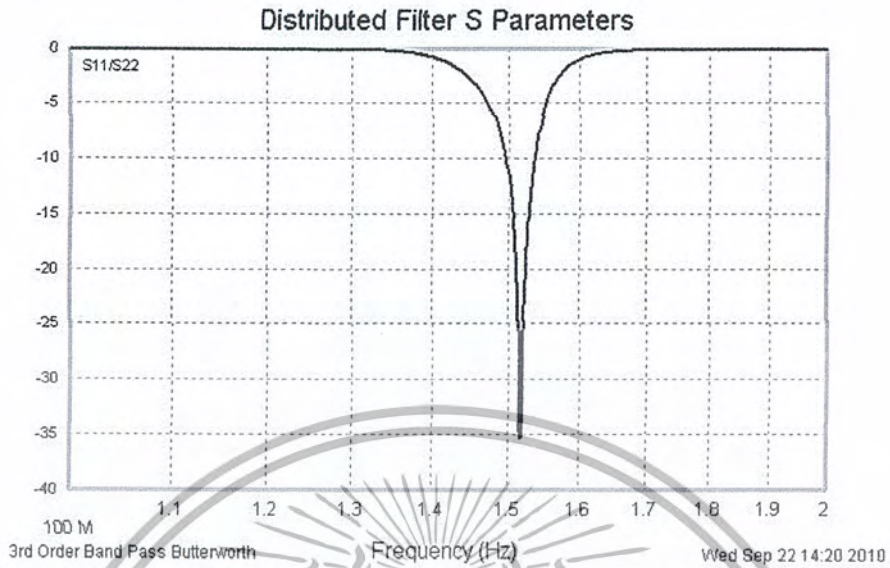


รูปที่ 4.2 ผลตอบสนองทางความถี่ของวงจรกรองความถี่ผ่าน 150 MHz

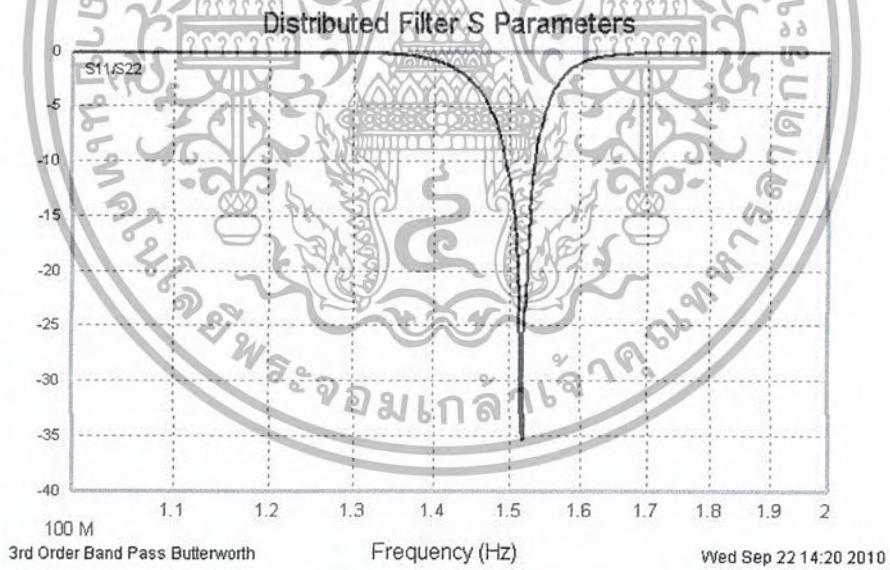
ค่าอัตราขยายในการส่งผ่าน (S_{21}) และค่าอัตราสูญเสียเนื่องจากการสะท้อน (Log Mag) จะแสดงดังรูปที่ 4.3 และ รูปที่ 4.4 ตามลำดับ



รูปที่ 4.3 ค่าอัตราขยายในการส่งผ่าน (S_{21}) ของวงจรกรองความถี่ผ่าน 150 MHz



รูปที่ 4.4 ค่าอัตราการสูญเสียเนื่องจากการสะท้อน (Log Mag) ของวงจรกรองความถี่ผ่าน 150 MHz ด้านอินพุต

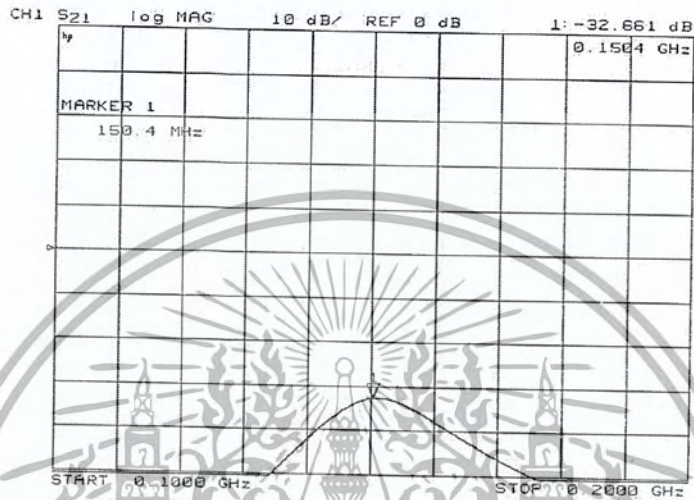


รูปที่ 4.5 ค่าอัตราการสูญเสียเนื่องจากการสะท้อน (Log Mag) ของวงจรกรองความถี่ผ่าน 150 MHz ด้านเอาต์พุต

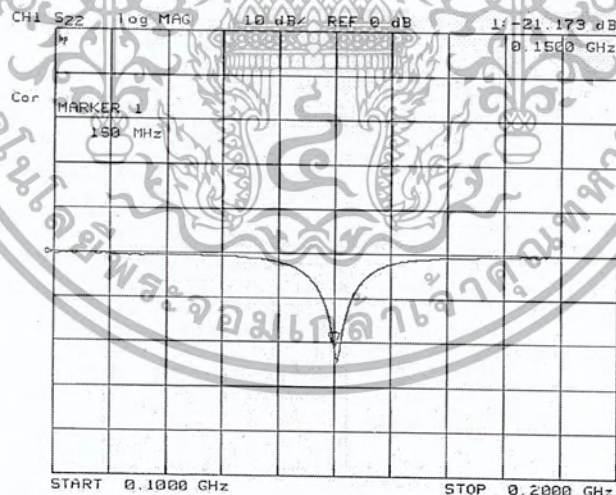
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.1.2 ผลการทดลองของวงจรกรองความถี่ผ่าน 150 MHz

เมื่อนำวงจรที่ได้จากการ Simulation ด้วยโปรแกรม Nuhertz Filter Solution 2009 มาทำการทดลองจริงแล้วนั้นจะได้ผลการทดลอง ดังนี้

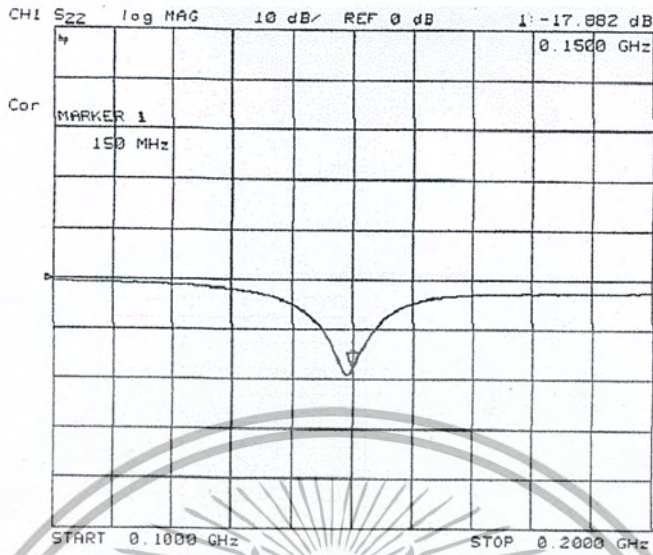


รูปที่ 4.6 ค่าอัตราการขยายในการส่งผ่าน (S_{21}) ของวงจรกรองความถี่ผ่าน 150 MHz

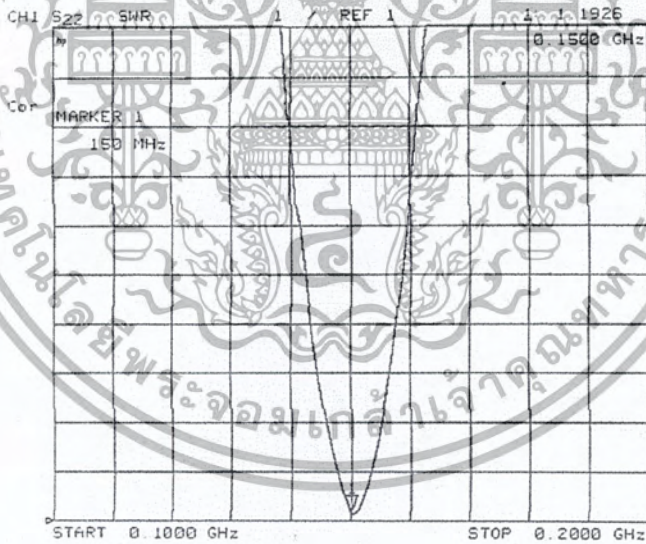


รูปที่ 4.7 ค่าอัตราการสูญเสียเนื่องจากการสะท้อน (Log Mag) ของวงจรกรองความถี่ผ่าน 150 MHz ด้านอินพุต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

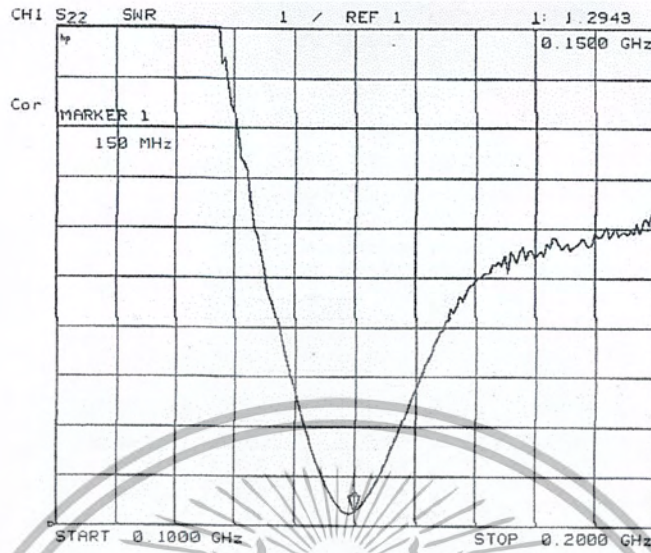


รูปที่ 4.8 ค่าอัตราความสูญเสียเนื่องจากการสะท้อน (Log Mag) ของวงจรความถี่ผ่าน 150 MHz ด้านเอาต์พุต

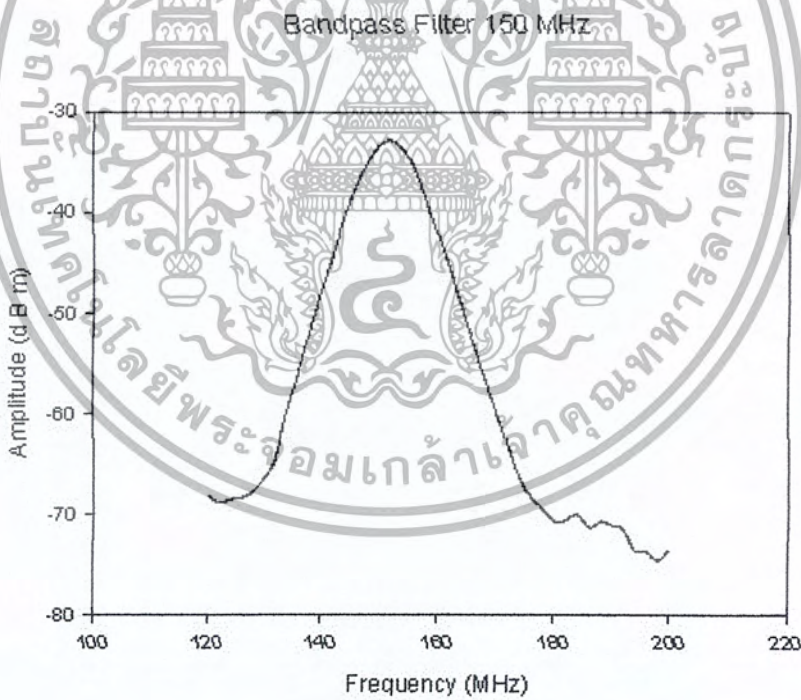


รูปที่ 4.9 ค่า SWR ของวงจรช่องความถี่ผ่าน 150 MHz ด้านอินพุต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.10 ค่า SWR ของวงจรกรองความถี่ผ่าน 150 MHz ด้านเอาต์พุต



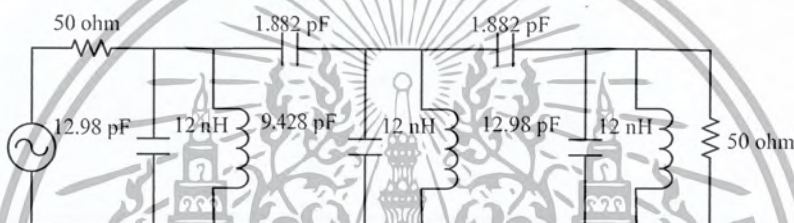
รูปที่ 4.11 ผลตอบสนองทางความถี่ของวงจรกรองความถี่ผ่าน 150 MHz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.2 วงจรกรองความถี่ผ่าน 400 MHz

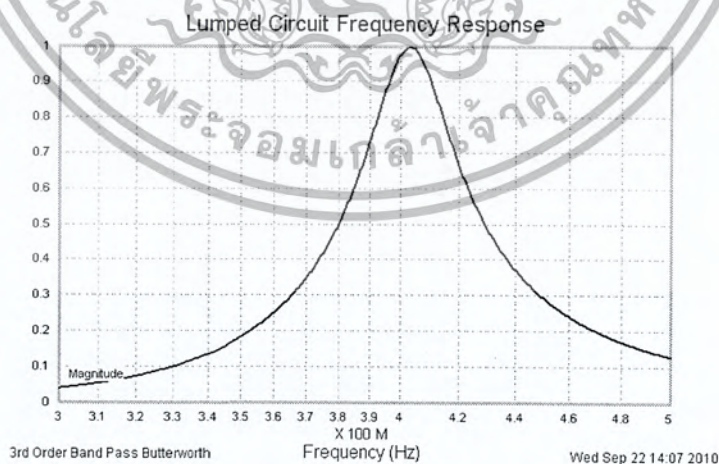
4.1.2.1 ผลการทดลองวงจรกรองความถี่ผ่าน 400 MHz ที่ได้จากการ Simulation ด้วยโปรแกรม Nuhertz Filter Solution 2009

เนื่องจากวงจรกรองความถี่ผ่านที่คำนวณได้จากบทที่ 3 นั้น ค่าตัวเหนี่ยวนำ ($L=0.995\text{nH}$) ของวงจรกรองความถี่ผ่าน 400 MHz นั้นไม่มีขายในท้องตลาดทั่วไป เมื่อทำการใช้โปรแกรมช่วยในการออกแบบแล้วนั้น โดยกำหนดค่าตัวเหนี่ยวนำ ($L=12\text{nH}$) ซึ่งเป็นค่ามาตรฐานที่หาซื้อได้ง่าย จะได้วงจรดังรูปที่ 4.12



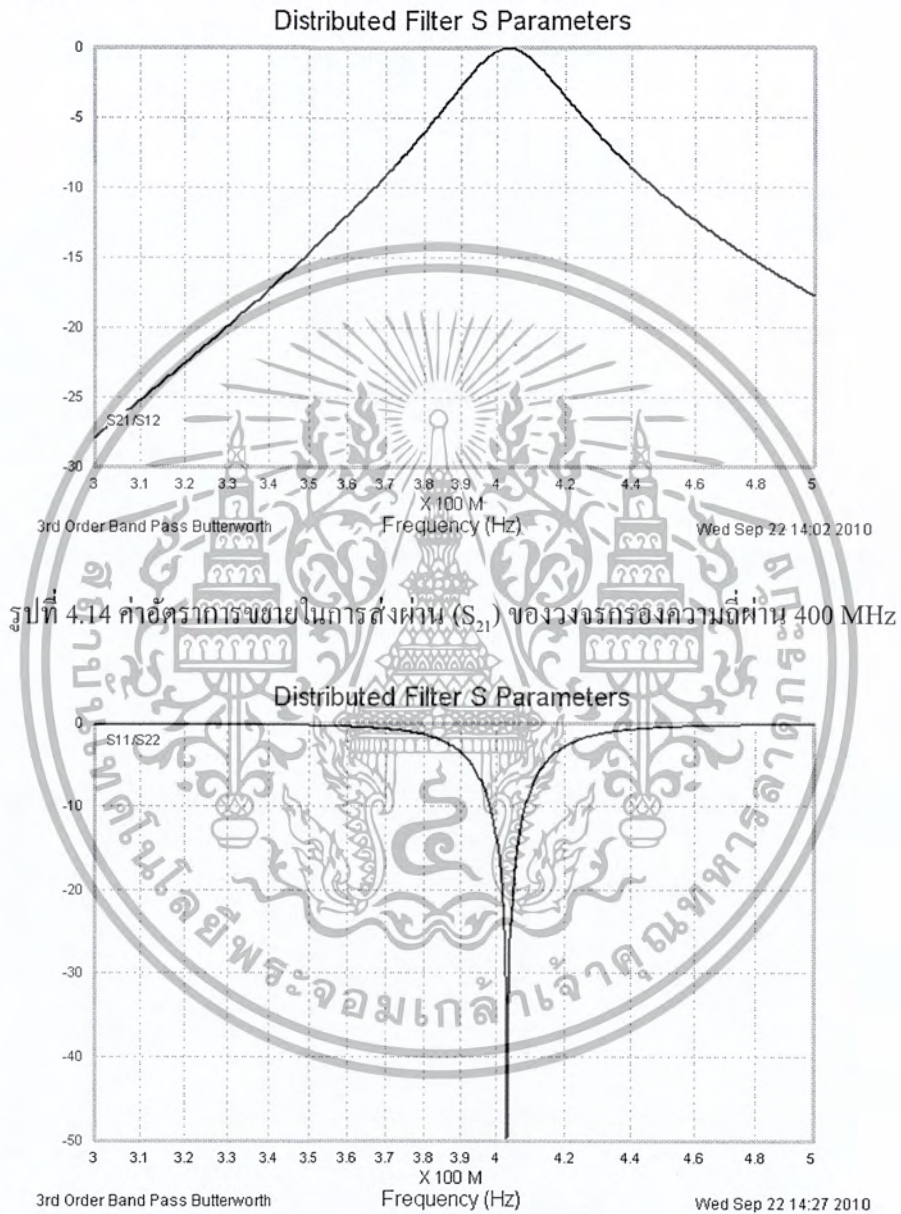
รูปที่ 4.12 วงจรกรองความถี่ผ่านที่ได้ทำการเปลี่ยนค่าตัวเหนี่ยวนำจาก 0.995 nH เป็น 12 nH

ซึ่งเมื่อสังเกตผลตอบสนองทางความถี่ที่ได้จะเห็นได้ว่าช่วงความถี่ผ่านนั้นเป็นไปตามลักษณะของวงจรที่ต้องการคือมีความถี่กึ่งกลาง 400 MHz และแบนด์วิดธ์ประมาณ 27 MHz ดังรูปที่ 4.13

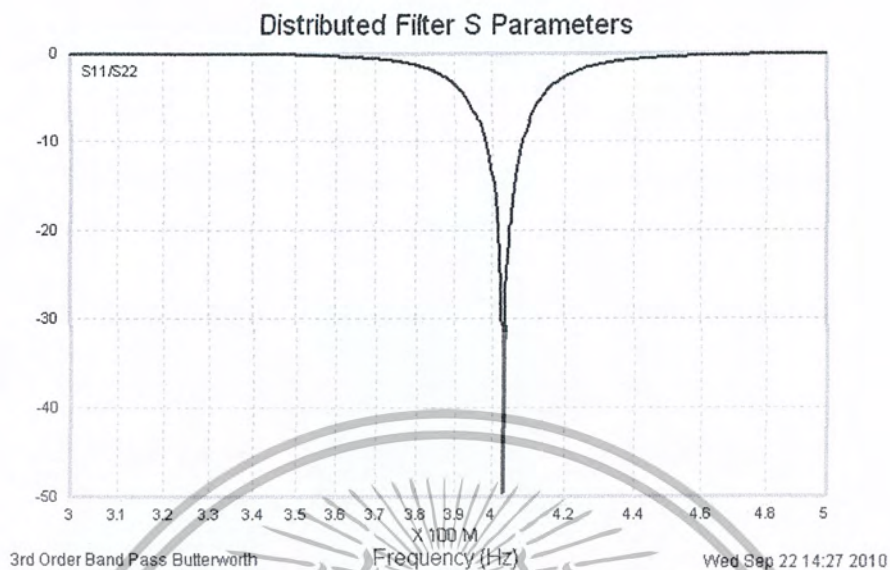


รูปที่ 4.13 ผลตอบสนองทางความถี่ของวงจรกรองความถี่ผ่าน 400 MHz

ค่าอัตราขยายในการส่งผ่าน (S_{21}) และค่าอัตราสูญเสียเนื่องจากการสะท้อน (Log Mag) จะแสดงดังรูปที่ 4.14 และ รูปที่ 4.15 ตามลำดับ



รูปที่ 4.15 ค่าอัตราสูญเสียเนื่องจากการสะท้อน (Log Mag) ของวงจรกรองความถี่ผ่าน 400 MHz ด้านอินพุต

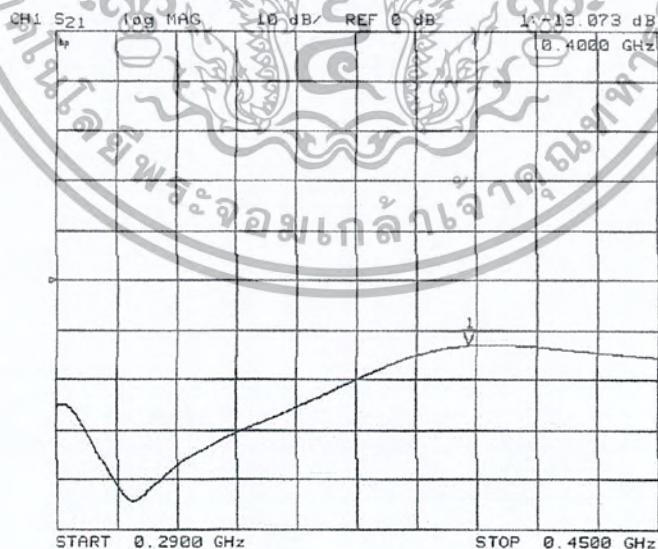


รูปที่ 4.16 ค่าอัตราการสูญเสียเนื่องจากการสะท้อน (Log Mag) ของวงจรกรองความถี่ผ่าน 400 MHz ด้านเอาต์พุต

4.1.2.2 ผลการทดลองของวงจรกรองความถี่ผ่าน 400 MHz

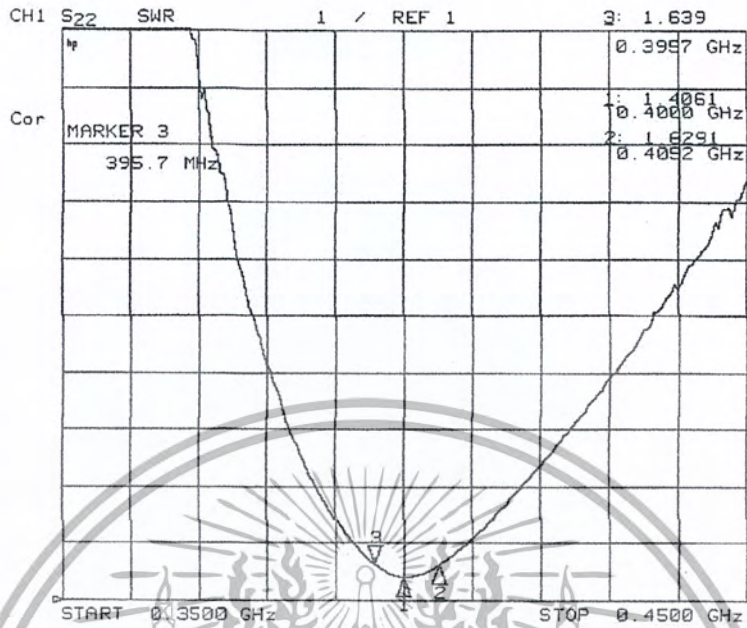
เมื่อนำวงจรที่ได้จากการ Simulation ด้วยโปรแกรม Nuhertz Filter Solution 2009

มาทำการทดลองจริงแล้วนั้นจะได้ผลการทดลอง ดังนี้

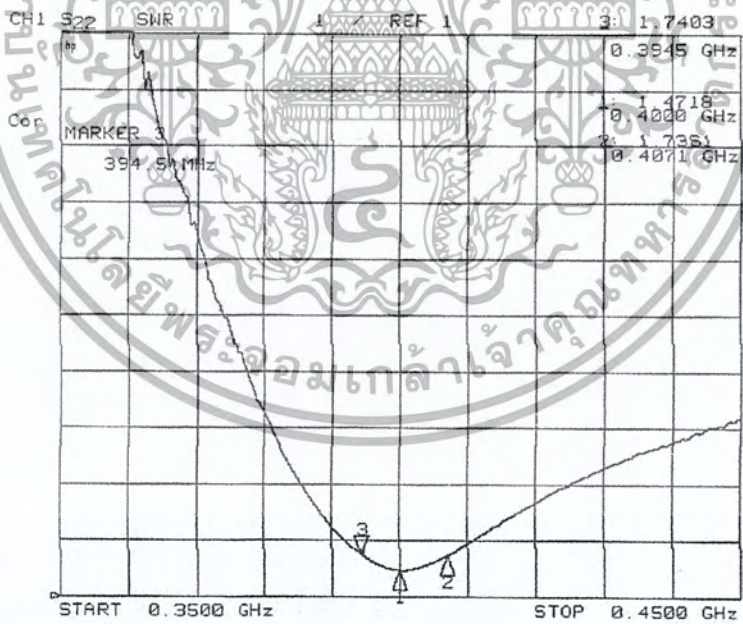


รูปที่ 4.17 ค่าอัตราการขยายในการส่งผ่าน (S_{21}) ของวงจรกรองความถี่ผ่าน 400 MHz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



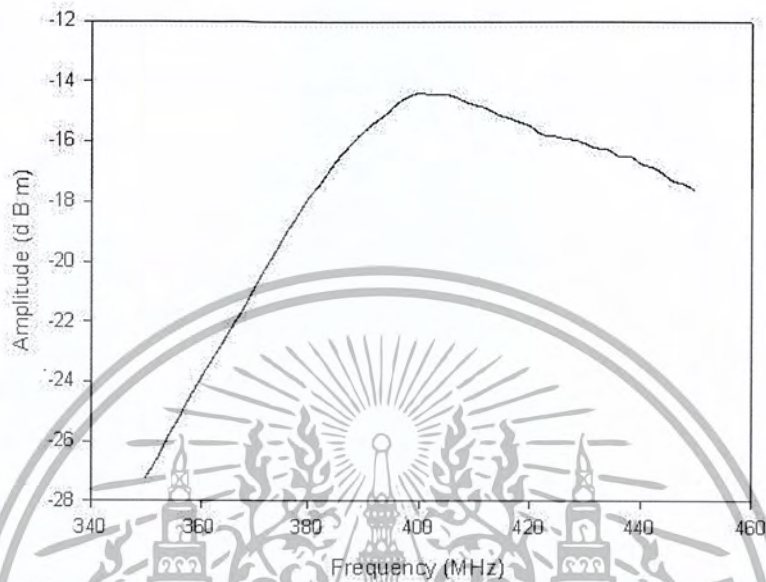
รูปที่ 4.20 ค่า SWR ของวงจรกรองช่วงความถี่ผ่าน 400 MHz ด้านอินพุต



รูปที่ 4.21 ค่า SWR ของวงจรกรองช่วงความถี่ผ่าน 400 MHz ด้านเอาต์พุต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

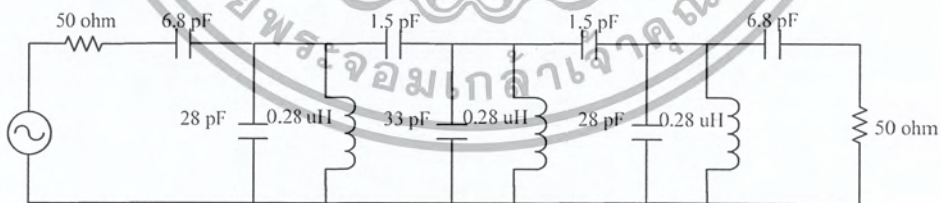
Bandpass Filter 400 MHz



รูปที่ 4.22 ผลตอบสนองทางความถี่ของวงจรกรองความถี่ผ่าน 400 MHz

4.1.3 วงจรกรองความถี่ผ่าน 150 MHz (ปรับปรุง)

เนื่องจากวงจรกรองความถี่ที่ได้ทำมาก่อนหน้านั้นมีประสิทธิภาพไม่ดีเท่าที่ควร จึงมีการออกแบบวงจรกรองความถี่ใหม่เพื่อพัฒนาวงจรกรองความถี่ให้มีประสิทธิภาพมากขึ้น ซึ่งจะได้วงจรดังรูปที่ 4.23

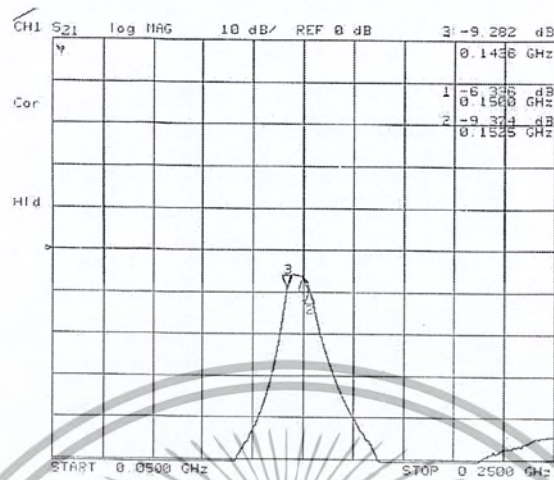


รูปที่ 4.23 วงจรกรองความถี่ 150 MHz (ปรับปรุง)

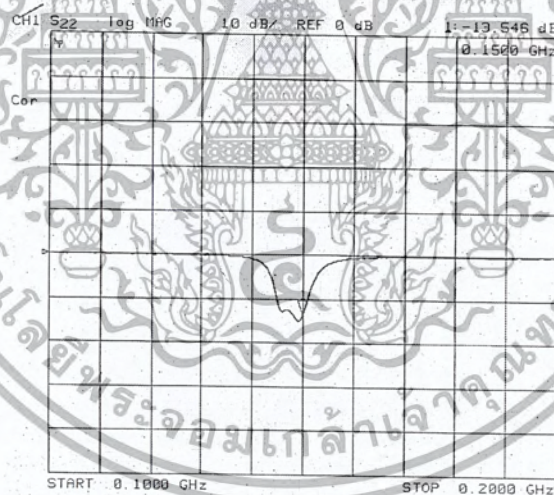
ผลการทดลองวงจรกรองความถี่ผ่าน 150 MHz (ปรับปรุง)

เมื่อนำวงจรที่ได้จากการ Simulation ด้วยโปรแกรม AADE Filter Design มาทำการทดลองจริงแล้วนั้นจะได้ผลการทดลอง ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

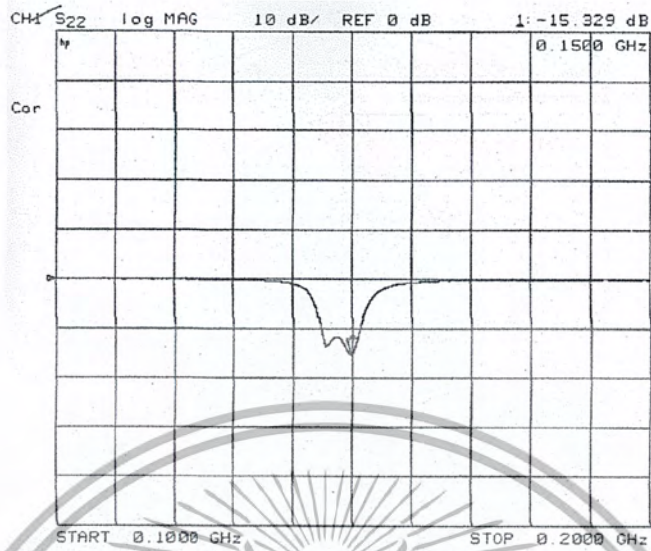


รูปที่ 4.24 ค่าอัตราขยายในการส่งผ่าน (S_{21}) ของวงจรกรองความถี่ผ่าน 150 MHz

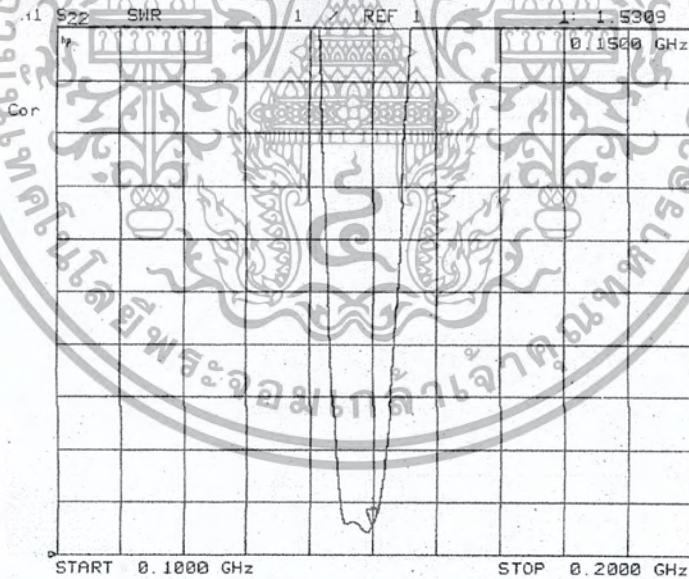


รูปที่ 4.25 ค่าอัตราการสูญเสียเนื่องจากการสะท้อน (Log Mag) ของวงจรกรองความถี่ผ่าน 150 MHz ด้านอินพุต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

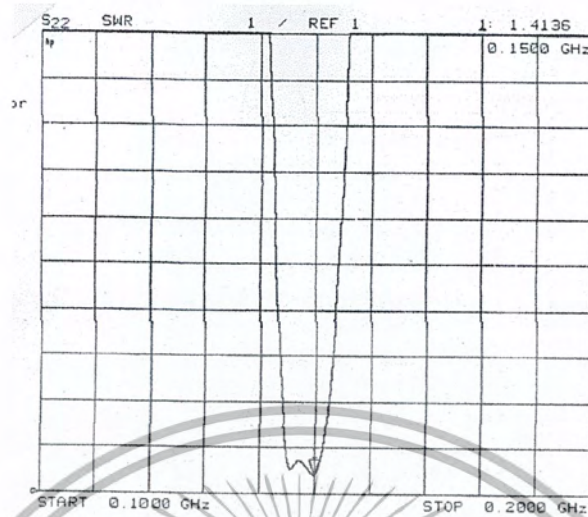


รูปที่ 4.26 ค่าอัตราการสูญเสียเนื่องจากการสะท้อน (Log Mag) ของวงจรกรองความถี่ผ่าน 150 MHz ด้านเอาต์พุต

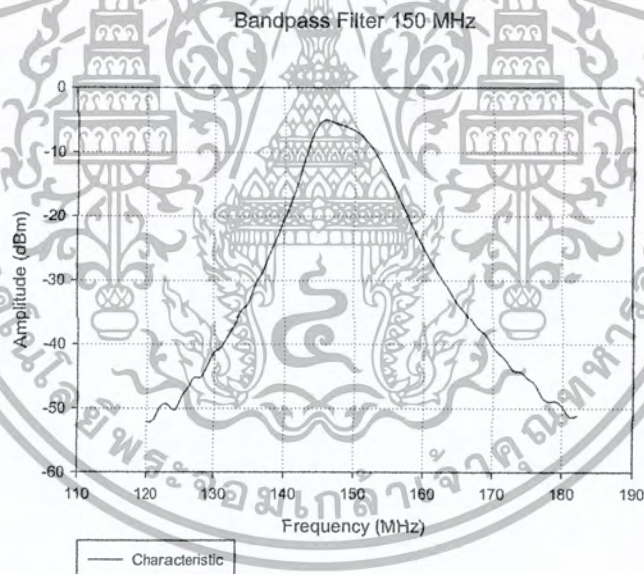


รูปที่ 4.27 ค่า SWR ของวงจรกรองช่วงความถี่ผ่าน 150 MHz ด้านอินพุต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.28 ค่า SWR ของวงจรกรองช่วงความถี่ผ่าน 150 MHz ด้านเอาต์พุต

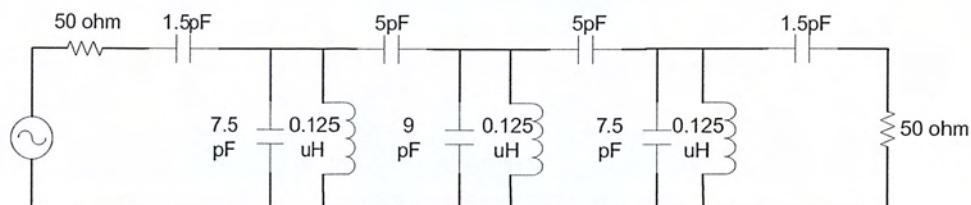


รูปที่ 4.29 ผลตอบสนองทางความถี่ของวงจรกรองความถี่ผ่าน 150 MHz (ปรับปรุง)

4.1.4 วงจรกรองความถี่ผ่าน 400 MHz (ปรับปรุง)

เนื่องจากวงจรกรองความถี่ที่ได้ทำมาก่อนหน้านี้นั้นมีประสิทธิภาพไม่ดีเท่าที่ควร จึงมีการออกแบบวงจรกรองความถี่ใหม่เพื่อพัฒนาวงจรกรองความถี่ให้มีประสิทธิภาพมากขึ้น ซึ่งจะได้วงจรดังรูปที่ 4.30

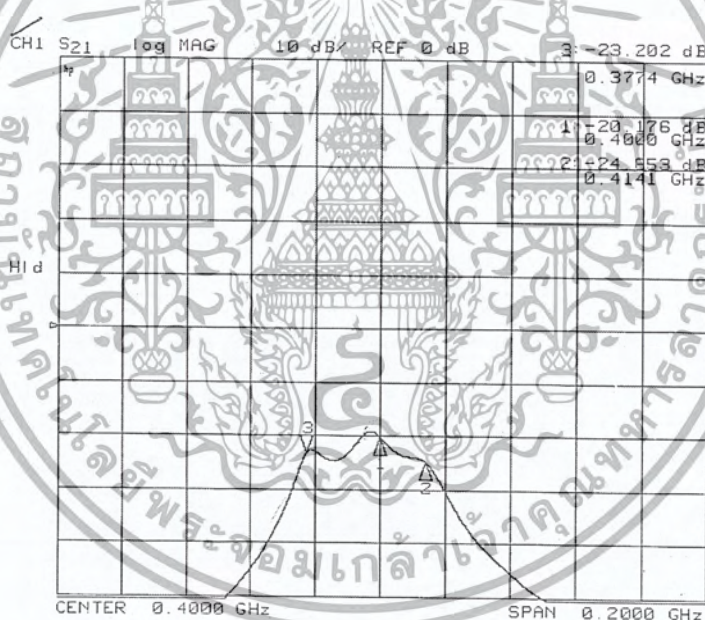
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.30 วงจรกรองความถี่ 400 MHz (ปรับปรุง)

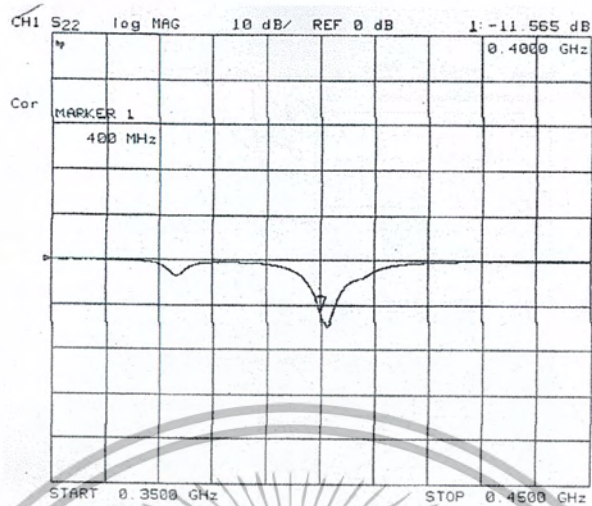
ผลการทดลองวงจรกรองความถี่ผ่าน 400 MHz (ปรับปรุง)

เมื่อนำวงจรที่ได้จากการ Simulation ด้วยโปรแกรม AAE Filter Design มาทำการทดลองจริงแล้วนั้นจะได้ผลการทดลอง ดังนี้

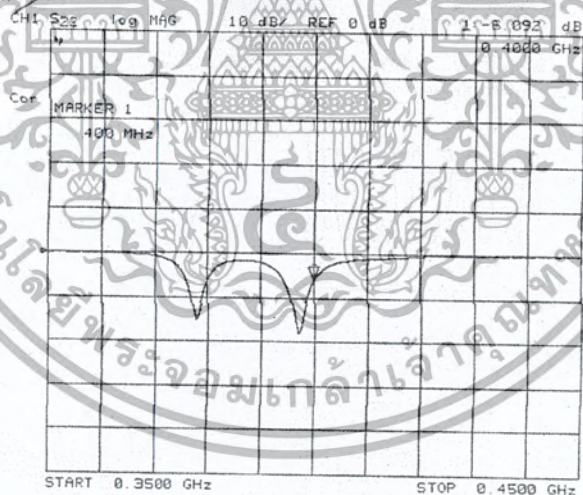


รูปที่ 4.31 ค่าอัตราการขยายในการส่งผ่าน (S_{21}) ของวงจรกรองความถี่ผ่าน 400 MHz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

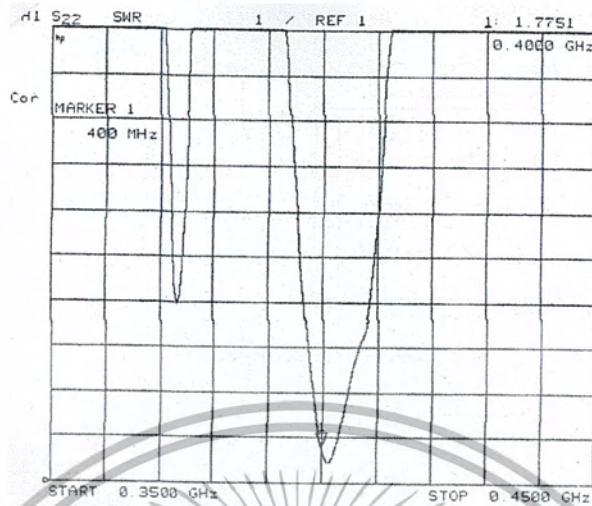


รูปที่ 4.32 ค่าอัตราการสูญเสียเนื่องจากการสะท้อน (Log Mag) ของวงจรรองความถี่ผ่าน 400 MHz ด้านอินพุต



รูปที่ 4.33 ค่าอัตราการสูญเสียเนื่องจากการสะท้อน (Log Mag) ของวงจรรองความถี่ผ่าน 400 MHz ด้านเอาต์พุต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



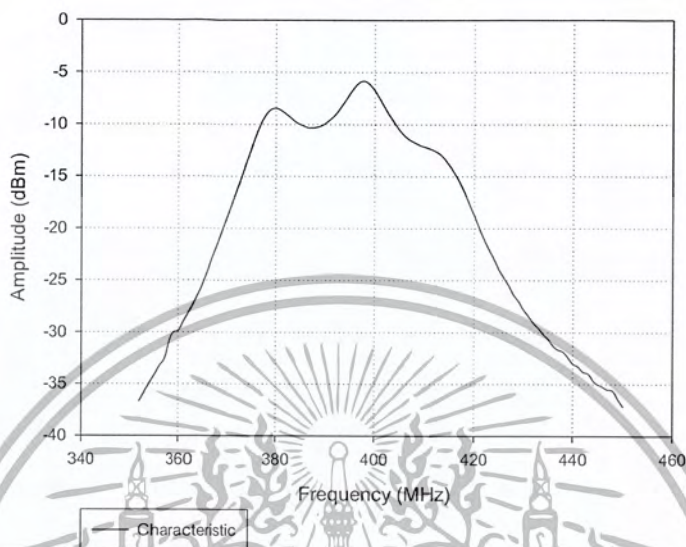
รูปที่ 4.34 ค่า SWR ของวงจรกรองช่วงความถี่ผ่าน 400 MHz ด้านอินพุต



รูปที่ 4.35 ค่า SWR ของวงจรกรองช่วงความถี่ผ่าน 400 MHz ด้านเอาต์พุต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Bandpass Filter 400 MHz



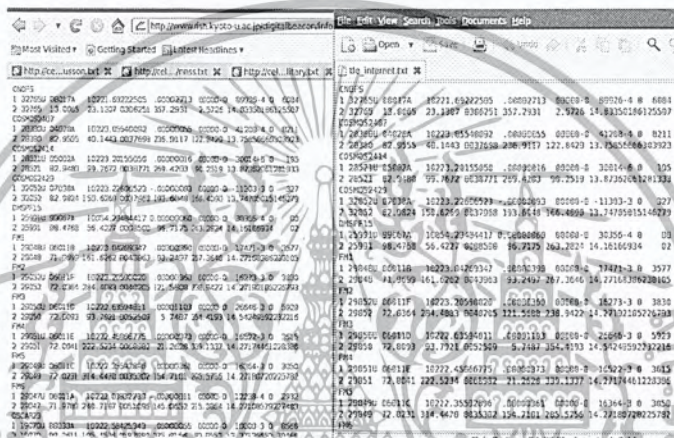
รูปที่ 4.36 ผลตอบสนองทางความถี่ของวงจรกรองความถี่ผ่าน 400 MHz (ปรับปรุง)

4.2 โปรแกรมควบคุมการบันทึกสัญญาณบิลคอน

โปรแกรมนี้ถูกสร้างขึ้นเพื่อใช้ในการบันทึกสัญญาณดาวเทียม โปรแกรมจะเริ่มทำงานในเวลาใดเวลาหนึ่งในทุกๆวัน ในที่นี้เรากำหนดให้โปรแกรมเริ่มทำงานทุกๆเที่ยงคืนของแต่ละวัน โดยโปรแกรมจะเริ่มเก็บค่าวันเวลาปัจจุบันและช่วงเวลาหนึ่งวันในหน่วยนาทีย(1440 s) พร้อมทั้ง ตำแหน่งละติจูด สองจุด และอัลติจูดตามที่กำหนดไว้ จากนั้นทำการอ่านและบันทึกค่า TLE จากเว็บไซต์มาเก็บไว้เป็นเท็กซ์ไฟล์ เพื่อนำไปใช้ในการคำนวณการโคจรผ่านของดาวเทียมต่อไป

โดยทำการทดลองการทำงานของโปรแกรมในส่วนของการอัปเดตค่าข้อมูลจากการเชื่อมต่ออินเทอร์เน็ต เมื่อเครื่องคอมพิวเตอร์มีการเชื่อมต่ออินเทอร์เน็ต โปรแกรมจะแสดง "Internet Connect" และจะทำการเก็บค่าข้อมูลมาเก็บไว้ในไฟล์ชื่อ tle_internet.txt ถ้าไม่สามารถเปิดเว็บไซต์ได้ โปรแกรมจะแสดง "Internet Disconnect" และคำนวณการโคจรผ่านของสัญญาณดาวเทียมด้วยค่า TLE ชุดเดิม

ผลการทดลอง พบว่าเมื่อทำการเชื่อมต่ออินเทอร์เน็ตเน็ตเข้ากับเครื่องคอมพิวเตอร์ โปรแกรมสามารถทำการเปิดเว็บไซต์ <http://www.rish.kyotou.ac.jp/digitalbeacon/information/cosmic.tle> และทำการอ่านข้อมูลจากเว็บไซต์มาเก็บเป็นเท็กซ์ไฟล์ได้พร้อมทั้งแสดง “Internet Connect” และทำการคำนวณการโคจรผ่านของดาวเทียมด้วยค่า TLE ใหม่ที่บันทึกได้ ในกรณีที่ขาดการเชื่อมต่ออินเทอร์เน็ตพบว่าโปรแกรมทำการแสดงค่า “Internet Disconnect” ดังรูปที่ 4.38(ข) และสามารถคำนวณการโคจรผ่านของสัญญาณดาวเทียมด้วยค่า TLE ชุดเดิม



รูปที่ 4.37 ค่า TLE ที่ได้จากเว็บไซต์

```

root@pae-desktop: /home/pae/Desktop/tle2azel-1.13b
File Edit View Terminal Help

root@pae-desktop: /home/pae/Desktop/tle2azel-1.13b# /usr/bin/python /home/pae/Desktop/tle2azel-1.13b/ordertime.py
usrp_cdm_cron.py
Internet connect
{'new Satellite': 'CUIOFS'}
{'new Satellite': 'COSMOS2407'}
{'new Satellite': 'COSMOS2414'}
{'new Satellite': 'COSMOS2429'}
{'new Satellite': 'COSMOS2454'}
{'new Satellite': 'COSMOS2461'}

```

(ก)

```

root@pae-desktop: /home/pae/Desktop/tle2azel-1.13b
File Edit View Terminal Help

root@pae-desktop: /home/pae/Desktop/tle2azel-1.13b# /usr/bin/python /home/pae/Desktop/tle2azel-1.13b/ordertime.py
usrp_cdm_cron.py
Internet Disconnect
{'new Satellite': 'CUIOFS'}
{'new Satellite': 'COSMOS2407'}
{'new Satellite': 'COSMOS2414'}
{'new Satellite': 'COSMOS2429'}

```

(ข)

รูปที่ 4.38 (ก) คอมพิวเตอร์เชื่อมต่ออินเทอร์เน็ต (ข) คอมพิวเตอร์ไม่เชื่อมต่ออินเทอร์เน็ต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากนั้น โปรแกรมจะทำการคำนวณการโคจรผ่านของดาวเทียม และเก็บเป็นค่าเท็กซ์ไฟล์ แสดงดังรูปที่ 4.39

```

-----
Lat,Long,Alt : 13:46:48.0, 106:46:48.0, 0.000 meters
Pressure,Temp: 0mB,0C
Start Time : 2010/9/20 20:30:10 UTC
Stop Time : 2010/9/21 21:30:10 UTC
Time step : 10.0 second(s)
-----
Files
-----
Input File : /home/pae/Desktop/tlc2azel-1.13b/catalog/tlc_internet.txt
Output File : stdout
Output Format: "%s %s %.4f %.4f " % (object_name,date,az,el)
-----
Visibility Area
-----
Azimuth (dd:mm:ss) : 0:00:00.0
Elevation (dd:mm:ss) : 90:00:00.0
Angle (dd:mm:ss) : 60:00:00.0
-----

```

NAME	DURAT (min)	ENTER (UTC)	EXIT (UTC)	AZ (deg)	EL (deg)	RA (hr)	DEC (deg)	dAZ (deg/min)	dEL (deg/min)
FM1	5.5	04:58:00	04:58:25	239.34	09.83	15.69	7.951	-7.5	13.2
FM2	4.9	03:25:00	03:25:30	101.68	83.35	15.92	7.201	11.7	13.8
FM3	1.8	04:58:38	04:58:48	73.67	05.73	13.51	19.28	11.6	14.9

```

-----
บรรทัดที่ 04:/home/pae/Desktop/tlc2azel-1.13b/plotmode/visibility... Plain Text 7 tab width: 8
-----

```

รูปที่ 4.39 แสดงข้อมูลการโคจรผ่านของดาวเทียม

จากนั้น โปรแกรมจะทำการเรียงลำดับค่าเวลาในการเข้าออกของดาวเทียมที่ได้จากการคำนวณ ให้เรียงค่าเวลาก่อนไปมาก โดยอาศัยการเปรียบเทียบเวลาในโมดูล datetime.time()

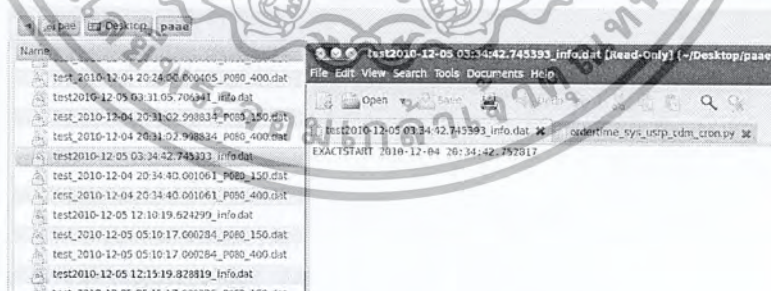
ผลการทดลอง พบว่าโปรแกรมสามารถเรียงลำดับค่าของเวลาได้โดยสามารถเรียงค่าเวลาจากมากไปน้อย ดังรูปที่ 4.40 คอลัมน์ด้านซ้ายเป็นค่าเวลาที่ได้จากการคำนวณ คอลัมน์ด้านขวาเป็นค่าเวลาที่ผ่านการเรียงลำดับแล้ว จะเห็นได้ว่าโปรแกรมสามารถเรียงลำดับค่าเวลาจากน้อยไปมากได้

1	Enter time	Exit time	Enter time	Exit time
2	datetime.time(1, 24, 14),	[datetime.time(1, 29, 7),	[datetime.time(1, 24, 11),	[datetime.time(1, 30, 32),
3	datetime.time(3, 7, 50),	datetime.time(3, 13, 31),	datetime.time(1, 24, 14),	datetime.time(1, 29, 7),
4	datetime.time(4, 52, 3),	datetime.time(4, 57, 18),	datetime.time(2, 15, 37),	datetime.time(2, 19, 11),
5	datetime.time(6, 36, 24),	datetime.time(6, 40, 46),	datetime.time(3, 7, 50),	datetime.time(3, 13, 31),
6	datetime.time(6, 21, 11),	datetime.time(6, 23, 14),	datetime.time(3, 58, 57),	datetime.time(4, 2, 50),
7	datetime.time(5, 57, 15),	datetime.time(6, 3, 23),	datetime.time(4, 52, 3),	datetime.time(4, 57, 18),
8	datetime.time(17, 26, 41),	datetime.time(17, 31, 32),	datetime.time(5, 7, 1),	datetime.time(5, 11, 40),
9	datetime.time(9, 55, 43),	datetime.time(10, 1, 12),	datetime.time(5, 28, 16),	datetime.time(5, 33, 36),
10	datetime.time(21, 19, 14),	datetime.time(21, 23, 42),	datetime.time(5, 57, 15),	datetime.time(6, 3, 23),
11	datetime.time(1, 24, 11),	datetime.time(1, 30, 32),	datetime.time(6, 33, 52),	datetime.time(6, 41, 46),
12	datetime.time(12, 37, 33),	datetime.time(12, 42, 40),	datetime.time(6, 36, 24),	datetime.time(6, 40, 46),
13	datetime.time(23, 6, 16),	datetime.time(23, 8, 32),	datetime.time(6, 40, 37),	datetime.time(6, 48, 18),
14	datetime.time(10, 14, 26),	datetime.time(10, 19, 48),	datetime.time(6, 40, 58),	datetime.time(6, 47, 19),
15	datetime.time(10, 15, 5),	datetime.time(10, 20, 37),	datetime.time(6, 43, 17),	datetime.time(6, 48, 13),
16	datetime.time(6, 43, 17),	datetime.time(6, 48, 13),	datetime.time(6, 53, 46),	datetime.time(7, 0, 53),
17	datetime.time(5, 7, 1),	datetime.time(5, 11, 40),	datetime.time(8, 13),	datetime.time(8, 15, 8),
18	datetime.time(17, 37, 44),	datetime.time(17, 42, 19),	datetime.time(8, 21, 11),	datetime.time(8, 23, 14),
19	datetime.time(2, 15, 37),	datetime.time(2, 19, 11),	datetime.time(9, 55, 43),	datetime.time(10, 1, 12),
20	datetime.time(14, 47, 32),	datetime.time(14, 47, 54),	datetime.time(10, 14, 26),	datetime.time(10, 19, 48),
21	datetime.time(8, 13),	datetime.time(8, 15, 8),	datetime.time(10, 15, 5),	datetime.time(10, 20, 37),
22	datetime.time(20, 42, 30),	datetime.time(20, 45, 13),	datetime.time(12, 37, 33),	datetime.time(12, 42, 40),
23	datetime.time(3, 58, 57),	datetime.time(4, 2, 50),	datetime.time(14, 47, 32),	datetime.time(14, 47, 54),
24	datetime.time(5, 48, 37),	datetime.time(6, 48, 15),	datetime.time(17, 26, 41),	datetime.time(17, 31, 32),
25	datetime.time(18, 18, 11),	datetime.time(18, 24, 51),	datetime.time(17, 37, 44),	datetime.time(17, 42, 19),
26	datetime.time(6, 40, 58),	datetime.time(6, 47, 19),	datetime.time(18, 11, 19),	datetime.time(18, 17, 54),
27	datetime.time(18, 27, 28),	datetime.time(18, 32, 30),	datetime.time(18, 18, 11),	datetime.time(18, 24, 51),
28	datetime.time(6, 53, 46),	datetime.time(7, 0, 53),	datetime.time(18, 27, 28),	datetime.time(18, 32, 30),
29	datetime.time(18, 39, 49),	datetime.time(18, 46, 19),	datetime.time(18, 39, 49),	datetime.time(18, 46, 19),
30	datetime.time(6, 33, 52),	datetime.time(6, 41, 46),	datetime.time(20, 42, 30),	datetime.time(20, 45, 13),
31	datetime.time(18, 11, 19),	datetime.time(18, 17, 54),	datetime.time(21, 19, 14),	datetime.time(21, 23, 42),
32	datetime.time(5, 28, 16),	datetime.time(5, 33, 36),	datetime.time(23, 6, 16),	datetime.time(23, 6, 32),

รูปที่ 4.40 ค่าเวลาที่ได้จากโปรแกรมก่อนและหลังเรียงลำดับ

จากนั้น โปรแกรมจะทำการคอยตรวจสอบช่วงเวลาปัจจุบันว่ามีการโคจรผ่านมาของดาวเทียมหรือไม่ โดยในกรณีที่ไม่มีการโคจรผ่านของดาวเทียม โปรแกรมก็จะทำการเช็คและรอคอยการมาถึงของดาวเทียมในรอบถัดไป ส่วนในกรณีที่มีการโคจรผ่านของดาวเทียมในช่วงเวลาปัจจุบัน โปรแกรมจะ ไปสั่งงานให้ฟังก์ชัน `save_data(enter,exit,saname)` ทำงาน

ผลการทดลองพบว่าฟังก์ชัน `save_data(enter,exit,saname)` จะสั่งงานให้คลาส `wfm_rx_block()` ทำงานและสร้างเท็กซ์ไฟล์ขึ้นมาเพื่อนันทักค่าเวลาที่เริ่มเก็บสัญญาณจริง ดังรูปที่ 4.41



รูปที่ 4.41 แสดงค่าเวลาที่เริ่มทำการบันทึกสัญญาณจริง

เมื่อคลาส `wfm_rx_block()` เริ่มทำงาน จะแสดงผลของสถานะของอุปกรณ์ในการบันทึกสัญญาณ หรือ USRP ขึ้นดังรูปที่ 4.42

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

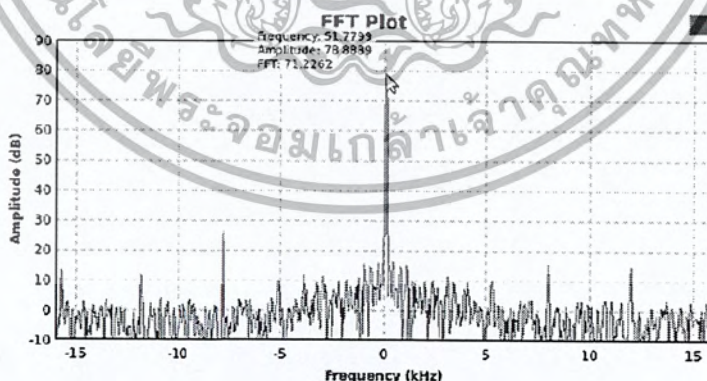
```

True
True
Welcome GNUFS
Obs start: 2011-01-08 20:03:12 . end: 2011-01-08 20:08:31
/home/long/Desktop/paperfile/2011-1-13b/result/test GNUFS 2011-01-08 20:08:31.400693 P000 400.dat
frequency: nfreq= 1 . foffset= [80]
Using TX d'board A: WBX NO RX
Using TX d'board D: Basic RX
LPF coef Len= 90
*** WBX side ***
Target Freq = 400340000.0
r.baseband freq = 400M
r.dsc freq = 0
r.residual freq = 0
r.inverted = false
OK
*** Basic RX side ***
Target Freq = 150000000.0
r.baseband freq = 0
r.dsc freq = 0.22M
r.residual freq = 0
r.inverted = false
OK
True
True
Welcome FM2
Obs start: 2011-01-08 20:34:26 . end: 2011-01-08 20:39:10
/home/long/Desktop/paperfile/2011-1-13b/result/test FM2 2011-01-08 20:39:10.364983 P000 400.dat
frequency: nfreq= 1 . foffset= [80]
Using TX d'board A: WBX NO RX

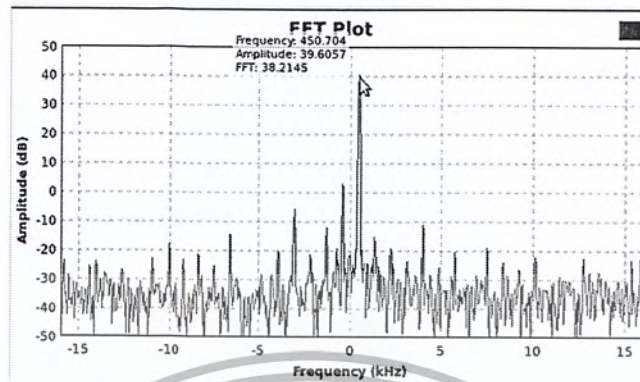
```

รูปที่ 4.42 สถานะเครื่องบันทึกสัญญาณ

จากนั้นทดลองนำค่าสัญญาณที่บันทึกได้มาแสดงผลผ่าน FFT Block โดยอาศัยกราฟิกเคิลยูเซอร์อินเตอร์เฟซซอฟต์แวร์ที่เรียกว่า GRC ในที่นี้ เราทำการป้อนสัญญาณอินพุต 400 MHz และ 150 MHz จากเครื่องกำเนิดสัญญาณความถี่สูง ให้กับ USRP จะ ได้ผลการทดลองดังรูปที่ 4.43 และ 4.44



รูปที่ 4.43 สัญญาณที่บันทึกได้เมื่อทำการป้อนสัญญาณความถี่ 150 MHz



รูปที่ 4.44 สัญญาณที่บันทึกได้เมื่อทำการป้อนสัญญาณความถี่ 400 MHz

4.3 อินเทอร์เฟซโปรแกรมการคำนวณการโคจรผ่านของดาวเทียมในการบันทึกสัญญาณบีดคอน

ทำการทดลองสร้างกราฟไฟคยูเซอร์อินเทอร์เฟซ (GUI) ให้ผู้ใช้งานใช้งานโปรแกรมคำนวณการโคจรผ่านของดาวเทียมได้สะดวกขึ้น โดยโปรแกรมจะทำการสร้างอินเทอร์เฟซให้ผู้ใช้ออกข้อมูลเกี่ยวกับบริเวณที่สนใจ ช่วงเวลาที่ทำการคำนวณ รวมถึงช่วงเวลาที่ต้องการคำนวณ

ผลการทดลอง พบว่าเมื่อโปรแกรมเริ่มทำงานจะปรากฏหน้าอินเทอร์เฟซของโปรแกรมขึ้นดังรูปที่ 4.45 ให้ผู้ใช้งานสามารถกรอกข้อมูลให้โปรแกรมนำไปใช้คำนวณต่อไป

Satellite prediction

Location
Latitude: Degree
Longitude: Degree
Altitude: Meter

Time
Date: Year: Month: Day:
Start time: 24 Hour: Minute: Second:
Duration: Minute

รูปที่ 4.45 อินเทอร์เฟซโปรแกรมคำนวณการโคจรผ่านของดาวเทียม

ในที่นี้ผู้ใช้ทำการป้อน Location เป็นละติจูด ลองจิจูด และ ค่าความสูงจากน้ำทะเล (Altitude) บริเวณที่เราสนใจหรือบริเวณเครื่องรับสัญญาณดาวเทียม และ ใส่ค่าวันและเวลาที่เรารเริ่มทำการสังเกต พร้อมทั้งช่วงเวลาที่ทำการสังเกตต่อไปหน่วยเป็นนาที โดยในที่นี้เราจะใส่ค่าละติจูด 13.7438 ลองจิจูด 100.78 ความสูงจากระดับน้ำทะเล 0.00 วันเวลาที่ทำการสังเกต 2010/9/1 14:30:00 มีช่วงเวลาที่ทำการสังเกต 25 ชั่วโมง ($60 \times 25 = 1500$) ดังรูปที่ 4.46

The screenshot shows a window titled "gui_run.py" with a "Sattelite prediction" interface. It includes input fields for Location (Latitude, Longitude, Altitude) and Time (Date, Start time, Duration). A "Predict" button is located at the bottom of the form.

รูปที่ 4.46 กำหนดค่าให้กับโปรแกรมคำนวณการโคจรผ่านของดาวเทียม

จุดนั้นทำการทดสอบการกรอกข้อมูลของผู้ใช้ในกรณีที่ผู้ใช้ใส่ข้อมูลผิดประเภท เช่นตำแหน่งควรจะได้เป็นตัวเลข ในกรณีที่ใส่เป็นตัวอักษร จะมีกล่องข้อความเตือนถึงการกรอกข้อมูลผิดประเภทของผู้ใช้

ผลการทดลอง ในกรณีที่เรใส่ชนิดของค่าตำแหน่งและช่วงเวลาผิดให้กับ GUI โปรแกรมจะทำการเตือนด้วยการแสดงกล่องข้อความ (Alert Box) พร้อมระบุตำแหน่งที่ป้อนชนิดของข้อมูลผิดลงไป เพื่อให้ผู้ใช้นั้นทำการใส่ค่าใหม่อีกครั้ง ดังรูปที่ 4.47 จะเห็นว่ามีกรป้อนชนิดของตำแหน่งผิด กล่องข้อความเตือนนั้นจะระบุว่าตำแหน่งใดผิด ในที่นี้มีการใส่ค่าละติจูดและความสูงผิด

รูปที่ 4.47 กล้องข้อความเตือนการกรอกข้อมูลเกินขีด

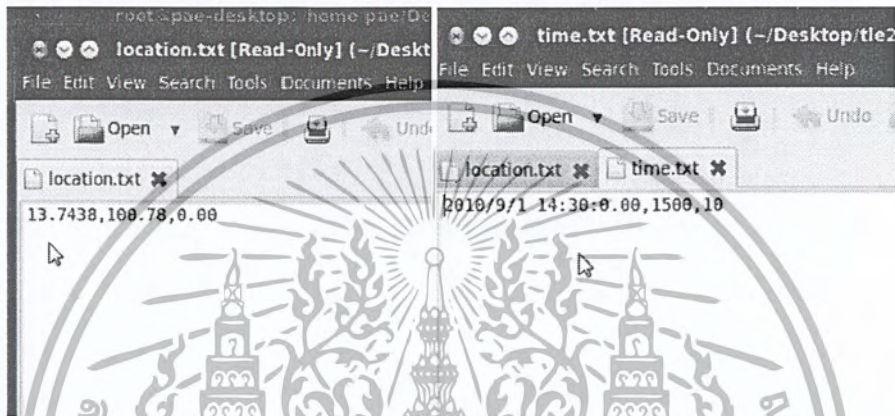
รวมถึงโปรแกรมนี้มีการเช็คไปถึงค่าของวันเวลาทั่วไปเช่นในกรณีที่ผู้ใช้ ป้อนเดือนเกินจำนวนจริงนั้นหมายถึงมีค่ามากกว่า 12 ในทำนองเดียวกัน ชั่วโมงต้องน้อยกว่า 24 และ นาทีต้องน้อยกว่า 60

เมื่อทำการทดลองใส่ค่าเกินช่วงที่กำหนดไว้พบว่ามีกล้องข้อความแสดงข้อความระบุถึงตำแหน่งที่ผู้ใช้กรอกข้อมูลผิดลงไป

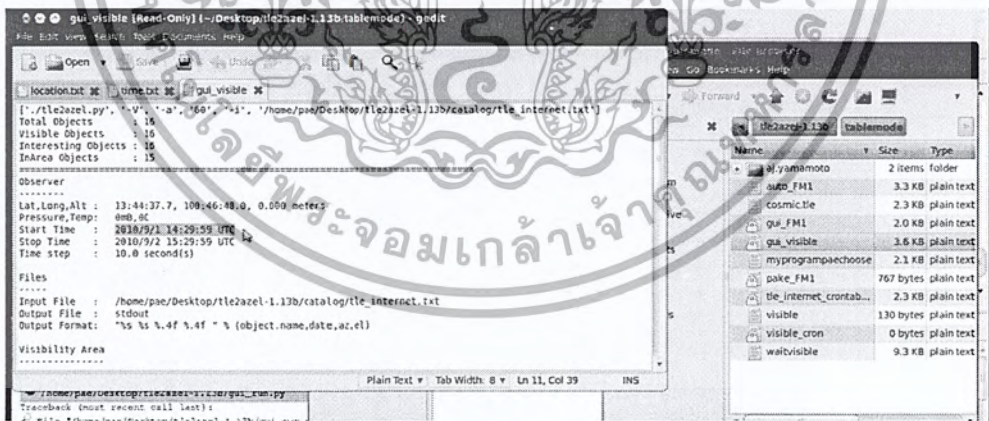
รูปที่ 4.48 กล้องข้อความเตือนการใส่ข้อมูลวัน เวลา ที่ไม่เป็นจริง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากนั้นทดลองทำการกรอกข้อมูลให้ถูกต้องครบถ้วนพบว่าโปรแกรมจะทำการเก็บไฟล์ location และ time เพื่อนำไปใช้เป็นอินพุตให้กับโปรแกรม tle2azel ต่อไป ดังรูปที่ 4.49(ก) และ รูปที่ 4.49(ข) จากนั้นทำการคำนวณการโคจรผ่านของดาวเทียมโดยเก็บผลลัพธ์ที่ได้เป็นเท็กซ์ไฟล์ ดังรูปที่ 4.50



รูปที่ 4.49 (ก)เพิ่มข้อมูลตำแหน่งที่เราสนใจ (ข) เพิ่มวัน เวลา ช่วงเวลาที่ทำการสังเกต



รูปที่ 4.50 ไฟล์ที่ทำการคำนวณการโคจรผ่านของดาวเทียม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

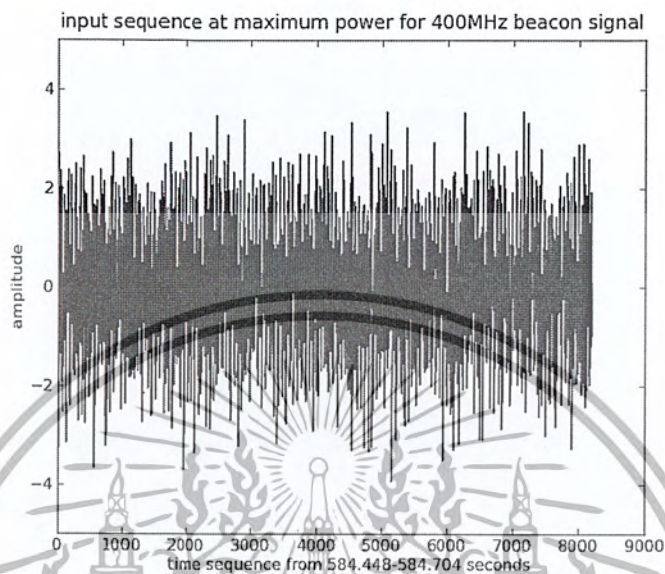
ผลการคำนวณการโคจรผ่านของดาวเทียมแสดงดังรูปที่ 4.49 พบว่าการมีคำนวณการโคจรผ่านในตำแหน่งบริเวณที่เราสนใจ ช่วงเวลาที่ทำการคำนวณ ตามที่ผู้ใช้ทำการกรอกลงในกราฟฟิคคอลลูเซชันอินเตอร์เฟส

จากการทดลองพบว่าการคำนวณการโคจรผ่านของดาวเทียม โปรแกรมสามารถสร้างอินเตอร์เฟสมารับข้อมูลแล้วนำข้อมูลที่ได้เป็นอินพุตให้โปรแกรมทำการคำนวณการโคจรผ่านของดาวเทียม พบว่าไฟล์ที่ได้นั้นได้รับค่าอินพุตตรงกันกับที่ป้อนให้กับ GUI จากนั้นได้มีการพัฒนาการใช้กราฟฟิคคอลลูเซชันอินเตอร์เฟสรวมกับการบันทึกสัญญาณจากดาวเทียมซึ่งพบว่าสามารถบันทึกค่าสัญญาณดาวเทียมได้

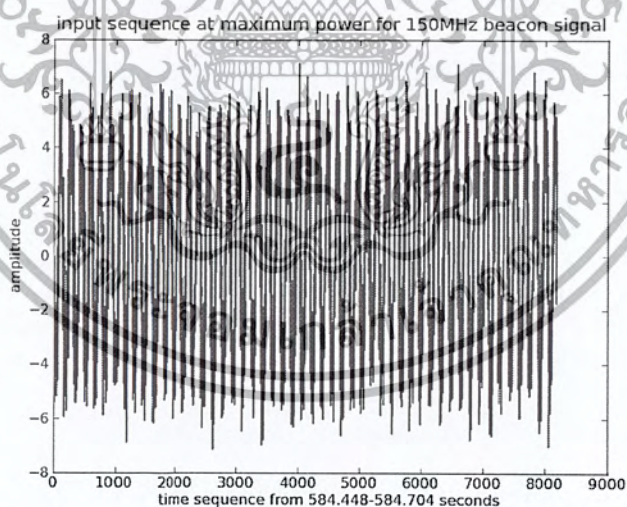
4.4 ภาคการประมวลผลข้อมูลสัญญาณบีคอน

ภาคการประมวลผลข้อมูลนั้นจะทำโดยใช้คอมพิวเตอร์ และ โปรแกรมภาษาไพธอน เป็นหลักในการทำงานทำการอ่านไฟล์ข้อมูลของสัญญาณบีคอน ทั้งสองไฟล์ โดยยกตัวอย่างการคำนวณค่าเฟสจากดาวเทียม COSMOS2414 ซึ่งแบ่งออกเป็น 2 ไฟล์ คือ ไฟล์ของสัญญาณบีคอน 400MHz และ 150 MHz จากนั้น ตามกระบวนการดึงไฟล์วาร์ทรูปที่ 3.12 เมื่อเปิดไฟล์ทั้งสองไฟล์ขึ้นมาแล้ว ทำการอ่านค่าภายในไฟล์ จะพบว่าไฟล์ ทั้งสองมีขนาดเท่ากันคือ 252936800 bytes (ประมาณ 241MB) จากนั้น ทำการอ่านข้อมูลเป็นตัวเลขเชิงซ้อน จะได้ค่าเลขเชิงซ้อนทั้งหมด 31612928 ค่า จากนั้นทำการเลือกบางช่วงของไฟล์ทั้งสอง นำมาพล็อตกราฟแสดงค่าสัญญาณในแกนเวลาจะพบว่าสัญญาณที่นำมาพล็อตกราฟนั้นเป็นสัญญาณในแกนเวลาที่มีสัญญาณรบกวนอยู่มาก ดังรูป ที่ 4.51 และ 4.52

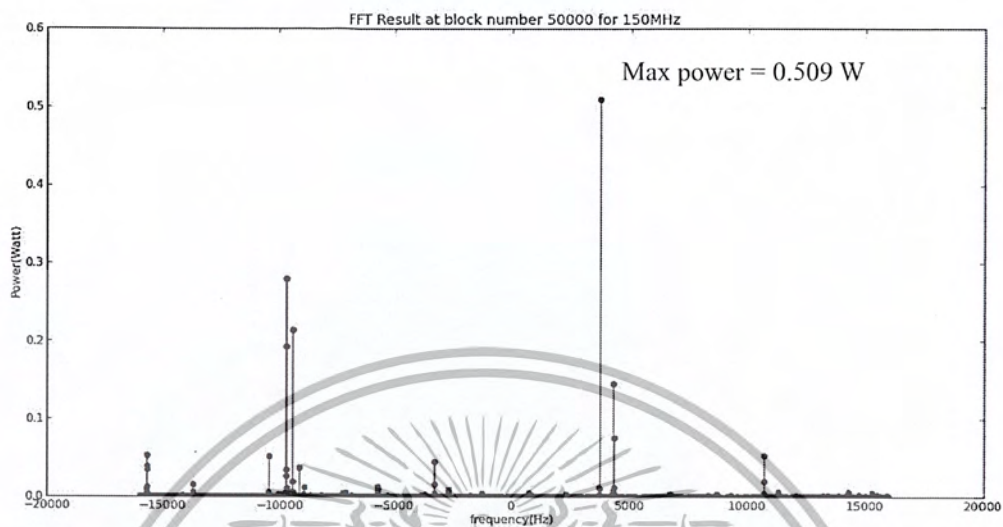
หลังจากอ่านไฟล์ทั้งสองแล้วจะทำการแบ่งไฟล์มาทำการแปลง FFT โดยแปลงข้อมูลทีละ 4096ค่า และทำการเลื่อนตำแหน่งข้อมูลไปที่ละ 128ค่า ซึ่งค่าเฟสที่ได้จะตอบสนองต่อการเปลี่ยนแปลงที่ 250Hz โดยทำการแปลง FFT ทั้งสิ้น $\frac{31612928}{4096} \times 32 = 246976$ ครั้ง โดยเมื่อแปลงแล้วจะผลการแปลง FFT เป็นค่าเลขเชิงซ้อนแสดงได้ตามสมการ 3.62 ซึ่งสามารถหาค่าขนาด ค่ากำลัง และค่าเฟสได้ตามสมการ 3.63, 3.64 และ 3.65 โดยสเปกตรัมแต่ละความถี่จะห่างกัน 7.8125 Hz โดยเมื่อได้ผลของการแปลง FFT แล้วจะนำมาพลอตสเปกตรัมของสัญญาณเพื่อหาค่าสเปกตรัมที่มีค่ากำลังสูงสุดในการแปลง FFT แต่ละครั้งซึ่งจะได้ค่าเฟสทั้งสิ้นจำนวน 2469768 ค่า โดยสามารถแสดงผลของการแปลง FFT มาพลอตขนาด และเฟสของสัญญาณได้ดังรูปที่ 4.53 - 4.64



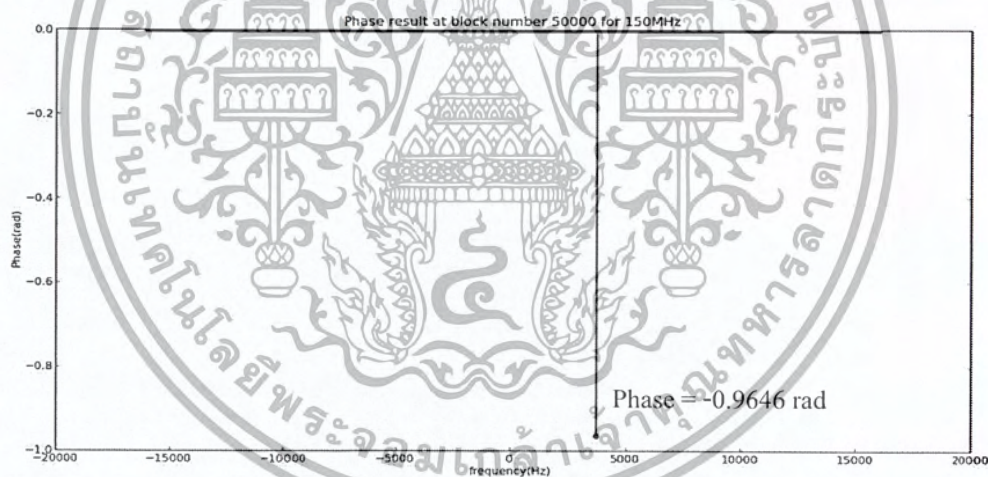
รูปที่ 4.51 รูปสัญญาณที่พล็อตจากไฟล์ข้อมูลบางส่วนของสัญญาณบีคอนความถี่ 400 MHz



รูปที่ 4.52 แสดงรูปสัญญาณที่พล็อตจากไฟล์ข้อมูลบางส่วนของสัญญาณบีคอนความถี่ 150 MHz

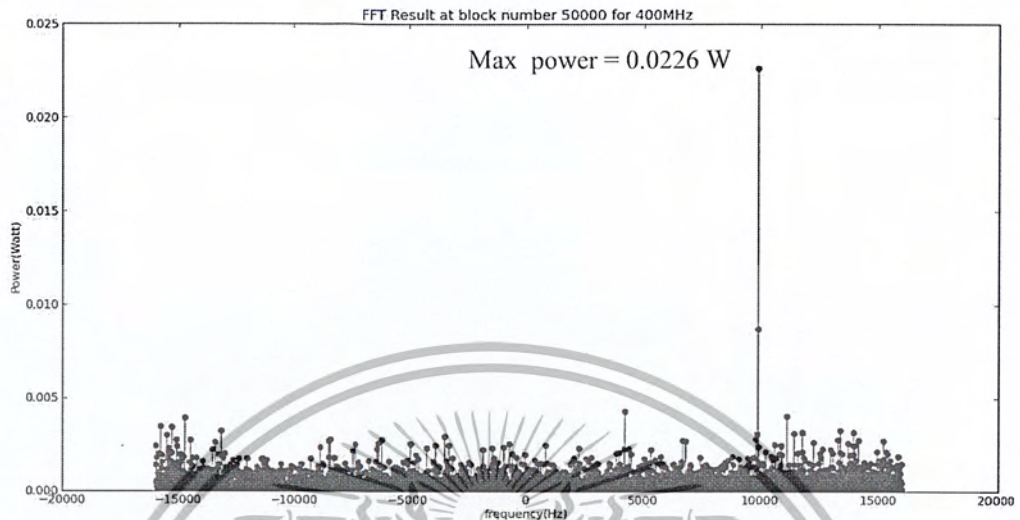


รูปที่ 4.53 สเปกตรัมกำลังสัญญาณปีคอนที่การแปลง FFT ครั้งที่ 50000 ความถี่ 150MHz

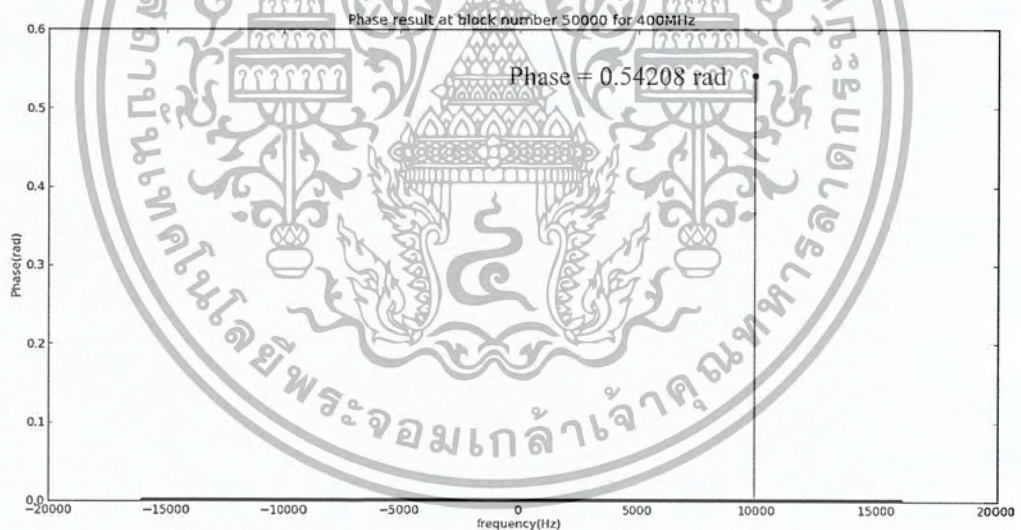


รูปที่ 4.54 สเปกตรัมทางเฟสของสัญญาณปีคอนที่การแปลง FFT ครั้งที่ 50000 ความถี่ 150MHz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

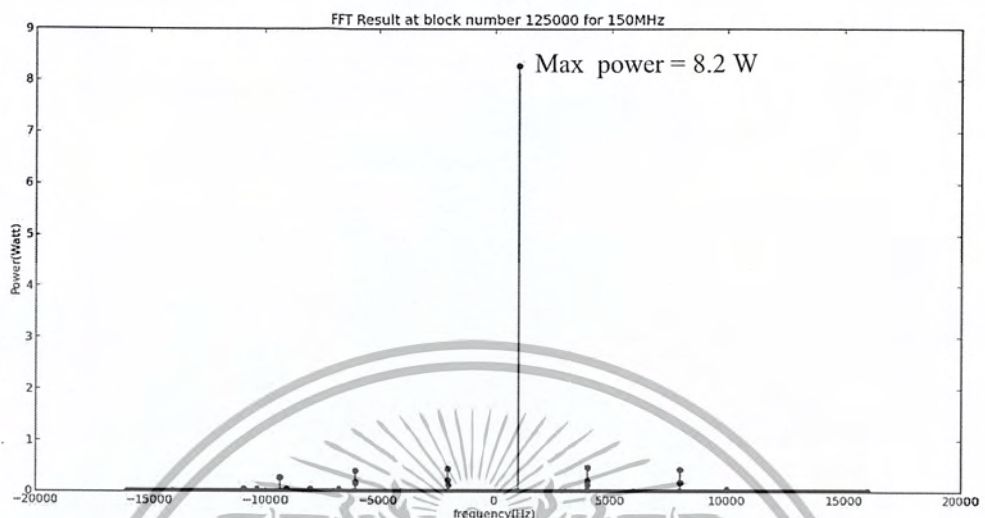


รูปที่ 4.55 สเปกตรัมทางเฟสของสัญญาณบีคอนที่การแปลง FFT ครั้งที่ 50000 ความถี่ 400MHz



รูปที่ 4.56 สเปกตรัมทางเฟสของสัญญาณบีคอนที่การแปลง FFT ครั้งที่ 50000 ความถี่ 400MHz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

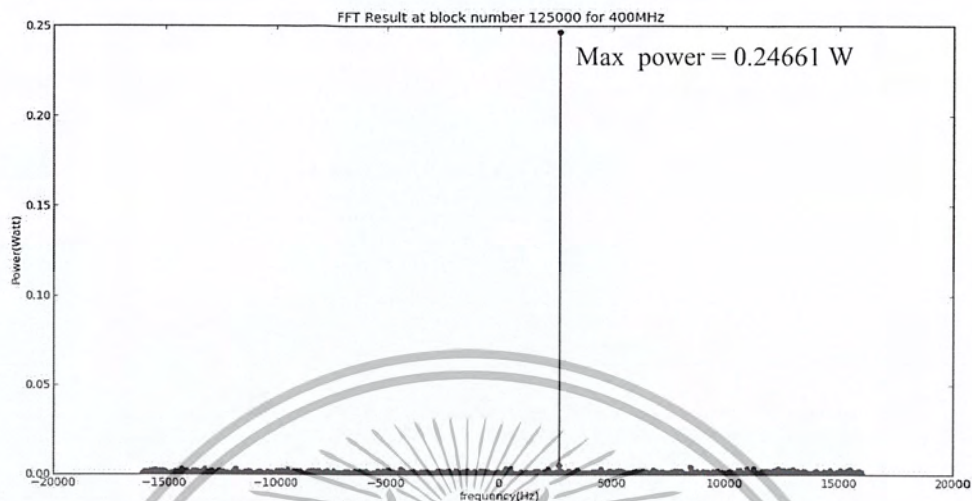


รูปที่ 4.57 สเปกตรัมกำลังสัญญาณบิตคอนที่การแปลง FFT ครั้งที่ 125000 ความถี่ 150MHz



รูปที่ 4.58 สเปกตรัมทางเฟสของสัญญาณบิตคอนที่การแปลง FFT ครั้งที่ 125000 ความถี่ 150MHz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

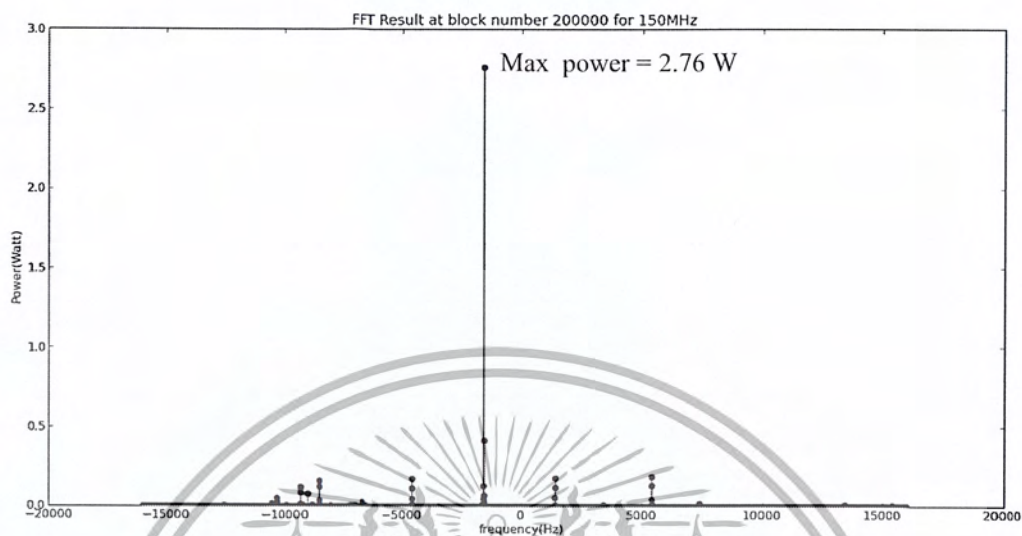


รูปที่ 4.59 สเปกตรัมกำลังสัญญาณบีคอนที่การแปลง FFT ครั้งที่ 125000 ความถี่ 400MHz



รูปที่ 4.60 สเปกตรัมทางเฟสของสัญญาณบีคอนที่การแปลง FFT ครั้งที่ 125000 ความถี่ 400MHz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

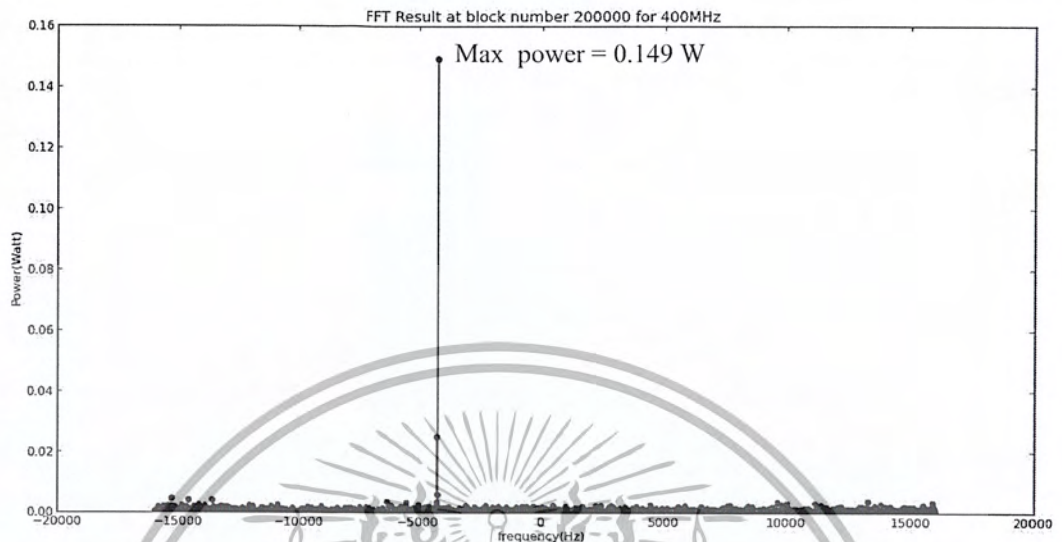


รูปที่ 4.61 สเปกตรัมกำลังสัญญาณบีคอนที่การแปลง FFT ครั้งที่ 200000 ความถี่ 150MHz



รูปที่ 4.62 สเปกตรัมทางเฟสของสัญญาณบีคอนที่การแปลง FFT ครั้งที่ 200000 ความถี่ 150MHz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.63 สเปกตรัมกำลังสัญญาณบีคอนที่การแปลง FFT ครั้งที่ 200000 ความถี่ 400MHz

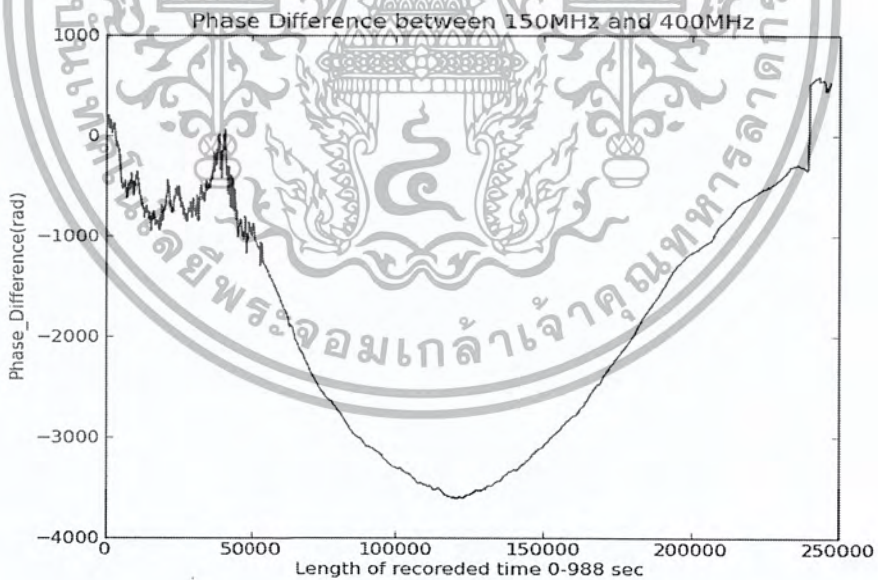


รูปที่ 4.64 สเปกตรัมทางเฟสของสัญญาณบีคอนที่การแปลง FFT ครั้งที่ 200000 ความถี่ 400MHz

โดยจากกราฟพบว่าในแต่ละช่วงการแปลง FFT ที่เวลาต่างกันจะพบว่าสเปกตรัมทางกำลังของสัญญาณบีคอนความถี่ 150MHz และ 400 MHz นั้นมีค่าสูงสุด ณ ตำแหน่งความถี่ที่

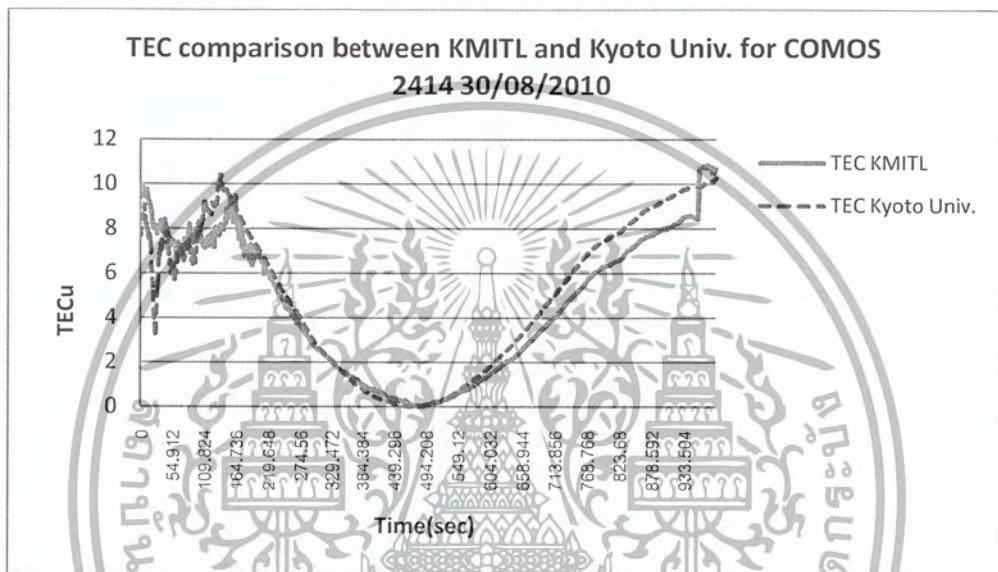
ต่างกัน โดยจะเห็นว่าในช่วงแรกของการแปลง FFT สเปกตรัมกำลังของสัญญาณบีกอนทั้งสองตอบสนองที่ความถี่ด้านบวกที่ประมาณ 10kHz หลังจากนั้นในช่วงกลางของการแปลง FFT จะเห็นว่าสัญญาณตอบสนองที่ความถี่ใกล้กับความถี่ 0 Hz ทางด้านลบ จากนั้นในช่วงสุดท้าย สเปกตรัมของสัญญาณจะไปตอบสนองที่ความถี่ใกล้กับ 0Hz ทางด้านลบ ซึ่งสาเหตุที่สเปกตรัมของสัญญาณบีกอนนั้นมีการตอบสนองที่ความถี่ต่างกัน ณ ช่วงเวลาของการแปลง FFT ที่ต่างกัน นั้นเป็นเพราะผลของการเกิดความถี่ดอปเปลอร์ (Doppler's Frequency) ซึ่งกล่าวได้ว่าถ้าผู้สังเกตอยู่นิ่งในขณะที่แหล่งกำเนิดเคลื่อนเข้าหาผู้สังเกต ความถี่ของสัญญาณที่ผู้รับรับได้นั้นจะสูงขึ้นกว่าสัญญาณจริง และ เช่นเดียวกันถ้าแหล่งกำเนิดเคลื่อนออกจากผู้สังเกต ความถี่ของสัญญาณที่ผู้รับรับได้นั้นจะความถี่จะน้อยกว่าความถี่สัญญาณจริงซึ่งสอดคล้องกับกราฟที่ได้แสดงไปข้างต้น

หลังจากได้ค่าเฟสของสัญญาณบีกอนทั้งสองความถี่มาแล้ว ก็จะเข้าสู่การคำนวณความต่างเฟส โดยทำการคูณค่าเฟสที่ได้จากสัญญาณบีกอนความถี่ 150MHz และ 400MHz ด้วยอัตราส่วน 8 และ 3 ตามลำดับเพื่อทำให้ความถี่สองค่าเท่ากันที่ 1200MHz หลังจากนั้นนำค่าเฟสมาลบกันจะได้ค่าความต่างเฟสที่เปลี่ยนแปลงตามเวลาแสดงดังรูปที่ 4.65



รูปที่ 4.65 ค่าผลต่างเฟสของสัญญาณบีกอน 150MHz และ 400MHz เปลี่ยนแปลงตามเวลา

โดยค่าเฟสที่ได้นั้นสามารถนำไปหาค่า TEC ได้ โดยจากสมการที่ 3.66 ซึ่งสามารถหาค่า TEC เปรียบเทียบกับค่า TEC ที่ของสัญญาณบีคอนเดียวกันแต่ใช้การคำนวณของ Kyoto University ซึ่งการคำนวณค่าเฟสนั้นได้ผ่านการทดสอบเปรียบเทียบกับเครื่องรับสัญญาณดาวเทียมแอนะล็อกแล้วเป็นค่าที่ยอมรับได้โดยสามารถแสดงได้ดังรูปที่ 4.66



รูปที่ 4.66 กราฟเปรียบเทียบค่า TEC ที่คำนวณได้ระหว่าง KMITL และ Kyoto University

จากกราฟแสดงการเปรียบเทียบจะพบว่าค่า TEC ที่ได้จากการคำนวณเมื่อทำการเปรียบเทียบกับค่ามาตรฐานแล้วพบว่าในช่วงแรกที่ดาวเทียมโคจรเข้าหาสถานีฐานและช่วงท้ายที่ดาวเทียมโคจรออกจากสถานีฐานนั้นค่า TEC ที่ได้มีความคลาดเคลื่อนกัน แต่ในช่วงกลางที่ดาวเทียมโคจรเข้าใกล้สถานีฐานมากที่สุดนั้นค่า TEC ที่ได้มีค่าใกล้เคียงกัน

บทที่ 5

สรุปผลและข้อเสนอแนะ

5.1 สรุปผล

5.1.1 วงจรกรองแถบความถี่ผ่าน

วงจรกรองแถบความถี่ผ่านในช่วง 150 MHz และ 400 MHz นั้นเมื่อสังเกตผลการทดลองที่ได้แสดงไว้ในบทที่ 4 ดังรูปที่ 4.29 และรูปที่ 4.36 นั้นจะเห็นว่าคุณลักษณะของวงจรกรองความถี่ผ่านทั้งสองความถี่มีค่าอัตราลดทอนที่ดีขึ้นเมื่อเทียบกับวงจรกรองที่ได้ทำมาก่อนหน้า นั้นมีผลตอบสนองเป็นไปตามที่ต้องการใช้งาน มีแบนด์วิดท์ใกล้เคียงกับที่ได้ทำการออกแบบไว้คือ 10 MHz และ 27 MHz

5.1.2 โปรแกรมควบคุมการบันทึกสัญญาณดาวเทียม (สัญญาณบีคอน)

จากการทดลองพบว่าโปรแกรมสามารถจัดเรียงลำดับของเวลา สามารถตรวจเช็คการโคจรผ่านของดาวเทียม พร้อมทั้งสามารถบันทึกสัญญาณที่ป้อนจากเครื่องกำเนิดสัญญาณได้ รวมทั้งโปรแกรมสามารถทำงานพื้นฐานเกี่ยวกับข้อมูลจำพวกการอ่านข้อมูลจากอินเทอร์เน็ต การจัดเก็บ ไฟล์ข้อมูล เป็นต้น

5.1.3 การสร้างกราฟฟิสิกอลยสเซอร์อินเทอร์เน็ตเฟส

จากการทดลองพบว่าการคำนวณการโคจรผ่านของดาวเทียม โปรแกรมสามารถสร้างอินเทอร์เน็ตเฟสรับข้อมูลแล้วนำข้อมูลที่ได้เป็นอินพุตให้โปรแกรมทำการคำนวณการโคจรผ่านของดาวเทียม พบว่าไฟล์ที่ได้นั้นได้รับค่าอินพุตตรงกันกับที่ป้อนให้กับ GUI จากนั้นยังสามารถบันทึกสัญญาณจากเครื่องกำเนิดสัญญาณในช่วงเวลาที่ป้อนให้กับ GUI ได้

5.1.4 ภาคการคำนวณผลต่างเฟสของสัญญาณบีคอน

จากการทดลองตามกระบวนการของโพลีชาร์ทของโปรแกรมคำนวณค่าผลต่างเฟสในบทที่ 3 นั้น พบว่าสามารถคำนวณค่าผลต่างเฟสของสองสัญญาณได้ โดยที่เมื่อนำผลต่างเฟสที่คำนวณได้นั้นมาทำการหาค่า TEC เพื่อทำการเปรียบเทียบกับวิธีการของ Kyoto University ซึ่งได้มาตรฐานแล้วพบว่ามีความใกล้เคียงกันแต่ยังมีความแตกต่างกันอยู่บ้างซึ่งเกิดจากค่าผิดพลาดของการคำนวณในขั้นตอนการหาค่าสูงสุดทางกำลังของสัญญาณเนื่องจากวิธีการที่ใช้อยู่ ซึ่งคือการหาค่ากำลังสูงสุดนำมาคำนวณเฟสนั้นบางครั้งค่ากำลังสูงสุดที่วัดได้อาจเป็นค่ากำลังของสัญญาณรบกวน

5.2 ข้อเสนอแนะ

5.2.1 วงจรกรองแถบความถี่ผ่าน

เนื่องจากอุปกรณ์ที่ใช้ในการทดลองนั้นมีความผิดพลาด (Error) เกิดขึ้นซึ่งส่งผลกระทบต่อผลตอบสนองของวงจรกรองช่วงความถี่ที่มีความคลาดเคลื่อนไป ประกอบด้วยผู้ใช้อาจมีความประมาทในการใช้เครื่องมือการทดลองเก็บผลการทดลองต่างๆ ส่งผลให้ได้ผลการทดลองที่คลาดเคลื่อนไป ดังนั้นในการออกแบบและเลือกใช้อุปกรณ์นั้นต้องเลือกใช้อุปกรณ์ที่มีคุณภาพ และผู้ใช้ควรศึกษาวิธีทำการทดลอง ตลอดจนการใช้เครื่องมืออย่างถูกต้องด้วย

5.2.2 โปรแกรมควบคุมการคำนวณหาค่าเฟส

ในส่วนของการคำนวณความต่างเฟสของสัญญาณบีคอนนั้นพบว่าเมื่อนำไปหาค่า TEC แล้ว ยังมีความคลาดเคลื่อนกับค่ามาตรฐานอยู่ ซึ่งอาจแก้ไขได้ด้วยการเพิ่มการคำนวณในส่วนของการหาค่าแอมพลิจูดสูงสุดของสเปกตรัมสัญญาณเพื่อให้โปรแกรมสามารถตรวจจับค่าแอมพลิจูดสูงสุดของสัญญาณได้แม่นยำขึ้น โดยนำแต่ค่าแอมพลิจูดของสัญญาณบีคอนจริงๆ มาคำนวณ โดยไม่ไปคำนวณค่าเฟสของสัญญาณรบกวนมาแทน

5.2.2 ภาคประมวลผลข้อมูลสัญญาณบีคอน

การรับสัญญาณดาวเทียมนั้นจะมีค่าเฉพาะของดาวเทียมแต่ละดวง ซึ่งควรศึกษาค่าเฉพาะพวกนี้ก่อนนำมาใช้งานเพื่อการปรับจูนของสัญญาณให้แม่นยำขึ้น รวมทั้งการตั้งค่า Template ของการควบคุม USRP นั้นมีความสำคัญมากควรศึกษาและตั้งค่าต่างๆ ให้เหมาะสมกับสัญญาณที่รับ

บรรณานุกรม

- [1] Seller, J.J., *Understanding Space, An Introduction to Astronautics*, McGraw-Hill, 3rd edition, 2005.
- [2] Bate, R.R., Mueller, D.D., and White, J.E., *Fundamentals of Astrodynamics*, Dover Publications Inc., 1971.
- [3] Davidoff, M., *The Satellite Experimenter's Handbook*, The American Radio Relay League, 1990
- [4] Curtis, H.D., *Orbital Mechanics for Engineering Students*, Elsevier, 2005.
- [5] สมภพ ภูริวิกรัยพงศ์ และ สุวัฒน์ กุลธนปรีดา "การคำนวณทางโคจรดาวเทียมวงโคจรโลกต่ำ โดยใช้ข้อมูลการวัดสัญญาณจีพีเอส" รายงานวิจัย สำนักงานกองทุนสนับสนุนการวิจัย
- [6] Lyon, G.Richard, *Understanding Digital Signal Processing*, Upper Saddle River, NJ. : Prentice Hall PTR, 2001
- [7] Mamoru Yamamoto. "Digital Beacon Receiver for ionospheric TEC measurement developed with GNU Radio." Kyoto University, 2008
- [8] Radmanesh, M.Matthew, *Radio Frequency and Microwave Electronics Illustrated*, Upper Saddle River, NJ. : Prentice Hall PTR, 2001
- [9] Huelsman, P.Lawrence, และ ศศ.ดร.จิรายุทธ มหันธนกุล, *การออกแบบวงจรแอนะล็อก (Active and Passive Filter Design)*. กรุงเทพมหานคร : ทิพย์จำกัด, 2547
- [10] Williams, B.Arthur, and Taylor, J.Fred, *Electronic Filter Design Handbook*, 4th ed. New York : McGraw-hill, 2006
- [11] <http://www.space.mict.go.th/>
- [12] <http://celestrak.com/columns/v04n03/>
- [13] <http://www.docstoc.com/docs>
- [14] <http://nullege.com/>

Program: Auto satellite prediction and record signal

```

#!/usr/bin/python

import urllib2

import re

import datetime

import sys

import os

import string

from gnuradio import gr, gru, eng_notation, optfir
from gnuradio import usrp
from gnuradio import blks2
from gnuradio.eng_option import eng_option
from optparse import OptionParser
from usrpm import usrp_dbid
import math
from time import sleep
from PyQt4.QtCore import *
from PyQt4.QtGui import *

class wfm_rx_block (gr.top_block):
    def __init__(self):
        gr.top_block.__init__(self)

        f1 = 400.000e6

        f2 = 150.000e6

        op = 80

        term = []

        SAT_1 = [80,'FM5','FM6','CNOFS','RADCAL','DMSPF15'] # +80

        SAT_2 = [-80,'OSCAR23','OSCAR25','OSCAR31','OSCAR32'] # -80

```

```
SAT_3 = [-200,'COSMOS2407','COSMOS2414'] # -200
```

```
SAT_4 = [200,'COSMOS2429'] #+200
```

```
pattern = [SAT_1,SAT_2,SAT_3,SAT_3,SAT_4]
```

```
for t in pattern : ## t = SAT_##
```

```
    for p in t : ## p = DATA in SAT_
```

```
        if (satname == p):
```

```
            op = t[0]
```

```
        else :
```

```
            pass
```

```
term.append(op)
```

```
name_f = datetime.datetime.utcnow()
```

```
cdm = os.getcwd()
```

```
self.file_u = cdm+"/result/test_%s_%s"%(satname,name_f)
```

```
if abs(f1) < 1e6:
```

```
    f1 *= 1e6
```

```
if abs(f2) < 1e6:
```

```
    f2 *= 1e6
```

```
f = (f1,f2)
```

```
g1 = 30
```

```
g2 = 17
```

```
db_gain = (g1,g2)
```

```
# Set tuning frequencies
```

```
#term = options.op.split('')
```

```
term = ["1"]
```

```
foffset = []
```

```
nfreq = 0
```

```
for fj in term:
```

```
    if nfreq >= 3:
```



```

        break

    else:

        nfreq = nfreq + 1

        foffset.append(int(fj))

print 'Frequency: nfreq=',nfreq,', foffset=', foffset

self.vol = .01

self.state = "FREQ"

self.u = usrp.source_c(0, nchan=2) # usrp is data source

adc_rate = self.u.adc_rate() # 64 MS/s
usrp_decim = 200
self.u.set_decim_rate(usrp_decim)
usrp_rate = adc_rate / usrp_decim # 320 kS/s
chanfilt_decim = 10
demod_rate = usrp_rate / chanfilt_decim # 32 kHz

# we're using both daughterboard slots
self.subdev0 = self.u.db(0,0)
self.subdev1 = self.u.db(1,0)

print "Using TX d'board %s" % (self.subdev0.side_and_name(),)
print "Using TX d'board %s" % (self.subdev1.side_and_name(),)

# WBX + "Using Q" config
self.u.set_mux(gru.hexint(0x00002210))

```

```

# deinterleave two channels from FPGA
di = gr.deinterleave(gr.sizeof_gr_complex)

# wire up the head of the chain
self.connect(self.u, di)

# taps for channel filter
chan_filt_coefs = optfir.low_pass (1, # gain
                                   usrp_rate, # sampling rate
                                   14e3, # passband cutoff
                                   24e3, # stopband cutoff
                                   0.1, # passband ripple
                                   60) # stopband attenuation
print "LPF coef len=", len(chan_filt_coefs)

### Set frequency for channel 0 (400MHz)
r = self.u.tune(self.subdev0.which(), self.subdev0, f[0])

if r:
    print "=== WBX side ==="
    print " Target Freq =", f[0]
    print " r.baseband_freq =", eng_notation.num_to_str(r.baseband_freq)
    print " r.dxc_freq =", eng_notation.num_to_str(r.dxc_freq)
    print " r.residual_freq =", eng_notation.num_to_str(r.residual_freq)
    print " r.inverted =", r.inverted
    print " OK"
else:
    print " Failed!"

```

```

### Set frequency for channel 1 (150MHz)
r = self.u.tune(self.subdev1.which(), self.subdev1, f[1])

if r:
    print "=== Basic RX side ==="
    print " Target Freq  =", f[1]
    print " r.baseband_freq =", eng_notation.num_to_str(r.baseband_freq)
    print " r.dxc_freq    =", eng_notation.num_to_str(r.dxc_freq)
    print " r.residual_freq =", eng_notation.num_to_str(r.residual_freq)
    print " r.inverted    =", r.inverted
    print " OK"
else:
    print " Failed!"

# set Gain for both channels
print self.subdev0.set_gain(db_gain[0])
print self.subdev1.set_gain(db_gain[1])

# Multiple tuning with foffset (up to 3 frequencies)
base_MHz = 50.0
for j in range(nfreq):
    f_offtune = (-foffset[j]*base_MHz*8.0, -foffset[j]*base_MHz*3.0)
    if foffset[j] >= 0:
        self.op_txt = 'P%03d' % abs(foffset[j])
    else:
        self.op_txt = 'N%03d' % abs(foffset[j])

```

```

# set graph for channel 0 (400MHz)

chan_filt = gr.freq_xlating_fir_filter_ccf (chanfilt_decim, chan_filt_coefs, f_offtune[0],
usrp_rate)

dst = gr.file_sink(gr.sizeof_gr_complex, "%s_%s_400.dat" % (self.file_u, self.op_txt))
self.connect((di, 0), chan_filt, dst)

# set graph for channel 1 (150MHz)

chan_filt = gr.freq_xlating_fir_filter_ccf (chanfilt_decim, chan_filt_coefs, f_offtune[1],
usrp_rate)

dst = gr.file_sink(gr.sizeof_gr_complex, "%s_%s_305.dat" % (self.file_u, self.op_txt))
self.connect((di, 1), chan_filt, dst)
def save_data(enter, exit, satname):
    fg = wfm_rx_block()
    print "Welcome", satname
    u_now = datetime.datetime.utcnow()
    d = u_now.date()
    start = datetime.datetime.combine(d, enter)
    end = datetime.datetime.combine(d, exit)
    print 'Obs start: ', start, ', end: ', end
    # Open logfile
    daynow = datetime.datetime.utcnow()
    filename = cdm + "/result/test_%s_%s_info.dat" % (satname, daynow)
    fd = open(filename, 'a')

    now = datetime.datetime.utcnow()
    timed = start - now

    while (timed.seconds > 3) & (timed.days != -1):

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

now = datetime.datetime.utcnow()

timed = start - now

sleep(1)

# Last 3 seconds just wait
now = datetime.datetime.utcnow()
timed = start - now
wait_sec = timed.seconds + timed.microseconds * 1.0e-6
#sleep(wait_sec)

fg.start () # fork thread(s) and return
now = datetime.datetime.utcnow()

# Report exact start time into logfile, and close the file
fd.write('EXACTSTART %s\n' % now)
fd.close()

timed = end - now

while (timed.seconds > 1) & (timed.days != -1):
    now = datetime.datetime.utcnow()
    timed = end - now

###    print timed.seconds

    sleep(1)

fg.stop ()

#start Program phase cal

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


```

class star:
    def __init__( self):
        self.name = ""
        self.duration = []
        self.enter = []
        self.exit = []
        self.azimuth = []
        self.elevation = []
    def ppp(self):
        return "Enter time:{0} Time now :{1} Exit time:{2}
".format(self.enter,self._duration,self._exit)
# =====
currentstar = ""
saving = 0
starlist = []
cdm = os.getcwd()
test = open( cdm+"/tablemode/visible" )
# split column in file
all_sat = []
for line in test:
    result = checker.match( line )
    if( result ):
        if (float(result.group(6)) > 30):
            all_sat.append(result.group(1))
            if( result.group(1) != currentstar ):
                print( "new Satellite", result.group(1) )
                saving = star()          # new star class

```

```

starlist.append( saving )

currentstar = result.group(1) # save for next compare

saving.name = result.group(1) # name it

saving.duration.append( float(result.group(2) ))

saving.enter.append( result.group(3) )

saving.exit.append( result.group(4) )

saving.azimuth.append(float( result.group(5) ))

saving.elevation.append( float(result.group(6) ))

else:

    pass

# =====
# try to check that time now is in range ?
enter_time = []
exit_time = []
for number,satname in enumerate(starlist):
    print (number)
    newnumber = number + 1
    for item1 in starlist[number].enter: ##for a-rai in list name a-rai
        newitem1 = item1.split(':')
        newentertime = datetime.time(int(newitem1[0]),int(newitem1[1]),int(newitem1[2]))
        enter_time.append(newentertime)
    for item2 in starlist[number].exit:
        newitem2 = item2.split(':')
        newexittime = datetime.time(int(newitem2[0]),int(newitem2[1]),int(newitem2[2]))
        exit_time.append(newexittime)
for a in range(len(enter_time)):
    for b in range(len(enter_time)-1):
        if (enter_time[b] > enter_time[b+1]):

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

temp_enter = enter_time[b]
enter_time[b] = enter_time[b+1]
enter_time[b+1] = temp_enter

temp_exit = exit_time[b]
exit_time[b] = exit_time[b+1]
exit_time[b+1] = temp_exit

temp_name = all_sat[b]
all_sat[b] = all_sat[b+1]
all_sat[b+1] = temp_name
else : pass

run_grc = 1
ij = 0
k = len(enter_time)
print k,'now'

while(run_grc):
    t2 = datetime.datetime.utcnow().time()
    if(ij != k):
        enter = enter_time[ij]
        exit = exit_time[ij]
        satname = all_sat[ij]

    if (enter < t2 ):
        if (t2 < exit):
            #Run phase analysis

```

```
    pae_pae = save_data(enter,exit,satname)

else :
    ij = ij+1
    #print ("It pass away")
if (ij == k):
    run_grc = 0
    #print "Stop"
#else :
    #print "It not arrive"
else :
    run_grc = 0
sys.exit()
```



Program: Open Website

```
import urllib2
```

```
try :
```

```
    req = urllib2.Request('http://www.rish.kyoto-u.ac.jp/digitalbeacon/information/cosmic.tle')
```

```
    response = urllib2.urlopen(req)
```

```
    the_page = response.read()
```

```
    tle_file = open("/home/pae/Desktop/tle2azel-1.13b/catalog/cosmic.tle","w")
```

```
    tle_file.write(the_page)
```

```
    tle_file.close
```

```
    print ("Internet Connect")
```

```
except :
```

```
    print ("Internet Disconnect")
```

```
pass
```



Program: Graphical user interface

```
#!/usr/bin/python
import sys
import os
import string
import datetime
from PyQt4.QtCore import *
from PyQt4.QtGui import*

## form this program we will get parameter to make txt file for satellite prediction
class Pae_gui_sat(QMainWindow):
    def __init__(self,parent=None,width=600,height=450):
        super(Pae_gui_sat,self).__init__(parent)
        self.setGeometry(100,100,width,height)

        self.pic = QLabel(self)
        self.pixmap = QPixmap("/home/pae/Desktop/images.jpg")
        self.pic.setPixmap(self.pixmap)
        self.pic.setGeometry(30,30,50,50)
        self.sattitle = QLabel("<font color = blue size=30><b>Sattelite
prediction</b></font>",self)
        self.sattitle.setGeometry(100,50,400,35)

        self.local = QLabel("<font size =5 ><b>Location</b></font> ",self)
        self.local.setGeometry(30,100,100,30)

        self.lat = QLabel("<font size=5>Latitude</font>",self)
        self.lat.setGeometry(30,130,100,30)
```

```

self.lat_i = QLineEdit("Latitude",self)
self.lat_i.setGeometry(160,130,200,30)
self.lat_m = QLabel("Degree",self)
self.lat_m.setGeometry(370,130,50,30)

self.long = QLabel("<font size=5>Longitude</font>",self)
self.long.setGeometry(30,160,120,30)
self.long_i = QLineEdit("Longitude",self)
self.long_i.setGeometry(160,160,200,30)
self.long_m = QLabel("Degree",self)
self.long_m.setGeometry(370,160,50,30)

self.alt = QLabel("<font size=5>Altitude</font>",self)
self.alt.setGeometry(30,190,200,30)
self.alt_i = QLineEdit("Altitude",self)
self.alt_i.setGeometry(160,190,200,30)
self.alt_m = QLabel("Meter",self)
self.alt_m.setGeometry(370,190,50,30)

self.time = QLabel("<font size=5 ><b>Time</b></font> ",self)
self.time.setGeometry(30,230,120,30)

self.date = QLabel("<font size=5>Date</font>",self)
self.date.setGeometry(30,255,120,30)
self.year_i = QLineEdit("Year:YYYY",self)
self.year_i.setGeometry(160,255,150,30)
self.mounth_i = QLineEdit("Mounth:MM",self)
self.mounth_i.setGeometry(330,255,150,30)

```

```

self.day_i = QLineEdit("Day:DD",self)
self.day_i.setGeometry(500,255,80,30)

self.start_time = QLabel("<font size=5>Start time</font>",self)
self.start_time.setGeometry(30,290,120,30)
self.hour_i = QLineEdit("24 Hour",self)
self.hour_i.setGeometry(160,290,80,30)
self.min_i = QLineEdit("Minute",self)
self.min_i.setGeometry(260,290,80,30)
self.sec_i = QLineEdit("Second",self)
self.sec_i.setGeometry(360,290,80,30)

self.duration = QLabel("<font size=5>Duration</font>",self)
self.duration.setGeometry(30,330,120,30)
self.duration_i = QLineEdit("",self)
self.duration_i.setGeometry(160,330,150,30)
self.duration_m = QLabel("Minute",self)
self.duration_m.setGeometry(320,330,80,30)

self.button = QPushButton(self)
self.button.setText("Predict")
self.button.setGeometry(250,380,100,40)

self.button.clicked.connect(self.save)

self.licens = QLabel("@By pearpae",self)
self.licens.setGeometry(470,400,200,30)

self.show()

```

```

##Condition of Data type
def save(self):
    correct = 1
    value = []
    try:
        value_lat = float( self.lat_i.text() )
    except:
        value.append("Latitude")
        correct = 0
    try:
        value_long = float( self.long_i.text() )
    except:
        value.append("Longitude")
        correct = 0
    try:
        value_alt = float( self.alt_i.text() )
    except:
        value.append("Altitude")
        correct = 0
    try:
        value_year = int( self.year_i.text() )
    except:
        value.append("Year")
        correct = 0
    try:
        value_mounth = int( self.mounth_i.text() )
        if (value_mounth >12):

```

```

correct = 0
value.append("Mounth")
except:
value.append("Mounth")
correct = 0
try:
value_day = int( self.day_i.text() )
if (value_day > 31):
correct = 0
value.append("Day")
except:
value.append("Day")
correct = 0
try:
value_hour = int( self.hour_i.text() )
if (value_hour > 24):
correct = 0
value.append("Hour")
except:
value.append("Hour")
correct = 0
try:
value_min = int( self.min_i.text() )
if (value_min > 60):
correct = 0
value.append("Minute")
except:
value.append("Minute")

```

```

correct = 0
try:
    value_sec = int( self.sec_i.text() )
    if (value_sec >60):
        correct = 0
        value.append("Secound")
except:
    value.append("Secound")
    correct = 0
try:
    value_duration = int( self.duration_i.text() )
except:
    value.append("Duration")
    correct = 0
#Check if parameter is correct we will record time,location in txt file
if( correct == 1 ):
    self.t = open("/home/tong/Desktop/paac/le2azel-1.13b/config/time.txt","w+")
    self.t.write(self.year_i.text()+"/"+self.mounth_i.text()+"/"+self.day_i.text()+
"+self.hour_i.text()+":"+self.min_i.text()+":"+self.sec_i.text()+","+self.duration_i.text()+",10")
    self.t.close()
    self.l = open("/home/tong/Desktop/paac/le2azel-1.13b/config/location.txt","w+")
    self.l.write(self.lat_i.text()+","+self.long_i.text()+","+self.alt_i.text())
    self.l.close()
    self.close()

```

```

os.system("./tle2azel.py -V -i /home/tong/Desktop/paac/tle2azel-
1.13b/catalog/tle_internet.txt > /home/tong/Desktop/paac/tle2azel-
1.13b/tablemode/gui_visible")
os.system("./ordertime_sys_usrp_cdm_passfilename_gui.py")
elif (correct == 0):
all_value = ",".join(value)
QMessageBox.information(self,"Alert Type","You put "+all_value+" in wrong Type")

myapp = QApplication (sys.argv)
pae = Pae_gui_sat()
myapp.exec_()

```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Program: Phase analysis

```

#!/usr/bin/python
import sys, os, datetime
import matplotlib
#matplotlib.use('Agg')
from matplotlib.pyplot import *
from scipy import *
from scipy import io, fftpack, optimize
import re

def phase_analysis(file150,file400):
    #default parameters

    sat_vel = 0.0

    skip_sec = 0.0

    read_sec = 999.0

    show_sec = read_sec

    # Basic constants define ###
    A_coef = 80.6
    c_speed = 2.998e8 # speed of light
    fr = 50.0e6      # basic frequency

    #This is for test the programme only
    #file150 = '20100830_212407_MUR_rawdata_N200_150.dat'

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

#file400 = '20100830_212407_MUR_rawdata_N200_400.dat'

print '150MHz datafile = ',file150
print '400MHz datafile = ',file400

#check data file and size
print '400MHz datafile = ',file400
data_size = os.stat(file400)[6] #Get filesize
print '150MHz datafile = ',file150
data_size = min(data_size,os.stat(file150)[6]) # Get filesize

print 'Data size in bytes',data_size
print '### Start analysis...###'

#analysis parameter set
sample_f= 32000.0
nfft = 4096
delta_f = sample_f / float(nfft) #frequency resolution

#Calculate length of observation in seconds
if read_sec == 999.0:
    read_sec = float((data_size // (8*sample_f)) - skip_sec) #read_sec = 988.0
    show_sec = read_sec # = 988.0
#print 'read_sec=',read_sec
# Calculate reading parameters

nskip_bytes = int((sample_f * skip_sec) // nfft) * nfft * 8 # nskip_bytes = 0

```

```

ntdata = int((sample_f * read_sec) // nfft) * nfft #ntdata = 31612928

nparam = nntdata / nfft -1 # Number of spectral data (half interleave)

print 'total data number = ',ntdata, ', read_second = ', nntdata/sample_f
print 'skip second = ', skip_sec , 'second of data presentation = ',show_sec

# Array def

buf_150 = arange(ntdata, dtype='F') #databuffer for 150MHz Beacon receiver
buf_400 = arange(ntdata, dtype='F') #databuffer for 150MHz Beacon receiver

ser_150 = arange(nfft, dtype='D')
ser_400 = arange(nfft, dtype='D')

fft_result400 = arange(nfft, dtype='D') #fft_result = [0.0+0.j ...8191.0+0.j]
fft_result150 = arange(nfft, dtype='D') #fft_result = [0.0+0.j ...8191.0+0.j]

value_max400 = [] #keep max values
value_max150 = []

n_overlap = 250/(sample_f/nfft) #32
t_width = nfft/n_overlap # 128
t_total = nntdata/t_width #246976

npfact=128

TEC = arange(t_total, dtype='f')

```

```
phase_sec = arange(t_total, dtype = 'f') * (float(npfact)/sample_f) + skip_sec
```

```
# Open data files and skip
```

```
f150 = open(file150, 'rb')
```

```
f150.seek(nskip_bytes)
```

```
f400 = open(file400, 'rb')
```

```
f400.seek(nskip_bytes)
```

```
# 400MHz data read
```

```
buf_400 = io.fread(f400, ntdata, 'F')
```

```
#print buf_400
```

```
#buf_4001 = real(buf_400)
```

```
# 150MHz data read
```

```
buf_150 = io.fread(f150, ntdata, 'F')
```

```
#buf_1501 = real(buf_150)
```

```
#print buf_150
```

```
t_length = 250/(sample_f/nfft) #32
```

```
t_width = nfft/t_length # 128
```

```
t_total = ntdata/t_width #246976
```

```
x_sec = arange(nparam, dtype = 'f') * (float(nfft)/sample_f) + skip_sec
```

```
freq = arange(nfft, dtype = 'f') * delta_f - sample_f/2.0
```

```
for ip in range(0,t_total):
```

```
    if (ip % 100) == 0:
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

print 'Phase calculating....ip = ',ip

noffset = ip * nfft/t_length

ser_400 = buf_400[noffset: noffset+nfft]
ser_150 = buf_150[noffset: noffset+nfft]

fft_result400 = fft(ser_400)/nfft
fft_result150 = fft(ser_150)/nfft

#fft_result400 = fftpack.ffshift(fft_result400) #Rotate freq
#fft_result150 = fftpack.ffshift(fft_result150) #Rotate freq

result400 = fft_result400[argmax(abs(fft_result400))]
result150 = fft_result150[argmax(abs(fft_result150))]

x_sec = arange(nparam, dtype='f') * (float(nfft)/sample_f/2.0) + skip_sec

value_max400.append(result400)
value_max150.append(result150)

##showfile(result400,'value400.txt')

phase400 = angle(value_max400)
phase400n = phase400*3

phase150 = angle(value_max150)
phase150n = phase150*8

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

phasedifference = phase150n-phase400n
unwrapphase = unwrap(phasedifference)

###calculate TEC value
coef = pi * A_coef / fr / c_speed * 55.0 / 24.0

TEC = (unwrapphase/(coef*1e16))
#plot(TEC)

#test = "test_COSMOS2429_2011-01-24 11:09:50.045116_P080_400.dat"
checker = re.compile(r"(\S+)\_(\S+)\_(\S+)\s+(\S+)\.+(\S+)\_(\S+)\_(\S+)\.dat$")
result = checker.match(file150)
base_dir = os.getcwd()
Out_filename = base_dir + '/Phase_result/' + result.group(2) + "_" + result.group(3) + "_" +
result.group(4) + '_summary.txt'

fig1 = figure()
plot(TEC)
xlabel('time(sec)')
ylabel('TECu')
title('TEST_TEC_%s'%result.group(2)) # put name
fig_filename1 = base_dir + '/Phase_result/' + 'TEC_%s.png'% result.group(2) # put name
savefig(fig_filename1)

#show()
# File close
f150.close

```

```

f400.close
del buf_400
del buf_150

npfact = 128
npft = nfft/npfact

### file output ###

f_out1 = open(Out_filename, 'w')
f_out1.write('time(s) TEC(TECu) STD_TEC (Total %d lines)\n'% nparam)

for ip in range(0, nparam):
    ist = ip*npst*2 + npst
    ien = ist + npst*2
    ave_TEC = TEC[ist:ien].mean()/1.0e16 # averaged TEC in TECu
    std_TEC = TEC[ist:ien].std()/1.0e16 # its standard deviation in TECu
    write_buf = '%g, %g, %g\n' %(x_sec[ip], ave_TEC, std_TEC)
    f_out1.write(write_buf)

f_out1.close

```

```
#rx file name from Pae's GRBRscript
def main() :

    print "Arument_are_coming"

    print sys.argv
    file150 = '%s %s'%(sys.argv[1],sys.argv[2])

    file400 = '%s %s'%(sys.argv[3],sys.argv[4])

    phase_analysis(file150,file400)
```

```
main()
```

```
110
```

