การประมาณหาที่ตั้งแหล่งกำเนิดสัญญาณจากการวัดมุมทิศ โดยใช้วิธี
การพัฒนาคาลมานฟิลเตอร์แบบขยาย

หอสมุดกลาง สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ร้อยเอก เอกพล จูฑะพันธุ์

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต
สาขาวิศวกรรมไฟฟ้า ภาควิชาวิศวกรรมไฟฟ้า
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2542

ISBN 974-333-546-3

ลิขสิทธิ์ของ จุฬาลงกรณ์มหาวิทยาลัย

Emitter Localization Estimation from Bearing Angle
Measurements by Modified Extended Kalman Filter Method

Captain Ekabhol Chudhabhan

A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree of Master of Engineering in Electrical Engineering
Department of Electrical Engineering
Faculty of Engineering
Chulalongkorn University
Academic Year 1999
ISBN 974-333-546-3

หัวข้อวิทยานิพนธ์	การประมาณหาที่ตั้งแหล่งกำเนิดสัญญาณจากการวัดมุมทิศ โดยใช้วิธี การพัฒนาคาลมานฟิลเตอร์แบบขยาย
โดย	ร้อยเอก เอกพล จูฑะพันธุ์
ภาควิชา	วิศวกรรมไฟฟ้า
อาจารย์ที่ปรึกษา	รองศาสตราจารย์ ดร. สมชาย จิตะพันธ์กุล
อาจารย์ที่ปรึกษาร่วม	นาวาอากาศเอก ดร. เพียร โตท่าโรง

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้นับวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญามหาบัณฑิต

....คณบดี คณะวิศวกรรมศาสตร์
(รองศาสตราจารย์ ดร. ธัชชัย สุมิตร)

คณะกรรมการสอบวิทยานิพนธ์

....ประธานกรรมการ
(รองศาสตราจารย์ ดร.เทียนชัย ประดิสถายน)

....อาจารย์ที่ปรึกษา
(รองศาสตราจารย์ ดร.สมชาย จิตะพันธ์กุล)

....อาจารย์ที่ปรึกษาร่วม
(นาวาอากาศเอก ดร.เพียร โตท่าโรง)

....กรรมการ
(รองศาสตราจารย์ ดร.วาทิต เบญจพลกุล)

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

เอกพล จูฑะพันธุ์, ร.อ. : การประมาณหาที่ตั้งแหล่งกำเนิดสัญญาณจากการวัดมุมทิศ โดยใช้วิธีการพัฒนาคาลมานฟิลเตอร์แบบขยาย (Emitter Localizatioin Estimation from Bearing Angle Measurements by Modified Extended Kalman Filter Method)
อ.ที่ปรึกษา : รศ.ดร.สมชาย จิตะพันธ์กุล, อ.ที่ปรึกษาร่วม : น.อ. ดร.เพียร โตท่าโรง
129 หน้า.

วิทยานิพนธ์ฉบับนี้นำเสนอแนวทางและ กรรมวิธีการพัฒนา อัลกอริทึมของงานทางด้าน Passive location estimation ทั้งในกรณีของ Moving observer และ Multiple fixed observers โดยมุ่งเน้นประเด็นไปในเรื่องของการเพิ่มความแม่นในการพิสูจน์ทราบตำแหน่งของแหล่งกำเนิดสัญญาณ ด้วยวิธี Modified Extended Kalman Filter (MEKF) โดยเปรียบเทียบกับอัลกอริทึมของ Extended Kalman Filter (EKF) ซึ่งได้มีการนำเสนอมาแล้ว การพัฒนาอัลกอริทึมอาศัยหลักการของ Linearization และ การประมวลผลแบบขนาน (Parallel algorithm) ซึ่งจากผลการทดสอบชี้ให้เห็นว่าผลที่ได้จากกรรมวิธีของ MEKF นั้นให้ความแม่นยำกว่า ผลที่ได้จาก กรรมวิธีของ EKF

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ภาควิชา	วิศวกรรมไฟฟ้า	ลายมือชื่อนิสิต	ร.อ. [signature]
สาขาวิชา	วิศวกรรมไฟฟ้า	ลายมือชื่ออาจารย์ที่ปรึกษา	[signature]
ปีการศึกษา	2542	ลายมือชื่ออาจารย์ที่ปรึกษาร่วม	น.อ. [signature]

397 25498 21 : MAJOR DIGITAL SIGNAL PROCESSING
KEY WORDS : Modified Extended Kalman Filter (MEKF)/Location Estimation (LOC.EST)/ Bearing Angle Estimation

EKABHOL CHUDHABHAN, CAPT. : Emitter Localization Estimation from Bearing Angle Measurements by Modified Extended Kalman Filter Method.
THESIS ADVISOR: ASSC.PROF. SOMCHAI JITAPUNKUL, Dr.Ing.
THESIS COADVISOR: Group Capt. PIAN TOTARONG, Ph.D.
129 pp.

In the field of passive position location estimation , the modified extended Kalman filter (MEKF) is proposed to improve the accuracy of moving observers and multiple fixed observers measurement system. The extended Kalman filter (EKF) has already been shown that it was one of the technique in this case. The modified version is achieved by an improved linearization procedure and , a parallel computation scheme. The simulation results shows that the performance of the MEKF method is better than the EKF method.

จุฬาลงกรณ์มหาวิทยาลัย

ภาควิชา	วิศวกรรมไฟฟ้า	ลายมือชื่อนิสิต	ร.อ.
สาขาวิชา	วิศวกรรมไฟฟ้า	ลายมือชื่ออาจารย์ที่ปรึกษา	
ปีการศึกษา	2542	ลายมือชื่ออาจารย์ที่ปรึกษาร่วม	Pian Totarong.

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้ สำเร็จลงได้เนื่องจากผู้วิจัยได้รับคำแนะนำ คำปรึกษา และการช่วยเหลือเป็นอย่างดียิ่ง จากรองศาสตราจารย์ ดร.สมชาย จิตะพันธ์กุล อาจารย์ที่ปรึกษาวิทยานิพนธ์ และ อีกท่านหนึ่งคือ น.อ. ดร.เพียร โตท่าโรง ผู้อำนวยการกองวิจัยและพัฒนาการสื่อสารและอิเล็กทรอนิกส์ ศูนย์วิจัย และพัฒนาการทหาร กองบัญชาการทหารสูงสุด อาจารย์ที่ปรึกษาร่วมวิทยานิพนธ์ ตั้งแต่การเสนอโครงร่างวิทยานิพนธ์ การรวบรวมแหล่งที่มาข้อมูล การวิเคราะห์รูปแบบของระบบ การทดสอบและประมวลผลของระบบ การบันทึกข้อผิดพลาด การแก้ไขข้อบกพร่อง ตลอดจนการสรุปผลและ เรียบเรียงเป็นเอกสารวิทยานิพนธ์ ทำให้ผู้วิจัยสามารถเรียบเรียงวิทยานิพนธ์ฉบับนี้สำเร็จลงได้ด้วยดี

นอกจากทั้งสองท่านที่ได้กล่าวมาแล้วข้างต้น ผู้วิจัยยังได้รับความกรุณาจาก

ร.อ.นพรัตน์	แสงหนุ่ม	อาจารย์โรงเรียนทหารสื่อสาร
ร.อ.พชรพล	สินธุวงศานนท์	อาจารย์โรงเรียนทหารสื่อสาร
ร.อ.เอกวัฒน	กิรติรัตนพฤกษ์	ผบ.ร้อย.รร.สส.

ซึ่งท่านเหล่านี้ได้ให้คำแนะนำ และให้กำลังใจเสมอมา ทำให้วิทยานิพนธ์ฉบับนี้สมบูรณ์และ สำเร็จลงได้ด้วยดี ท้ายที่สุดนี้ ผู้วิจัยใคร่ขอกราบขอบพระคุณอย่างสูงที่สุด แด่ พ.อ.ไฉน จูฑะพันธุ์ และอาจารย์ไขศรี จูฑะพันธุ์ ซึ่งเป็น บิดาและ มารดาของผู้วิจัย ซึ่งทั้งสองท่านได้ให้การสนับสนุนและ เป็นแรงใจให้แก่ผู้วิจัยมาโดยตลอดจนสำเร็จการศึกษา

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

สารบัญ

หน้า

บทคัดย่อภาษาไทย.....ง

บทคัดย่อภาษาอังกฤษ.....จ

กิตติกรรมประกาศ.....ฉ

สารบัญตาราง.....ญ

สารบัญภาพ.....ฎ

บทที่ 1. บทนำ.....1

ความเป็นมาและความสำคัญของปัญหา.....1

วัตถุประสงค์.....2

ขอบเขตของการวิจัย.....3

วิธีการดำเนินการค้นคว้าและวิจัย.....3

ประโยชน์ที่คาดว่าจะได้รับ.....3

บทที่ 2. การประมาณหาที่ตั้งของแหล่งกำเนิดสัญญาณ.....4

รูปแบบของการประมาณหาที่ตั้งแหล่งกำเนิดสัญญาณ.....4

1. Moving Observer Measurements.....4
2. Multiple fixed Observer Measurements.....5

การประมาณหาที่ตั้งแหล่งกำเนิดสัญญาณ จากการวัดมุมทิศ.....8

1. Moving Observer Measurements.....8
2. Multiple fixed Observer Measurements.....8

อัลกอริทึมสำหรับการประมาณหาที่ตั้งแหล่งกำเนิดสัญญาณจากการวัดมุมทิศ

1. Standard Kalman filter (SKF).....10
2. Torrieri's algorithm.....13
3. Extended Kalman filter (EKF).....17

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

สารบัญ (ต่อ)

หน้า

อัลกอริทึมที่นำเสนอ....20

1. รายละเอียดของอัลกอริทึม Modified extended Kalman Filler (MEKF) ที่นำเสนอ....23

Modified extended Kalman filter (MEKF)....23

EKFT....25

EKFS....28

SKF....30

บทที่ 3. การจำลองบนคอมพิวเตอร์....35

การสร้างแบบจำลองบนคอมพิวเตอร์....35

Moving observer measurement....35

Multiple fixed observers measurement....36

รูปแบบของการผลที่ได้จากการทดสอบ....37

ผลการทดสอบที่ได้จากคอมพิวเตอร์....38

Moving observer measurement (EKF & MEKFS)....38

ผลการทดสอบกรณีที่ 1 จำนวน 2 Observations....38

ผลการทดสอบกรณีที่ 2 จำนวน 9 Observations....40

Multiple fixed observer measurement (EKF & MEKFS)....42

ผลการทดสอบกรณีที่ 1 จำนวน 2 Observations และ Emitter อยู่ใน Observers area....42

ผลการทดสอบกรณีที่ 2 จำนวน 2 Observations และ Emitter อยู่นอก Observers area....46

ผลการทดสอบกรณีที่ 3 จำนวน 3 Observations และ Emitter อยู่ใน Observers area....50

ผลการทดสอบกรณีที่ 4 จำนวน 3 Observations และ Emitter อยู่นอก Observers area....54

ผลการทดสอบกรณีที่ 5 จำนวน 4 Observations และ Emitter อยู่ใน Observers area....58

ผลการทดสอบกรณีที่ 6 จำนวน 4 Observations และ Emitter อยู่นอก Observers area....62

Multiple fixed observer measurement (KF+TR & MKF+TR)....66

ผลการทดสอบกรณีที่ 1 จำนวน 2 Observations และ Emitter อยู่ใน Observers area....66

ผลการทดสอบกรณีที่ 2 จำนวน 2 Observations และ Emitter อยู่นอก Observers area....68

ผลการทดสอบกรณีที่ 3 จำนวน 3 Observations และ Emitter อยู่ใน Observers area....70

ผลการทดสอบกรณีที่ 4 จำนวน 3 Observations และ Emitter อยู่นอก Observers area....72

สารบัญ (ต่อ)

หน้า

ผลการทดสอบกรณีที่ 5 จำนวน 4 Observations และ Emitter อยู่ใน Observers area............74

ผลการทดสอบกรณีที่ 6 จำนวน 4 Observations และ Emitter อยู่นอก Observers area.........76

การแสดงผลการทดสอบในรูป Circular Error Probability..78

กรณีของ EKF...78

จำนวน 10 Observations...78

จำนวน 20 Observations...79

จำนวน 30 Observations...79

จำนวน 40 Observations...80

กรณีของ MEKF..82

จำนวน 10 Observations...82

จำนวน 20 Observations...82

จำนวน 30 Observations...83

จำนวน 40 Observations...83

สรุปผลการทดสอบ...85

1. Moving Observers (EKF & MEKFS)..85

2. Multiple fixed Observers (EKF & MEKFS)..86

3. Multiple fixed Observers (KF+TR & MKF+TR)..86

วิจารณ์ผลการทดสอบ..94

บทที่ 4. ข้อสรุป และข้อเสนอแนะ...100

ข้อสรุป...100

ข้อเสนอแนะ...102

รายการอ้างอิง..103

ภาคผนวก...104

โปรแกรมการประมวลผล EKF กรณีที่ 1..104

โปรแกรมการประมวลผล MEKF#1 กรณีที่ 1..107

โปรแกรมการประมวลผล MEKF#2 กรณีที่ 1..110

โปรแกรมการประมวลผล MEKF#3 กรณีที่ 1..113

โปรแกรมการประมวลผล MEKF#4 กรณีที่ 1..116

โปรแกรมการประมวลผล MEKF#5 กรณีที่ 1..119

โปรแกรมการประมวลผล MKF+TR 126

ประวัติผู้เขียน 129

สารบัญตาราง

หน้า

ตารางที่ 2.1 MEKF ทั้ง 5 กรณี....24
ตารางที่ 3.1 การเปรียบเทียบ Position estimates ของรูปที่ 3.1....39
ตารางที่ 3.2 การเปรียบเทียบ Position estimates ของรูปที่ 3.2....39
ตารางที่ 3.3 การเปรียบเทียบ MS Position error ของรูปที่ 3.3....39
ตารางที่ 3.4 การเปรียบเทียบ MS Position error ของรูปที่ 3.4....41
ตารางที่ 3.5 การเปรียบเทียบ MS Position error ของรูปที่ 3.5....41
ตารางที่ 3.6 การเปรียบเทียบ MS Position error ของรูปที่ 3.6....41
ตารางที่ 3.7 การเปรียบเทียบ MS Position error ของรูปที่ 3.7....43
ตารางที่ 3.8 การเปรียบเทียบ MS Position error ของรูปที่ 3.8....45
ตารางที่ 3.9 การเปรียบเทียบ MS Position error ของรูปที่ 3.9....47
ตารางที่ 3.10 การเปรียบเทียบ MS Position error ของรูปที่ 3.10....49
ตารางที่ 3.11 การเปรียบเทียบ MS Position error ของรูปที่ 3.11....51
ตารางที่ 3.12 การเปรียบเทียบ MS Position error ของรูปที่ 3.12....53
ตารางที่ 3.13 การเปรียบเทียบ MS Position error ของรูปที่ 3.13....55
ตารางที่ 3.14 การเปรียบเทียบ MS Position error ของรูปที่ 3.14....57
ตารางที่ 3.15 การเปรียบเทียบ MS Position error ของรูปที่ 3.15....59
ตารางที่ 3.16 การเปรียบเทียบ MS Position error ของรูปที่ 3.16....61
ตารางที่ 3.17 การเปรียบเทียบ MS Position error ของรูปที่ 3.17....63
ตารางที่ 3.18 การเปรียบเทียบ MS Position error ของรูปที่ 3.18....65
ตารางที่ 3.19 การเปรียบเทียบ MS Position error ของรูปที่ 3.19....67
ตารางที่ 3.20 การเปรียบเทียบ MS Position error ของรูปที่ 3.20....67
ตารางที่ 3.21 การเปรียบเทียบ MS Position error ของรูปที่ 3.21....69
ตารางที่ 3.22 การเปรียบเทียบ MS Position error ของรูปที่ 3.22....69
ตารางที่ 3.23 การเปรียบเทียบ MS Position error ของรูปที่ 3.23....71
ตารางที่ 3.24 การเปรียบเทียบ MS Position error ของรูปที่ 3.24....71
ตารางที่ 3.25 การเปรียบเทียบ MS Position error ของรูปที่ 3.25....73

สารบัญตาราง (ต่อ)

หน้า

ตารางที่ 3.26 การเปรียบเทียบ MS Position error ของรูปที่ 3.26....73

ตารางที่ 3.27 การเปรียบเทียบ MS Position error ของรูปที่ 3.27....75

ตารางที่ 3.28 การเปรียบเทียบ MS Position error ของรูปที่ 3.28....75

ตารางที่ 3.29 การเปรียบเทียบ MS Position error ของรูปที่ 3.29....77

ตารางที่ 3.30 การเปรียบเทียบ MS Position error ของรูปที่ 3.30....77

ตารางที่ 3.31 ค่า Circular Error Probability ของรูปที่ 3.34....81

ตารางที่ 3.32 ค่า Circular Error Probability ของรูปที่ 3.38....84

Moving observers (EKF & MEKFS)

ตารางที่ 3.33 ลำดับค่าความแม่นยำของการเปรียบเทียบ ระหว่าง EKF & MEKFS ทั้ง 5 ประเภท..85

ตารางที่ 3.34 ลำดับค่าความแม่นยำของการเปรียบเทียบ ระหว่าง EKF & MEKFS ทั้ง 5 ประเภท..85

Multiple fixed observers (EKF & MEKFS)

ตารางที่ 3.35 ลำดับค่าความแม่นยำของการเปรียบเทียบ ระหว่าง EKF & MEKFS ทั้ง 5 ประเภท..86

ตารางที่ 3.36 ลำดับค่าความแม่นยำของการเปรียบเทียบ ระหว่าง KF+TR & MKF+TR....86

Circular Error Probability (CEP)

ตารางที่ 3.37 ลำดับค่าความแม่นยำในกรณีของ Moving observers กรณี 2 observers....87

ตารางที่ 3.38 ลำดับค่าความแม่นยำในกรณีของ Moving observers กรณี 9 observers....87

ตารางที่ 3.39 ลำดับค่าความแม่นยำในกรณีของ Multiple fixed observers กรณีที่ 1....88

ตารางที่ 3.40 ลำดับค่าความแม่นยำในกรณีของ Multiple fixed observers กรณีที่ 2....88

ตารางที่ 3.41 ลำดับค่าความแม่นยำในกรณีของ Multiple fixed observers กรณีที่ 3....89

ตารางที่ 3.42 ลำดับค่าความแม่นยำในกรณีของ Multiple fixed observers กรณีที่ 4....89

ตารางที่ 3.43 ลำดับค่าความแม่นยำในกรณีของ Multiple fixed observers กรณีที่ 5....90

ตารางที่ 3.44 ลำดับค่าความแม่นยำในกรณีของ Multiple fixed observers กรณีที่ 6....90

จุฬาลงกรณ์มหาวิทยาลัย

สารบัญภาพ

หน้า

รูปที่ 2.1 ระบบการประมาณหาที่ตั้งแหล่งกำเนิดสัญญาณ....4
รูปที่ 2.2 Passive position location system geometry....5
รูปที่ 2.3 Intersecting hyperbolas from three observers....6
รูปที่ 2.4 Stationary emitter localization....7
รูปที่ 2.5 Flowchart for Kalman filter algorithm....12
รูปที่ 2.6 Geometry of transmitter, reference point, and station....13
รูปที่ 2.7 รูปแบบของ Observers แบบต่างๆ ที่ใช้ในการทดสอบ....22
รูปที่ 2.8 Parallel algorithm for MEKF....24

Moving observer (EKF & MEKFS)

กรณีที่ 1 - จำนวน 2 Observations

รูปที่ 3.1 Position estimates ระหว่าง EKF & MEKFS ประเภทต่างๆ บนแกน X....38
รูปที่ 3.2 Position estimates ระหว่าง EKF & MEKFS ประเภทต่างๆ บนแกน Y....38
รูปที่ 3.3 MS Position error ระหว่าง EKF & MEKFS ประเภทต่างๆ บนระนาบ XY....38

กรณีที่ 2 - จำนวน 9 Observations

รูปที่ 3.4 MS Position error ระหว่าง EKF & MEKFS ประเภทต่างๆ บนแกน X....40
รูปที่ 3.5 MS Position error ระหว่าง EKF & MEKFS ประเภทต่างๆ บนแกน Y....40
รูปที่ 3.6 MS Position error ระหว่าง EKF & MEKFS ประเภทต่างๆ บนระนาบ XY....40

Multiple fixed observers (EKF & MEKFS)

กรณีที่ 1 - จำนวน 2 Observations (Emitter in observers area)

รูปที่ 3.7 MS Position error ระหว่าง EKF & MEKFS ประเภทต่างๆ บนแกน X....42
รูปที่ 3.7.1 MS Position error ระหว่าง MEKFS ประเภทต่างๆ บนแกน X....42
รูปที่ 3.8 MS Position error ระหว่าง EKF & MEKFS ประเภทต่างๆ บนแกน Y....44
รูปที่ 3.8.1 MS Position error ระหว่าง MEKFS ประเภทต่างๆ บนแกน Y....44

กรณีที่ 1 - จำนวน 2 Observations (Emitter out observers area)

รูปที่ 3.9 MS Position error ระหว่าง EKF & MEKFS ประเภทต่างๆ บนแกน X....46
รูปที่ 3.9.1 MS Position error ระหว่าง MEKFS ประเภทต่างๆ บนแกน X....46
รูปที่ 3.10 MS Position error ระหว่าง EKF & MEKFS ประเภทต่างๆ บนแกน Y....48.
รูปที่ 3.10.1 MS Position error ระหว่าง MEKFS ประเภทต่างๆ บนแกน Y....48

สารบัญภาพ (ต่อ)

หน้า

กรณีที่ 3 – จำนวน 3 Observations (Emitter in observers area)
รูปที่ 3.11 MS Position error ระหว่าง EKF & MEKFS ประเภทต่างๆ บนแกน X 50
รูปที่ 3.11.1 MS Position error ระหว่าง MEKFS ประเภทต่างๆ บนแกน X 50
รูปที่ 3.12 MS Position error ระหว่าง EKF & MEKFS ประเภทต่างๆ บนแกน Y 52
รูปที่ 3.12.1 MS Position error ระหว่าง MEKFS ประเภทต่างๆ บนแกน Y 52
กรณีที่ 4 – จำนวน 3 Observations (Emitter out observers area)
รูปที่ 3.13 MS Position error ระหว่าง EKF & MEKFS ประเภทต่างๆ บนแกน X 54
รูปที่ 3.13.1 MS Position error ระหว่าง MEKFS ประเภทต่างๆ บนแกน X 54
รูปที่ 3.14 MS Position error ระหว่าง EKF & MEKFS ประเภทต่างๆ บนแกน Y 56
รูปที่ 3.14.1 MS Position error ระหว่าง MEKFS ประเภทต่างๆ บนแกน Y 56
กรณีที่ 5 – จำนวน 4 Observations (Emitter in observers area)
รูปที่ 3.15 MS Position error ระหว่าง EKF & MEKFS ประเภทต่างๆ บนแกน X 58
รูปที่ 3.15.1 MS Position error ระหว่าง MEKFS ประเภทต่างๆ บนแกน X 58
รูปที่ 3.16 MS Position error ระหว่าง EKF & MEKFS ประเภทต่างๆ บนแกน Y 60
รูปที่ 3.16.1 MS Position error ระหว่าง MEKFS ประเภทต่างๆ บนแกน Y 60
กรณีที่ 6 – จำนวน 4 Observations (Emitter out observers area)
รูปที่ 3.17 MS Position error ระหว่าง EKF & MEKFS ประเภทต่างๆ บนแกน X 62
รูปที่ 3.17.1 MS Position error ระหว่าง MEKFS ประเภทต่างๆ บนแกน X 62
รูปที่ 3.18 MS Position error ระหว่าง EKF & MEKFS ประเภทต่างๆ บนแกน Y 64
รูปที่ 3.18.1 MS Position error ระหว่าง MEKFS ประเภทต่างๆ บนแกน Y 64
Multiple fixed observers (KF+TR & MKF+TR)
กรณีที่ 1 – จำนวน 2 Observations (Emitter in observers area)
รูปที่ 3.19 MS Position error ระหว่าง KF+TR & MKF+TR บนแกน X 66
รูปที่ 3.20 MS Position error ระหว่าง KF+TR & MKF+TR บนแกน Y 66
กรณีที่ 2 – จำนวน 2 Observations (Emitter out observers area)
รูปที่ 3.21 MS Position error ระหว่าง KF+TR & MKF+TR บนแกน X 68
รูปที่ 3.22 MS Position error ระหว่าง KF+TR & MKF+TR บนแกน Y 68

สารบัญภาพ (ต่อ)

หน้า

กรณีที่ 3 - จำนวน 3 Observations (Emitter in observers area)
รูปที่ 3.23 MS Position error ระหว่าง KF+TR & MKF+TR บนแกน X 70
รูปที่ 3.24 MS Position error ระหว่าง KF+TR & MKF+TR บนแกน Y 70
กรณีที่ 4 - จำนวน 3 Observations (Emitter out observers area)
รูปที่ 3.25 MS Position error ระหว่าง KF+TR & MKF+TR บนแกน X 72
รูปที่ 3.26 MS Position error ระหว่าง KF+TR & MKF+TR บนแกน Y 72
กรณีที่ 5 - จำนวน 2 Observations (Emitter in observers area)
รูปที่ 3.27 MS Position error ระหว่าง KF+TR & MKF+TR บนแกน X 74
รูปที่ 3.28 MS Position error ระหว่าง KF+TR & MKF+TR บนแกน Y 74
กรณีที่ 6 - จำนวน 2 Observations (Emitter out observers area)
รูปที่ 3.29 MS Position error ระหว่าง KF+TR & MKF+TR บนแกน X 76
รูปที่ 3.30 MS Position error ระหว่าง KF+TR & MKF+TR บนแกน Y 76
การแสดงผลการทดสอบในรูป Circular Error Probability (CEP)
รูปที่ 3.31 Circular Error Probability 100 Monte Carlo runs, 10 Observations (EKF)...............78
รูปที่ 3.32 Circular Error Probability 100 Monte Carlo runs, 20 Observations (EKF)...............79
รูปที่ 3.33 Circular Error Probability 100 Monte Carlo runs, 30 Observations (EKF)...............79
รูปที่ 3.34 Circular Error Probability 100 Monte Carlo runs, 40 Observations (EKF)...............80
รูปที่ 3.35 Circular Error Probability 100 Monte Carlo runs, 10 Observations (MEKF)............78
รูปที่ 3.36 Circular Error Probability 100 Monte Carlo runs, 20 Observations (MEKF)............79
รูปที่ 3.37 Circular Error Probability 100 Monte Carlo runs, 30 Observations (MEKF)............79
รูปที่ 3.34 Circular Error Probability 100 Monte Carlo runs, 40 Observations (MEKF)............80

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

อภิธานคำศัพท์

(Standard) Kalman filter	คาลมานฟิลเตอร์แบบมาตรฐาน
Extended Kalman filter	คาลมานฟิลเตอร์แบบขยาย
Modified extended Kalman filter	คาลมานฟิลเตอร์แบบขยายชนิดดัดแปลง
Observer	เครื่องดักรับหาทิศของสัญญาณ
Electronic Warfare Support Measures	มาตรการสนับสนุนอิเล็กทรอนิกส์
Electronic Counter Measurement	มาตรการต่อต้านทางอิเล็กทรอนิกส์
Electronic Counter Counter Measurement	มาตรการตอบโต้การต่อต้านทางอิเล็กทรอนิกส์
Bearing angle measurement	การวัดมุมทิศ
Bearing line	เส้นทาง/ทิศทางของมุมทิศ
Moving observer	เครื่องดักรับหาทิศของสัญญาณแบบเคลื่อนที่
Fixed observers	เครื่องดักรับหาทิศของสัญญาณแบบอยู่กับที่
Parallel algorithm	กรรมวิธีการประมวลผลแบบคู่ขนาน
Hopping frequency	ความถี่ก้าวกระโดด
Jamming	การรบกวนสัญญาณวิทยุต่อฝ่ายเรา
Line of bearing	เส้นทางหรือทิศทางของสัญญาณ
Emitter location, Source	แหล่งกำเนิดสัญญาณ
Location estimation	การประมาณที่ตั้งแหล่งกำเนิดสัญญาณ

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

หอสมุดกลาง สถาบันวิทยบริการ จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 1

บทนำ

ความเป็นมาและความสำคัญของปัญหา

สืบเนื่องจากการที่กองทัพไทยได้เล็งเห็นถึงความสำคัญของ ระบบสงครามอิเล็กทรอนิกส์และ ได้มีนโยบายในการขยายและผลักดัน การพัฒนาระบบสงครามอิเล็กทรอนิกส์ของกองทัพ ให้เกิดประสิทธิภาพสูงสุดนั้น ได้มีหลายหน่วยงานและ หลายบุคลากรที่ได้ศึกษาและ ค้นคว้าวิจัยภารกิจทางด้านนี้โดยเฉพาะ ซึ่งในแต่ละส่วนนั้น ได้มีการดำเนินการพัฒนาอย่างต่อเนื่องมาโดยตลอด ซึ่งในการพัฒนาดังกล่าวนี้ จะแบ่งออกเป็น 3 แนวทางการพัฒนาใหญ่ๆนั่นคือ **การพัฒนามาตรการสนับสนุนอิเล็กทรอนิกส์ (Electronic Warfare Support Measures: ESM)** กล่าวคือเป็นการพัฒนาปฏิบัติการค้นหา ดักรับ พิสูจน์ทราบ บันทึก และวิเคราะห์เพื่อรวบรวมเป็นข้อมูลนำมาสนับสนุน ECM, ECCM และการใช้กำลังทางยุทธวิธีอื่นๆ อาทิเช่น Radio direction finding เป็นต้น, **การพัฒนามาตรการต่อต้านทางอิเล็กทรอนิกส์ (Electronic Counter Measurement: ECM)** กล่าวคือเป็นการพัฒนาการปฏิบัติที่กระทำเพื่อขัดขวางหรือลดประสิทธิผลการใช้คลื่นแม่เหล็กไฟฟ้าของข้าศึกอาทิเช่น Jamming, การลวงด้วยวิธีต่างๆ, CHAFE และอื่นๆอีกมาก อีกส่วนคือ **การพัฒนามาตรการตอบโต้การต่อต้านทางอิเล็กทรอนิกส์ (Electronic Counter Counter Measurement: ECCM)** กล่าวคือเป็นการพัฒนายุทธวิธีทางอิเล็กทรอนิกส์ที่ป้องกันเครื่องส่งของเรามิให้ข้าศึกรบกวน และป้องกันการค้นหาเป้าหมายของข้าศึกเป็นส่วนช่วยให้แถบแม่เหล็กไฟฟ้าของเรายังคงอยู่ได้ อาทิเช่น การใช้ระบบ Frequency hopping เพื่อหลบหลีก Jamming ของข้าศึก หรือการใช้ระบบ Communication security (COMSEC) ด้วยการเข้ารหัส (Encryption) เพื่อป้องกันการดักรับของข้าศึกเป็นต้น

ในวิทยานิพนธ์ฉบับนี้ ได้มุ่งเน้นไปที่การพัฒนามาตรการสนับสนุนอิเล็กทรอนิกส์ (ESM) โดยมุ่งประเด็นความสำคัญไปที่ Radio direction finding หรือ การประมาณหาที่ตั้งแหล่งกำเนิดสัญญาณวิทยุ ซึ่งการพัฒนางานทางด้านนี้ได้ดำเนินการพัฒนาอย่างต่อเนื่องมาโดยตลอด ได้แก่

Spingarn , K.(1987) [1] นำเสนอวิธีการประมาณหาที่ตั้งแหล่งกำเนิดสัญญาณ จากการวัดมุมทิศโดยใช้วิธี extended Kalman filter (EKF) ในกรณีของ moving observer ที่ติดตั้งบนอากาศยาน โดยการพัฒนาอัลกอริทึมจาก standard Kalman filter (SKF) ที่อยู่ในรูปของ state variables โดยที่ extended Kalman filter มี state variables เป็น nonlinear function

ร.อ.สุรเดช เคารพครู (1996) [2] นำเสนอวิธีการประมาณหาที่ตั้งแหล่งกำเนิดสัญญาณ จากการวัดมุมทิศ โดยใช้วิธี extended Kalman filter (EKF) ในกรณีของ multiple fixed observers โดยใช้หลักการเดียวกันกับ Spingarn

Torrieri , D.(1984) [3] นำเสนอวิธีการประมาณหาที่ตั้งแหล่งกำเนิดสัญญาณ จากการวัดมุมทิศ โดยนำอัลกอริทึมของ Stansfield [4] มาพัฒนาแล้วใช้ การหาค่าผิดพลาดที่น้อยที่สุดเชิงเส้น ในการประมาณหาที่ตั้งแหล่งกำเนิดสัญญาณ โดยเสนอว่า ในกรณีที่ ความแปรปรวนของสัญญาณรบกวน มีค่าน้อยมากๆ อัลกอริทึมของ Torrieri จะให้ผลที่แม่นยำกว่า อัลกอริทึมของ Stansfield

จากที่ได้กล่าวมาในข้างต้นนั้น จะสังเกตได้ว่า การประมาณแหล่งที่ตั้งกำเนิดสัญญาณนั้น จะนำไปใช้ในภารกิจทางทหารเสียเป็นส่วนใหญ่ ซึ่งปัญหาที่เกิดขึ้นนั้นยังคงเป็นปัญหาสำคัญมาโดยตลอด ได้แก่ ความแม่นยำในการพิสูจน์ทราบตำแหน่งของแหล่งกำเนิดสัญญาณ,ความรวดเร็วในการพิสูจน์ทราบตำแหน่งของแหล่งกำเนิดสัญญาณโดยแม่นยำ ซึ่งปัจจัยเหล่านี้สามารถเป็นตัวบ่งชี้ถึงชัยชนะได้ จากเหตุผลดังกล่าว ทางกองทัพไทยจึงได้ให้ความสนใจ กับการวิจัยด้านนี้อย่างจริงจัง เพื่อพัฒนาอาวุธยุทโธปกรณ์ ในกองทัพให้มีประสิทธิภาพ และทันสมัยต่อไป

ในการพัฒนาเพื่อเพิ่มประสิทธิภาพ อัลกอริทึมของ extended Kalman filter (EKF) นั้น CHARLES K. CHUI, GUANRONG CHEN, and HERMAN C. CHUI (1990) ได้นำเสนอวิธี modified extended Kalman filter (MEKF) [5] โดยนำไปใช้ในงานของ System parameter identification และได้ผลสรุปออกมาว่า MEKF นั้นสามารถให้ความแม่นยำมากกว่า วิธีของ EKF

ด้วยเหตุนี้เป็นสิ่งจูงใจให้ผู้วิจัย ทำการศึกษาและวิจัยในเรื่องของ การพัฒนาอัลกอริทึม โดยใช้วิธี modified extended Kalman filter (MEKF) เพื่อเพิ่มประสิทธิภาพให้กับ ระบบการประมาณหาที่ตั้งแหล่งกำเนิดสัญญาณจากการวัดมุมทิศ ทั้งในกรณี เครื่องดักรับหาทิศสัญญาณแบบเคลื่อนที่ และเครื่องดักรับหาทิศสัญญาณแบบอยู่กับที่ โดยเปรียบเทียบกับ ระบบการประมาณหาที่ตั้งแหล่งกำเนิดสัญญาณจากการวัดมุมทิศ โดยวิธีเดิมนั่นคือ extended Kalman filter (EKF) ในทั้ง 2 กรณี [1],[2]

<u>วัตถุประสงค์</u>

1. เพื่อพัฒนา อัลกอริทึมของวงจรกรองคาลมานแบบยืดขยาย (EKF) ให้มีความแน่นอน และแม่นยำมากยิ่งขึ้น
2. เพื่อพัฒนาระบบการหาที่ตั้งแหล่งกำเนิดสัญญาณ จากการวัดมุมทิศ ทั้งในกรณีเครื่องดักรับหาทิศสัญญาณเคลื่อนที่ และเครื่องดักรับหาทิศสัญญาณอยู่กับที่ ให้มีความแม่นยำมากยิ่งขึ้น
3. ศึกษา และเปรียบเทียบ อัลกอริทึมที่นำเสนอ กับอัลกอริทึมเดิม เพื่อนำผลมาเป็นแนวทางในการศึกษาและ พัฒนาวิธีการหาที่ตั้งแหล่งกำเนิดสัญญาณ โดยวิธีอื่นต่อไป

4. เสนอแนะแนวทางในการพัฒนา ระบบการประมาณหาที่ตั้งแหล่งกำเนิดสัญญาณ โดยแนะนำอัลกอริทึม และรูปแบบของระบบชนิดอื่นๆ ซึ่งยังมิได้มีการวิจัย

ขอบเขตของการวิจัย

1. พัฒนา และประยุกต์ กรรมวิธีแบบเดิมของการหาที่ตั้งแหล่งกำเนิดสัญญาณ จากการวัดมุมทิศ ด้วยกรรมวิธีแบบใหม่ ทั้งในกรณีเครื่องดักรับหาทิศสัญญาณเคลื่อนที่ และในกรณีเครื่องดักรับหาทิศสัญญาณอยู่กับที่
2. แสดงการเปรียบเทียบให้เห็นว่ากรรมวิธีใหม่ที่วิทยานิพนธ์ฉบับนี้ได้นำเสนอนั้น ให้ผลลัพธ์ที่แม่นยำกว่ากรรมวิธีแบบเดิม

วิธีดำเนินการวิจัย

1. ศึกษารายละเอียดเกี่ยวกับ การหาที่ตั้งแหล่งกำเนิดสัญญาณจากการวัดมุมทิศด้วยวิธี EKF
2. วิเคราะห์โปรแกรม การจำลองการหาที่ตั้งแหล่งกำเนิดสัญญาณด้วยวิธี EKF บนคอมพิวเตอร์
3. วิเคราะห์ข้อผิดพลาดที่เกิดขึ้นและ แนวทางการเพิ่มประสิทธิภาพของระบบ
4. ศึกษา ค้นคว้าอัลกอริทึมใหม่ ที่สามารถเพิ่มประสิทธิภาพของระบบและ แก้ไขข้อผิดพลาดดังกล่าวได้
5. รวบรวมข้อมูลและ บทความทั้งหมดที่เกี่ยวกับอัลกอริทึมใหม่นี้ เพื่อเป็นหลักฐานและ รูปแบบในการนำมาประยุกต์ใช้ในส่วนที่ต้องการเพิ่มประสิทธิภาพและ แก้ไขข้อผิดพลาดต่อไป
6. นำอัลกอริทึมใหม่ดังกล่าวมาประยุกต์เข้ากับ ระบบการหาที่ตั้งแหล่งกำเนิดสัญญาณแบบเดิม
7. ทดสอบระบบโดยการ ทดสอบปฏิบัติแบบจำลองบนคอมพิวเตอร์ ด้วยอัลกอริทึมใหม่ (ขั้นต้น)
8. พัฒนาและ ปรับปรุงการทำแบบจำลองบนคอมพิวเตอร์จนได้ผลเป็นที่น่าพอใจ
9. สรุปรวบรวมข้อมูลและ ผลการทดสอบแบบจำลองบนคอมพิวเตอร์ ทั้งหมดเพื่อเรียบเรียงเป็นเอกสารวิทยานิพนธ์

ประโยชน์ที่คาดว่าจะได้รับ

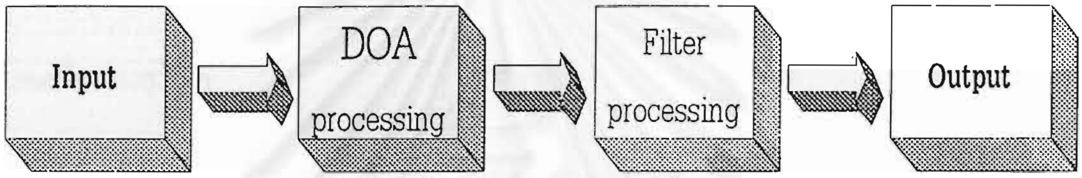
เป็นพัฒนาการอีกก้าวหนึ่งของระบบ การหาที่ตั้งแหล่งกำเนิดสัญญาณ ที่มีประสิทธิภาพในด้านความแม่นยำมากขึ้น ซึ่งเป็นไปตามนโยบายการพัฒนาการสงครามอิเล็กทรอนิกส์ ของกองทัพ โดยที่สามารถนำไปพัฒนาและ ปรับปรุงระบบที่ผ่านมาให้มีประสิทธิภาพสูงขึ้น และเป็นแนวทางในการพัฒนาระบบการหาที่ตั้งแหล่งกำเนิดสัญญาณ ด้วยวิธีอื่นๆ ต่อไป

บทที่ 2

การประมาณหาที่ตั้งแหล่งกำเนิดสัญญาณ

ระบบการประมาณหาที่ตั้งแหล่งกำเนิดสัญญาณ

ระบบการประมาณหาที่ตั้งแหล่งกำเนิดสัญญาณแบ่งออกเป็น 2 ส่วนหลักใหญ่ๆได้แก่ Direction of Arrival (DOA) processing ซึ่งคือการหาทิศทางการแพร่คลื่นแม่เหล็กไฟฟ้า ของแหล่งกำเนิดสัญญาณโดยประมาณ โดยที่ยังมิได้ผ่านการกรองละเอียดจากวงจรกรองชนิดต่างๆ และอีกส่วนคือ Filter processing คือการนำข้อมูลที่ได้จาก DOA processing มาผ่านการกรองละเอียด จากวงจรกรอง เพื่อให้ได้พิกัดของแหล่งกำเนิดสัญญาณที่แม่นยำมากยิ่งขึ้น ซึ่งสามารถแสดงเป็น block diagram ได้ดังรูปที่ 2.1 [2]



รูปที่ 2.1 ระบบการประมาณหาที่ตั้งแหล่งกำเนิดสัญญาณ

เนื่องจากในระหว่างการเดินทางของคลื่นวิทยุนั้น จะต้องผ่านสิ่งกีดขวางต่างๆ ไม่ว่าจะเป็นสิ่งปลูกสร้างของมนุษย์ หรือ ธรรมชาติสร้างขึ้นก็ตาม หรือแม้แต่ การหักเห สะท้อน ของคลื่นในชั้นบรรยากาศ ทำให้ค่าที่วัดได้ มีสัญญาณรบกวน (Noise) ปะปนอยู่ด้วย ดังนั้นค่าที่ได้จึงมีความคลาดเคลื่อนจากความเป็นจริง การแก้ไขปัญหาดังกล่าวนั้น ได้มีการค้นคว้า วิจัยและพัฒนาระบบอย่างต่อเนื่องมาโดยตลอด [1],[2],[3],[4] ซึ่งสามารถกระทำได้ในทั้ง 2 ขั้นตอน คือ ทั้งขั้นตอน DOA processing และขั้นตอน Filter processing

วิทยานิพนธ์ฉบับนี้ นำเสนอการแก้ไขปัญหาดังกล่าวซึ่งอยู่ในขั้นตอนของ Filter processing โดยใช้ Modified Extended Kalman filter (MEKF) เป็นกรรมวิธีที่ใช้ในการประมวลผล

รูปแบบของการประมาณหาที่ตั้งแหล่งกำเนิดสัญญาณ

อัลกอริทึมที่ใช้ในการแก้ปัญหาเกี่ยวกับ การประมาณหาที่ตั้งแหล่งกำเนิดสัญญาณนั้น มีหลากหลายวิธีแตกต่างกันไป ขึ้นอยู่กับรูปแบบเหตุการณ์ที่เกิดขึ้น ซึ่งโดยทั่วไปแล้วเราสามารถแบ่งรูปแบบของการประมาณหาที่ตั้งแหล่งกำเนิดสัญญาณ ตามลักษณะของการทำงานได้เป็น 2 ประเภทใหญ่ ๆ ดังนี้

หอสมุดกลาง สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

1. เครื่องดักรับหาทิศสัญญาณเคลื่อนที่ (Moving observer measurements)

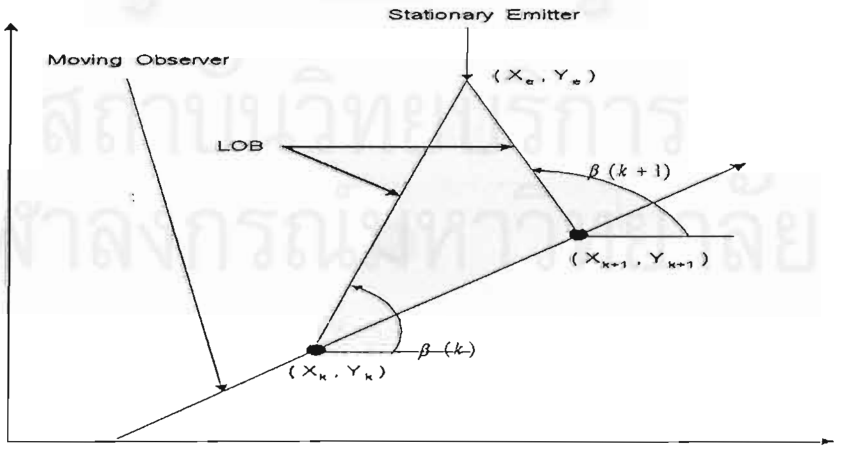
เป็นการประมาณหาที่ตั้งแหล่งกำเนิดสัญญาณรูปแบบหนึ่งโดยใช้ observer ติดตั้งบนอากาศยาน โดยต้องทำการวัดอย่างน้อยตั้งแต่ 2 ครั้งขึ้นไป ในขณะที่อากาศยานกำลังเคลื่อนที่ ซึ่งวิธีที่นิยมใช้สำหรับ ในกรณีนี้ ได้แก่

1.1 Frequency difference of arrival (FDOA) [6]

Frequency difference of arrival (FDOA) หรือเรียกอีกอย่างว่า Difference Dopple เป็นวิธีการวัดความต่างของความถี่ในช่วงเวลาเดียวกัน ของเครื่องดักรับหาทิศสัญญาณที่ติดตั้งบนอากาศยาน 2 เครื่อง ในขณะอากาศยานเคลื่อนที่ โดยในทางปฏิบัตินั้น เครื่องดักรับหาทิศสัญญาณต้อง Synchronized ได้แม่นยำในช่วงเวลาเดียวกัน และการวัดการเปลี่ยนแปลงตำแหน่งในระหว่างช่วงการวัดความถี่ จะต้องมีความเที่ยงตรงสูงมาก

1.2 Bearing angle measurements [8]

Bearing angle measurements คือ วิธีการประมาณหาที่ตั้งแหล่งกำเนิดสัญญาณจากการวัดมุมทิศ ที่ต้องใช้เครื่องดักรับหาทิศสัญญาณที่ได้ติดตั้งไว้บนอากาศยาน เพื่อหา Line of bearing (LOB) หรือ Direction of arrival (DOA) ของแต่ละเครื่องในขณะที่อากาศยานกำลังเคลื่อนที่ จากนั้นนำมา plot เพื่อที่จะหาจุดตัดจาก LOB ทั้งหมดที่มีในระบบ และจุดตัดที่ได้นั้นก็คือ ตำแหน่งที่ตั้งของแหล่งกำเนิดสัญญาณ ดังแสดงในรูปที่ 2.1



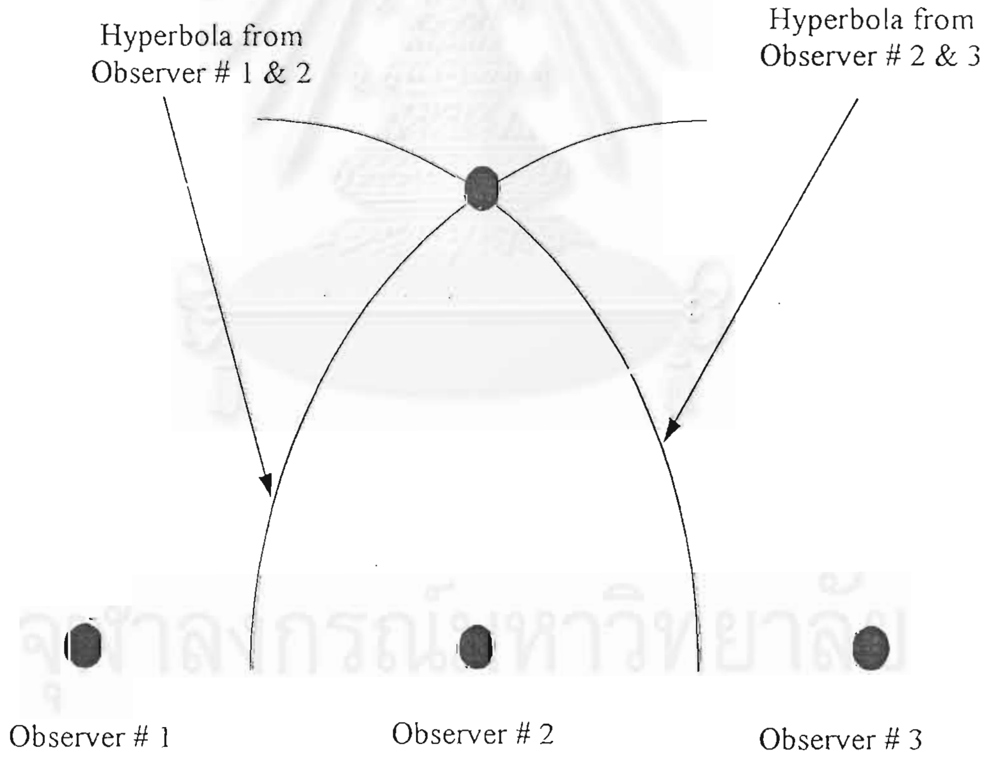
รูปที่ 2.2 Passive position location system geometry

2. เครื่องดักรับหาทิศสัญญาณอยู่กับที่ (Multiple fixed observers)

เป็นการประมาณการหาที่ตั้งแหล่งกำเนิดสัญญาณ โดยใช้เครื่องดักรับหาทิศสัญญาณ อย่างน้อย 2 เครื่องขึ้นไป โดยที่ เครื่องดักรับหาทิศสัญญาณที่ใช้ในการวัดนั้นตั้งอยู่กับที่ ซึ่งวิธีที่นิยมใช้ สำหรับการประมาณหาที่ตั้งแหล่งกำเนิดสัญญาณด้วยวิธีนี้ ได้แก่

2.1 Time difference of arrival (TDOA) [2],[7]

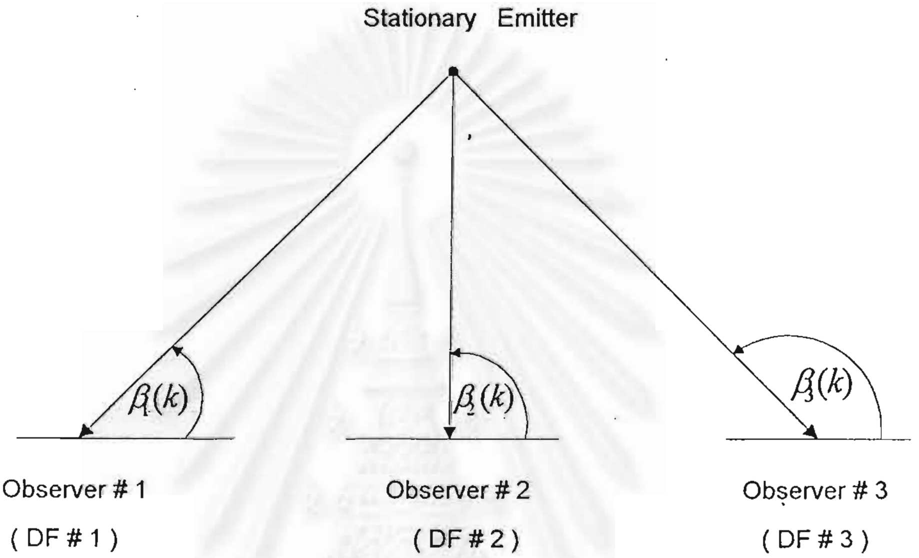
การประมาณหาที่ตั้งแหล่งกำเนิดสัญญาณแบบ Time difference of arrival (TDOA) หรือเรียกอีกอย่างหนึ่งว่า Hyperbolic localization จะหาตำแหน่งของ แหล่งกำเนิดสัญญาณโดยการดำเนินกรรมวิธีการวัดเวลาที่เข้ามาของสัญญาณ มายังสถานีเครื่องรับ จำนวนตั้งแต่ 3 สถานีขึ้นไปโดยใช้หลักการในการหาครั้งละ 2 สถานีเพื่อให้ได้เส้นโค้ง hyperbola 1 เส้น และต้องใช้จำนวนสถานีอย่างน้อย 3 สถานีขึ้นไป เพื่อให้เกิดจุดตัดระหว่างเส้นโค้ง hyperbola ทั้ง 2 เส้น ซึ่งจุดตัดที่ได้นั้นก็คือ ที่ตั้งของแหล่งกำเนิดสัญญาณ แสดงได้ดังรูปที่ 2.3



รูปที่ 2.3 Intersecting hyperbolas three observes

2.2 Bearing angle measurements [8]

Bearing angle measurements คือ วิธีการประมาณหาที่ตั้งแหล่งกำเนิดสัญญาณจากการวัดมุมทิศ ที่ต้องใช้เครื่องดักรับหาทิศสัญญาณ ที่ตั้งอยู่กับที่อย่างน้อย 2 เครื่องขึ้นไป เพื่อหา Line of bearing (LOB) หรือ Direction of arrival (DOA) ในแต่ละเครื่อง ซึ่งจะได้จุดตัดจาก LOB ทั้งหมดที่มีในระบบ และจุดตัดที่ได้นั้นก็คือ ตำแหน่งที่ตั้งของแหล่งกำเนิดสัญญาณ ดังแสดงในรูปที่ 2.4



รูปที่ 2.4 Stationary emitter localization

จากข้างต้นที่ได้กล่าวมานั้น ในทั้ง 2 กรณีคือทั้ง เครื่องดักรับหาทิศสัญญาณเคลื่อนที่ (moving observer) [1] และ เครื่องดักรับหาทิศสัญญาณอยู่กับที่นั้น (multiple fixed observers) [2] ได้เลือกใช้วิธีของ Bearing angle measurements ทั้งคู่เนื่องจาก วิธี Bearing angle measurements นั้นเป็นวิธีที่ให้ความแม่นยำ และความเชื่อถือได้ ได้มากกว่าวิธีอื่นๆ และที่สำคัญคือ วิธีนี้เป็นวิธีที่ค่อนข้างง่าย และไม่ซับซ้อนมากในทางปฏิบัติ

ดังนั้นในวิทยานิพนธ์ฉบับนี้ จึงได้ศึกษาวิธีการประมาณหาที่ตั้งแหล่งกำเนิดสัญญาณในทั้ง 2 กรณี โดยใช้วิธีการประมาณที่ตั้งแหล่งกำเนิดสัญญาณ จากการวัดมุมทิศ (Bearing angle measurements)

การประมาณหาที่ตั้งแหล่งกำเนิดสัญญาณ จากการวัดมุมทิศ

เราพิจารณาแยกออกเป็น 2 กรณีคือ

1. เครื่องดักรับหาทิศสัญญาณเคลื่อนที่ (Moving observer measurement)

ในกรณีเครื่องดักรับหาทิศสัญญาณเคลื่อนที่นั้น KARL SPINGARN [1] ได้รวบรวม สรุป และเปรียบเทียบ อัลกอริทึมไว้ 2 กรณีใหญ่ๆ ได้แก่

1.1 Nonlinear least square [1]

เป็นวิธีที่สามารถให้ความแม่นยำสูงวิธีหนึ่ง โดยการทำงานของระบบ จะเก็บค่าของข้อมูลที่ได้เข้ามาเอาไว้ก่อน (Bearing angle ; $\bar{\beta}(k)$) เพื่อทำการปรับค่า ให้ได้ค่าที่เหมาะสม ($\hat{\beta}(k)$) จากนั้นจึงทำการประมวลผลในขั้นตอนสุดท้ายเพื่อให้ได้ พิกัด $(\hat{x},\hat{y})$ ที่ต้องการ ซึ่งเราเรียกกรรมวิธีนี้ว่า Batch process [1] ซึ่งในทางปฏิบัติการจริงนั้นเราไม่นิยมใช้วิธีนี้ เนื่องจากเหตุการณ์จริงที่เกิดขึ้นนั้นเป็นระบบแห่งเวลาจริง หรือ real time นั่นคือในขั้นตอนของการประมวลผลนั้นจะถูกดำเนินการไปพร้อม ๆ กับข้อมูลที่ได้มา ณ. เวลานั้น ๆ

1.2 Extended Kalman filter (EKF) [1]

เป็นวิธีที่นิยมนำไปใช้แก้ปัญหาในทางปฏิบัติการจริง ของระบบที่มีข้อมูลไม่เป็นฟังก์ชันเชิงเส้น เนื่องจากระบบของอัลกอริทึมเป็นแบบ ประมวลผลในเวลาจริง นั่นคือสามารถประมวลผลได้ในทันทีที่ได้รับข้อมูลเข้ามา (Bearing angle ; $\beta_m(k)$) ในขณะนั้น ทำให้เราสามารถได้รับผลลัพธ์ ที่ใกล้เคียงความเป็นจริงมากที่สุด

ในวิทยานิพนธ์ฉบับนี้ ในกรณีเครื่องดักรับหาทิศสัญญาณเคลื่อนที่นั้น จะใช้ อัลกอริทึมของ Extended Kalman filter (EKF) ในการนำมาเปรียบเทียบกับ อัลกอริทึมที่จะนำเสนอ

2. เครื่องดักรับหาทิศสัญญาณอยู่กับที่ (Multiple fixed observers measurements)

ในเครื่องดักรับหาทิศสัญญาณอยู่กับที่นั้น ร.อ.สุรเดช เคารพครู [2] ได้รวบรวม สรุป และเปรียบเทียบ ไว้ 3 กรณีใหญ่ ๆ ได้แก่

2.1 Location estimation (Loc.Est.) from optimal bearing angle estimation [2]

เป็นการหาค่ามุมทิศที่เหมาะที่สุดของแต่ละ Observer ก่อนโดยการให้มุมทิศที่วัดได้จาก Observer แต่ละ Observer ผ่านขบวนการ Optimal bearing angle estimation ซึ่งก็คือ ขบวนการ Noise smoothing ซึ่งในขั้นตอนนี้ ในงานวิจัยของ ร.อ.สุรเดช เคารพครู นั้นได้นำเสนอขั้นตอนการประมาณค่าข้อมูลออปติมัลจากค่าวัดที่มีการรบกวน อยู่ 2 วิธี ได้แก่

- Maximum likelihood estimation (ML)
- Standard Kalman filtering (SKF)

และเมื่อได้ผ่านขั้นตอนนี้แล้วจะได้ Optimal angle แล้วจึงผ่านขบวนการ Location estimation (Loc.Est) ซึ่งมี Input เป็นมุมทิศ ซึ่งในขั้นตอน Loc.Est. นั้น ร.อ.สุรเดช เคารพครู ได้นำเสนออยู่ 4 วิธี ได้แก่

- Simple average method
- Least square method
- Stansfield's algorithm method
- Torrieri's algorithm method

ซึ่งเมื่อผ่านขั้นตอน Loc.Est. แล้วจะได้ Output ออกมาอยู่ในรูปพิกัด $(\hat{x},\hat{y})$ ซึ่งเป็นค่าที่ Optimal

2.2 Optimal location estimation from location estimation (Loc.Est.) [3]

เป็นการหาตำแหน่งของแหล่งกำเนิดสัญญาณ (Emitter location) จาก Measured bearing angle ก่อนโดยให้ Measured bearing angle vector ผ่านเข้าขบวนการ Loc.Est. (รายละเอียดตามข้อ 2.1) จะได้ Estimation location ออกมา แล้วนำมาผ่านขบวนการ Optimal Loc.Est. หรือว่า Noise smoothing (รายละเอียดตามข้อ 2.1) และท้ายที่สุดจะได้ Output ออกมาอยู่ในรูปของพิกัด $(\hat{x},\hat{y})$ ซึ่งเป็นค่าที่ เหมาะสมที่สุด

2.3 Extended Kalman filter (EKF)

เป็นกรรมวิธีเดียวกันกับ EKF ในหัวข้อ 1.2 นั่นคือเป็นระบบแบบ ประมวลผลในเวลาจริงจะทำการประมวลในทันทีที่ Input เข้ามา แต่จะแตกต่างกันตรงที่ จำนวนของ Direction finders (DF) ,ห้วงเวลา (Timing) ในการดักรับหาทิศทางของสัญญาณ

ในวิทยานิพนธ์ฉบับนี้ในกรณีเครื่องดักรับหาทิศสัญญาณอยู่กับที่นั้นจะใช้อัลกอริทึมของ Standard Kalman filter (SKF) + Torrieri's algorithm (TR) หรือ (SKF + TR) และ Extended Kalman filter (EKF) ในการนำมาเปรียบเทียบกับ อัลกอริทึมที่จะนำเสนอ (MEKF)

<u>อัลกอริทึม สำหรับการประมาณหาที่ตั้งแหล่งกำเนิดสัญญาณจากการวัดมุมทิศ ที่นำมาใช้ในวิทยานิพนธ์ฉบับนี้</u>

1. <u>Standard Kalman filter (SKF)</u> [1],[2],[9]

ในอัลกอริทึมของ SKF นี้จะอยู่ในรูปแบบของ Discrete filter หรือเป็นวิธีการแก้ปัญหาของข้อมูลที่เป็น Discrete data linear filtering โดยการประมาณหาค่าของสถานะใด ๆ ที่ต้องการ $\hat{X}(k/k)$ จากการวัดค่าสัญญาณรบกวน $v(k)$ ซึ่งเป็นการทำนายค่าที่เวลา (k) จากข้อมูล ณ.เวลา $(k-1)$ นั่นหมายความว่า ถ้าต้องการหาค่า $\hat{X}(k/k)$ เราจำเป็นต้องทราบค่า $\hat{X}(k-1/k-1)$, $\hat{X}(k/k-1)$ และ Covariance $P(k-1/k-1)$, $P(k/k-1)$ รวมทั้ง Transition matrix ด้วย โดยมีรูปแบบ และรายละเอียดดังนี้ [1]

The extrapolation of the state estimate

$$\hat{X}(k/k-1) = \Phi\,\hat{X}(k-1/k-1) \tag{2.1}$$

โดยที่

Φ = Transition matrix

$\hat{X}(k-1/k-1)$ = The smoothed position estimate

$\hat{X}(k/k-1)$ = The prediction position estimate

The error covariance extrapolation

$$P(k/k-1) = \Phi\ P(k-1/k-1)\ \Phi^T + Q \tag{2.2}$$

โดยที่

Q = The process noise defined a small positive number

The bearing angle measurement model

$$\beta(k) = h(\hat{X}(k)) + \nu(k) \tag{2.3}$$

โดยที่

$h(\hat{X}(k))$ = nonlinear function ของ ตำแหน่งของแหล่งกำเนิดสัญญาณ

$\nu(k)$ = Zero mean white Gaussion noise sequence

Kalman gain

$$K = P(k/k-1)h^T(\hat{X}(k/k-1))[h(\hat{X}(k/k-1))P(k/k-1) \times h^T(\hat{X}(k/k-1)) + R(k)]^{-1} \tag{2.4}$$

โดยที่

R = ค่าความแปรปรวนของสัญญาณรบกวน $\nu(k)$

The state estimate equation

$$\hat{X}(k/k) = \hat{X}(k/k-1) + K[\beta(k) - h(\hat{X}(k/k-1))] \tag{2.5}$$

โดยที่

$\hat{\beta}(k) = h(\hat{X}(k/k-1))$ = มุมที่ประมาณได้จาก state ที่แล้ว

$\beta(k) = h(\hat{X}(k))$ = มุมที่เพิ่งวัดได้

The error covariance update

$$P(k/k) = [I - Kh(\hat{X}(k/k-1))]P(k/k-1) \tag{2.6}$$

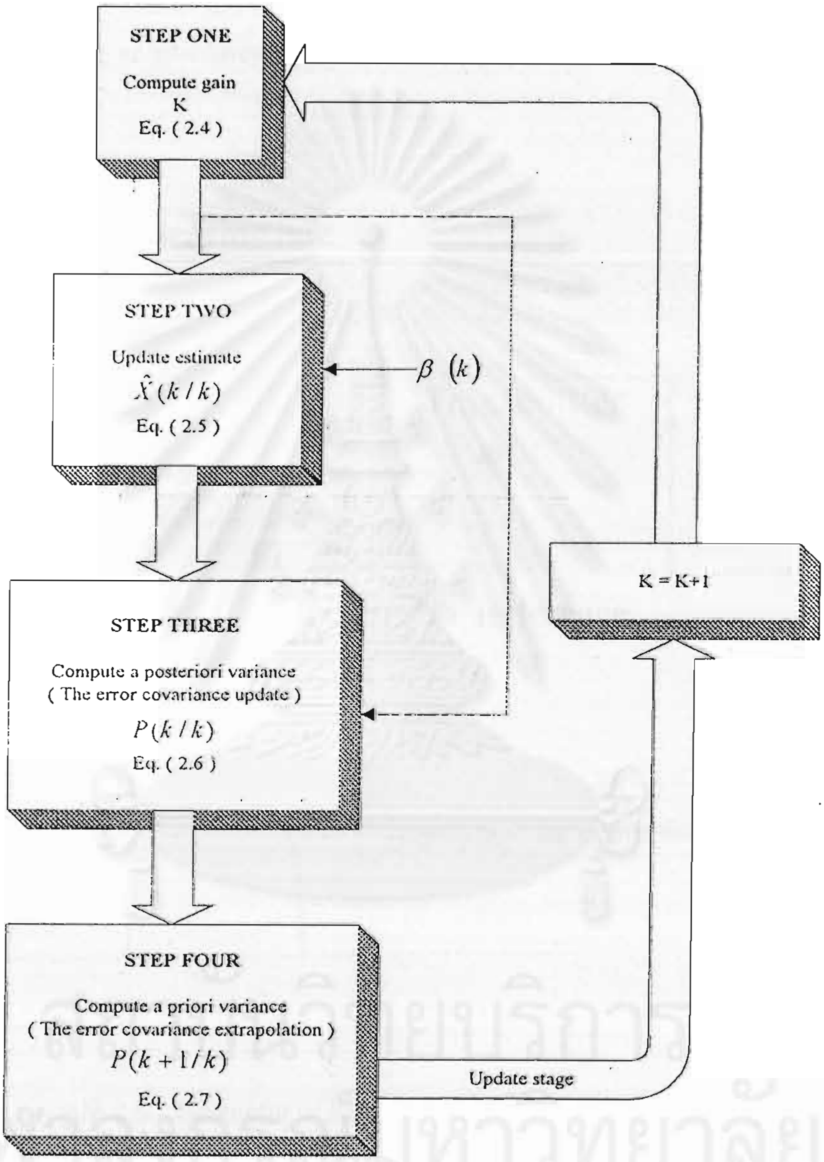
โดยที่

I = Identity matrix

The error covariance extrapolation (next state)

$$P(k+1/k) = \Phi\, P(k/k)\ \Phi^T + Q \tag{2.7}$$

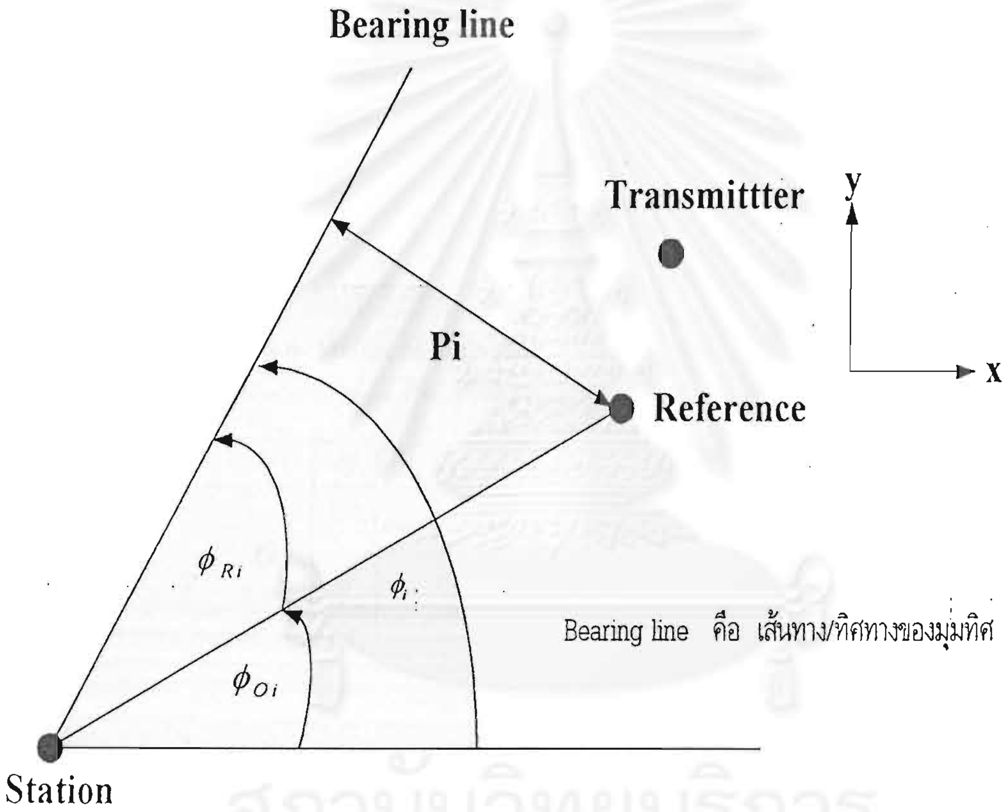
จากสมการที่ (2.1) - (2.7) ข้างต้นสามารถสรุปการทำงานเป็น Flowchart ได้ดังนี้ [9] สมการที่ (2.1) - (2.3) เป็นขั้นตอนการกำหนดค่าเริ่มต้นก่อนจะเข้าทำการประมวลผล



รูปที่ 2.5 Flowchart for Kalman filter algorithm

2. Torrieri's algorithm [2],[3]

Torrieri ได้นำเสนออัลกอริทึมของ วิธีการประมาณหาที่ตั้งแหล่งกำเนิดสัญญาณ จากการวัดมุมทิศ โดยนำอัลกอริทึมของ Stansfield [4] มาเป็นต้นแบบในการพัฒนา แล้วใช้ Linearized least square ในการประมาณหาที่ตั้งแหล่งกำเนิดสัญญาณ ในสามเหลี่ยมของการคลาดเคลื่อน Torrieri อ้างว่า อัลกอริทึมของเขาให้ความแม่นยำมากกว่าวิธีของ Stansfield ในกรณีที่ Variance มีค่าน้อยมากๆ อัลกอริทึมของ Torrieri จะให้ผลลัพธ์แม่นยำกว่าอัลกอริทึมของ Stansfield สามารถสรุปเป็นสมการดังนี้



รูปที่ 2.6 Geometry of Emitter, จุดอ้างอิง, and station

ข้อมูลที่ได้จากในการวัดทั้งหมด N ครั้งของ $r_i, i = 1,2,...,N$ จะถูกเก็บไว้ในตำแหน่งต่างๆ ซึ่งเราสามารถเขียนสมการ N - Dimensions ให้เป็นสมการเดียว ได้ดังนี้

$$r = f(x) + n \tag{2.8}$$

โดยที่

x = Position coordinates

$\bar{n}$ = Multivariate random vector โดยมี covariance matrix ขนาด $N \times N$

Covariance ของ $\bar{n}$ สามารถหาได้จาก

$$N = E\{(n - E[n])(n - E[n])^T\} \tag{2.9}$$

โดยทั่วๆไปแล้ว ค่าของ $f(x)$ นั้นจะเป็น nonlinear vector ฟังก์ชัน ซึ่งเราสามารถ approximate ให้เป็น Linear ได้โดยการใช้ Taylor's series ในการกระจายแล้วใช้ 2 พจน์แรก เราจะได้สมการ

$$f(x) \approx f(x_0) + G(x - x_0) \tag{2.10}$$

โดยที่

x_0 = ตำแหน่งของจุดอ้างอิง

G = $N \times n$ Matrix of derivatives evaluated at x_0

$$G = \begin{bmatrix} \left.\frac{\partial f_1}{\partial x_1}\right|_{x=x_0} & \cdots & \left.\frac{\partial f_1}{\partial x_n}\right|_{x=x_0} \\ \vdots & \ddots & \vdots \\ \left.\frac{\partial f_N}{\partial x_1}\right|_{x=x_0} & \cdots & \left.\frac{\partial f_N}{\partial x_n}\right|_{x=x_0} \end{bmatrix} \tag{2.11}$$

ดังนั้นค่าประมาณของตำแหน่งแหล่งกำเนิดสัญญาณ สามารถเขียนเป็นสมการได้

$$\hat{x} = x_0 + (G^T N^{-1} G)^{-1} G^T N^{-1} [r - f(x_0)] \tag{2.12}$$

เราแทนค่าสมการที่ (2.8) ลงในสมการที่ (2.12) จะได้

$$\hat{x} = x + (G^T N^{-1} G)^{-1} G^T N^{-1} [f(x) - f(x_0) - G(x - x_0) + n] \tag{2.13}$$

โดยมี Covariance คือ

$$p = E[(\hat{x} - E[\hat{x}])(\hat{x} - E[\hat{x}])^T] = (G^T N^{-1} G)^{-1} \tag{2.14}$$

จากสมการที่ (2.11) เราแทน $\boldsymbol{x} = \boldsymbol{R}$ และ $\boldsymbol{x}_0 = \boldsymbol{R}_0$

โดยที่ $\boldsymbol{R}$ เป็นเวกเตอร์ 2 มิติ ที่ได้จากการประมาณตำแหน่งประกอบด้วย x และ y

$\boldsymbol{R}_0$ เป็นเวกเตอร์ 2 มิติ ของ จุดอ้างอิง ประกอบด้วย x_0 และ y_0

จะได้

$$G = \begin{bmatrix} -(\sin\phi_{01})/D_{01} & (\cos\phi_{01})/D_{01} \\ \vdots & \vdots \\ -(\sin\phi_{0N})/D_{0N} & (\cos\phi_{0N})/D_{0N} \end{bmatrix} \tag{2.15}$$

โดยที่

$$D_{0i} = \sqrt{(x_0 - x_i)^2 + (y_0 - y_i)^2} \quad ; \quad i = 1,2,\ldots,N \tag{2.16}$$

ϕ_{0i} = มุมทิศระหว่างสถานีที่ 1 และ จุดอ้างอิง

จากสมการที่ (2.9) เราให้ความผิดพลาดในการวัดมุมแต่ละครั้งเป็นอิสระต่อกันจะได้

$$N = \begin{bmatrix} \sigma_{\phi 1}^2 & & 0 \\ & \ddots & \\ 0 & & \sigma_{\phi N}^2 \end{bmatrix} \tag{2.17}$$

โดยที่ $\sigma_{\phi i}^2$ คือ Variance ในการวัดครั้งที่ i เมื่อ $i = 1,2,\ldots,N$

สมการของ Least square หรือ maximum likelihood estimator คือ

$$\hat{R} = R_0 + (G^T N^{-1} G)^{-1} G^T N^{-1} \phi_r \tag{2.18}$$

โดยที่

$\phi_r = \phi - f(R_0)$ และ $\phi = f(R) + \boldsymbol{n}$

จากสมการที่ (2.14),(2.15)และ (2.17) เราจะได้ Covariance matrix ของ $\hat{R}$ คือ

$$\sigma_1^2 = E[(\hat{x}-x)^2] = \frac{\mu}{\mu\lambda - \upsilon^2} \tag{2.19}$$

$$\sigma_2^2 = E[(\hat{y}-y)^2] = \frac{\lambda}{\mu\lambda - \upsilon^2} \tag{2.20}$$

$$\sigma_{12} = E[(\hat{x}-x)(\hat{y}-y)] = \frac{\upsilon}{\mu\lambda - \upsilon^2} \tag{2.21}$$

ดังนั้นจากสมการที่ (2.15), (2.17), และ (2.18) เราจะได้สมการ Linearized least square estimator คือ

$$\hat{x} = x_0 + \frac{1}{\mu\lambda - \upsilon^2}\sum_{i=1}^{N}\phi_{ri}\frac{(\upsilon\cos\phi_{0i} - \mu\sin\phi_{0i})}{D_{0i}\sigma_{\phi_i}^2} \tag{2.22}$$

$$\hat{y} = y_0 + \frac{1}{\mu\lambda - \upsilon^2}\sum_{i=1}^{N}\phi_{ri}\frac{(\lambda\cos\phi_{0i} - \upsilon\sin\phi_{0i})}{D_{0i}\sigma_{\phi_i}^2} \tag{2.23}$$

โดยที่

$$\mu = \sum_{i=1}^{N}\frac{\cos^2\phi_{0i}}{D_{0i}^2\sigma_{\phi_i}^2} \tag{2.24}$$

$$\lambda = \sum_{i=1}^{N}\frac{\sin^2\phi_{0i}}{D_{0i}^2\sigma_{\phi_i}^2} \tag{2.25}$$

$$\upsilon = \sum_{i=1}^{N}\frac{\sin\phi_{0i}\cos\phi_{0i}}{D_{0i}^2\sigma_{0i}^2} \tag{2.26}$$

$$\phi_{ri} = \phi_i - \phi_{0i} \quad ; \phi_i \text{ คือ Bearing angle} \tag{2.27}$$

โดยการใช้เงื่อนไขของ Torrieri ดังต่อไปนี้คือ

กำหนดให้ p_i = ระยะทางที่สั้นที่สุดจาก จุดอ้างอิง ถึง Bearing line ของสถานี i ซึ่งถ้า จุดอ้างอิง ใกล้ แหล่งกำเนิดสัญญาณมากแล้ว จะได้ว่า

$$\phi_{ri} = p_i / D_{0i} \quad ; i = 1,2,\ldots,N \quad \text{และ}$$

$$\cos\phi_{0i} = \cos\phi_i \quad ; \quad \sin\phi_{0i} = \sin\phi_i \quad ; i = 1,2,\ldots,N$$

3. Extended Kalman filter (EKF) [1],[2],[5]

อัลกอริทึมของ Extended Kalman filter (EKF) ถูกนำมาประยุกต์ใช้ในระบบ nonlinear system ต่างๆ มากมาย เช่น System parameter identification [5], การประมาณหาที่ตั้งแหล่งกำเนิดสัญญาณ [1],[2] เป็นต้น ซึ่งในที่นี้จะกล่าวถึงเฉพาะงานในส่วนของ การประมาณแหล่งที่ตั้งกำเนิดสัญญาณ

ในงานด้านการประมาณหาที่ตั้งแหล่งกำเนิดสัญญาณนั้น K.SPINGARN [1] ได้พัฒนาและประยุกต์อัลกอริทึม EKF นี้ โดยนำมาใช้ในกรณีของเครื่องดักรับหาทิศสัญญาณแบบเคลื่อนที่ โดยใช้พื้นฐานจาก Standard Kalman filter (SKF) และในส่วนของกรณีของเครื่องดักรับหาทิศสัญญาณแบบอยู่กับที่นั้น ร.อ.สุรเดช เคารพครู [2] ได้นำอัลกอริทึม EKF นี้ ไปประยุกต์ใช้ โดยใช้พื้นฐานจาก SKF เช่นเดียวกัน ซึ่งในทั้ง 2 กรณีนั้น สามารถสรุปได้ว่าอัลกอริทึมของ EKF นั้น ให้ผลลัพธ์ได้แม่นยำ หรือใกล้เคียงตำแหน่งของแหล่งกำเนิดสัญญาณ ได้ มากกว่า อัลกอริทึมของ SKF

ทั้งนี้เนื่องจากข้อมูลที่เข้ามาในระบบนี้ไม่เป็นระบบฟังก์ชันเชิงเส้น ดังนั้นอัลกอริทึมแบบ SKF จึงไม่สามารถให้ผลลัพธ์ได้ดีเท่าที่ควร ดังนั้นจึงต้องมีการขยายในส่วนที่ไม่เป็นฟังก์ชันเชิงเส้น โดยการใช้ Taylor's series ในการขยาย ซึ่งในหลักการดังกล่าวนี้ เราเรียกว่า Extended Kalman filter (EKF)

เราสามารถสรุปขั้นตอนการประมวลผลของ EKF ได้ดังนี้ [1],[2]

The extrapolation of the state estimate

$$\hat{X}(k/k-1) = \Phi\ \hat{X}(k-1/k-1) \tag{2.28}$$

โดยที่

Φ = Transition matrix

$\hat{X}(k-1/k-1)$ = The smoothed position estimate

$\hat{X}(k/k-1)$ = The prediction position estimate

The error covariance extrapolation

$$P(k/k-1) = \Phi\ P(k-1/k-1)\ \Phi^T + Q \tag{2.29}$$

โดยที่

Q = The process noise defined a small positive number

The bearing angle measurement model

$$\beta(k) = h(\hat{X}(k)) + \nu(k) \tag{2.30}$$

โดยที่

$h(\hat{X}(k))$ = nonlinear function ของ ตำแหน่งของแหล่งกำเนิดสัญญาณ

$\nu(k)$ = Zero mean white Gaussion noise sequence

Linearization (Jacobian matrix)

$$\hat{\beta}(k) = h(\hat{x}(k/k-1)) = \tan^{-1}\left[\frac{(\hat{y}_e(k/k-1) - y(k))}{(\hat{x}_e(k/k-1) - x(k))}\right] \tag{2.31}$$

โดยที่

$\hat{x}_e(k/k-1), \hat{y}_e(k/k-1)$ = ตำแหน่งของแหล่งกำเนิดสัญญาณที่ประมาณค่าได้ ณ. เวลา k

$x(k), y(k)$ = ตำแหน่งของ Observer ณ. เวลา k

เรานิยาม $H(\hat{x}(k/k-1))$ เป็น Jacobian matrix ดังนี้

$$\begin{aligned} H(\hat{x}(k/k-1)) &= \left.\frac{\partial h(x)}{\partial x}\right|_{x=\hat{x}(k/k-1)} \\ &= [H_{11} \quad H_{12}] \end{aligned} \tag{2.32}$$

$$H_{11} = \left.\frac{\partial h}{\partial x_e}\right|_{x_e=\hat{x}_e(k/k-1)} \tag{2.33}$$

$$H_{12} = \left.\frac{\partial h}{\partial y_e}\right|_{y_e=\hat{y}_e(k/k-1)} \tag{2.34}$$

เราให้ $h = \tan^{-1}[u]$ (2.35)

โดยที่ $u = \left(\frac{\hat{y}_e(k/k-1) - y(k)}{\hat{x}_e(k/k-1) - x(k)}\right)$ (2.36)

ดังนั้นในสมการที่ (2.33) จะได้

$$H_{11} = \frac{-u}{(1+u^2)(\hat{x}_e(k/k-1) - x(k))} \tag{2.37}$$

และในสมการที่ (2.34) ได้

$$H_{12} = \frac{1}{(1+u^2)(\hat{x}_e(k/k-1) - x(k))} \tag{2.38}$$

Kalman gain

$$K = P(k/k-1)H^T(\hat{X}(k/k-1))[H(\hat{X}(k/k-1))P(k/k-1) \times H^T(\hat{X}(k/k-1)) + R(k)]^{-1} \tag{2.39}$$

โดยที่

R = ค่าความแปรปรวนของสัญญาณรบกวน $\nu(k)$

The state estimate equation

$$\hat{X}(k/k) = \hat{X}(k/k-1) + K[\beta(k) - H(\hat{X}(k/k-1))] \tag{2.40}$$

โดยที่

$\hat{\beta}(k) = H(\hat{X}(k/k-1))$ = มุมที่ประมาณได้จาก State ที่แล้ว

$\beta(k) = H(\hat{X}(k))$ = มุมที่เพิ่งวัดได้

The error covariance update

$$P(k/k) = [I - KH(\hat{X}(k/k-1))]P(k/k-1) \tag{2.41}$$

โดยที่

I = Identity matrix

The error covariance extrapolation (next state)

$$P(k+1/k) = \Phi\, P(k+1/k)\, \Phi^T + Q \tag{2.42}$$

อัลกอริทึมที่นำเสนอในวิทยานิพนธ์ (Modified extended Kalman filter : MEKF)

จากผลการวิจัยของ CHARLES K. CHUI, GUANRONG CHEN, HERMAN C. CHUI เรื่อง *"MEKF and a real-time Parallel Algorithm for System Parameter Identification"* [5] ดังที่ได้กล่าวไว้แล้วในข้างต้นนั้น สรุปได้ว่า ในเรื่องของ System Parameter Identification นั้นวิธีการของ MEKF นั้นให้ผลที่แม่นยำกว่าวิธีการของ EKF

ดังนั้นในวิทยานิพนธ์ฉบับนี้ จะนำเสนอวิธี ***Modified Extended Kalman Filter (MEKF)*** เพื่อนำมาปรับปรุงและพัฒนาในเรื่องของ การหาที่ตั้งแหล่งกำเนิดสัญญาณจากการวัดมุมทิศ ด้วยวิธี EKF ให้มีประสิทธิภาพและ ความแม่นยำมากยิ่งขึ้น โดยนำมาปรับปรุงทั้งใน 2 กรณี นั่นคือทั้งในกรณีเครื่องดักรับหาทิศสัญญาณเคลื่อนที่ และ เครื่องดักรับหาทิศสัญญาณอยู่กับที่ ซึ่งในกรณีของเครื่องดักรับหาทิศสัญญาณอยู่กับที่นั้น ในงานวิจัยของ ร.อ.สุรเดช เคารพครู [2] นั้นได้แสดงให้เห็นว่า วิธี EKF เป็นวิธีที่ให้ความแม่นยำมากกว่าวิธีอื่นๆ ซึ่งจากผลการทดสอบจะเห็นได้ว่าเป็นไปตามนั้นจริง และจากผลการทดสอบนั้นเองจะสังเกตเห็นได้ว่ายังมีอีก 1 วิธีที่ให้ความแม่นยำใกล้เคียงกัน นั่นคือวิธีของ Location Estimation from Optimal Bearing Angle Estimation [2] โดยที่ ในขั้นตอนของ Optimal Bearing Angle Estimation นั้นใช้วิธีของ Kalman Filter (KF) และ ในขั้นตอนของ Location Estimation ใช้วิธีของ Torrieri's Algorithm (TR) หรือกล่าวง่ายๆก็คือการใช้ Kalman Filter + Torrieri's Algorithm (KF + TR) นั่นเอง

ดังนั้นในวิทยานิพนธ์ฉบับนี้เราจึงได้พิจารณาถึงความสำคัญและ ความแม่นยำของวิธีนี้ และได้นำมาประยุกต์ขึ้นเป็นวิธีใหม่อีก 1 วิธี นั่นคือ ***Modified Kalman Filter (MKF) + Torrieri's Algorithm (TR) หรือ (MKF + TR)*** กล่าวคือ เราจะนำขั้นตอน KF ซึ่งเป็นขั้นตอนของการ Optimal Bearing Angle Estimation มา Modify ในลักษณะเดียวกันกับ EKF ดังที่ได้อธิบายมาแล้วข้างต้น จากนั้นจึงนำมาผ่าน Torrieri's Algorithm ซึ่งเป็นกระบวนการของ Location Estimation เพื่อประมาณหาที่ตั้งแหล่งกำเนิดสัญญาณต่อไป ซึ่งในวิทยานิพนธ์ฉบับนี้เราได้ใช้ Monte Carlo Simulation ในการทำ Numerical Simulation เพื่อแสดงผลการทดสอบและ ข้อเปรียบเทียบในทุกกรณีให้เห็นบนคอมพิวเตอร์ โดยแสดงผลลัพธ์ในรูปของ Square Location Error และ Circular Error Probability (CEP) และได้นำมา plot เป็นกราฟเปรียบเทียบในทุกกรณี ซึ่งอยู่ในรูปแบบของ Mean Square Error (MSE) และ Circular Error Probability (CEP)

หอสมุดกลาง สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

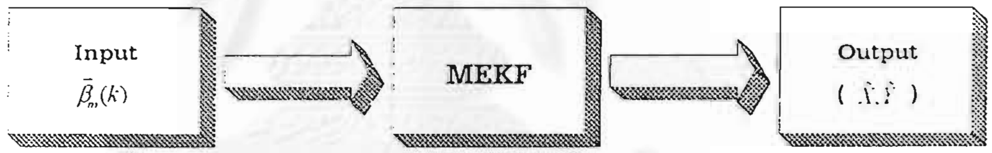
สรุปทั้ง 3 วิธีที่นำเสนอในวิทยานิพนธ์ฉบับนี้สามารถเขียน Block Diagram ได้ดังนี้

1. ***เครื่องดักรับสัญญาณเคลื่อนที่ (Moving observer measurements)***

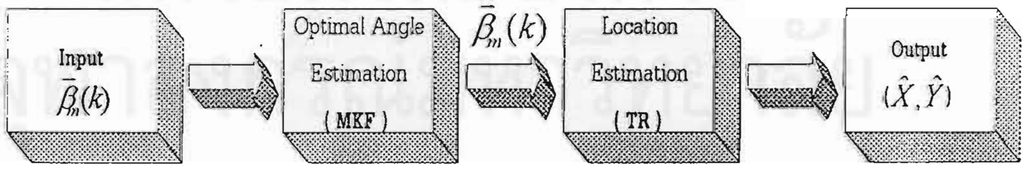
1.1) Modified extended Kalman filter (MEKF)

2. ***เครื่องดักรับสัญญาณอยู่กับที่ (Multiple fixed observers measurements)***

2.1) Modified extended Kalman filter (MEKF)

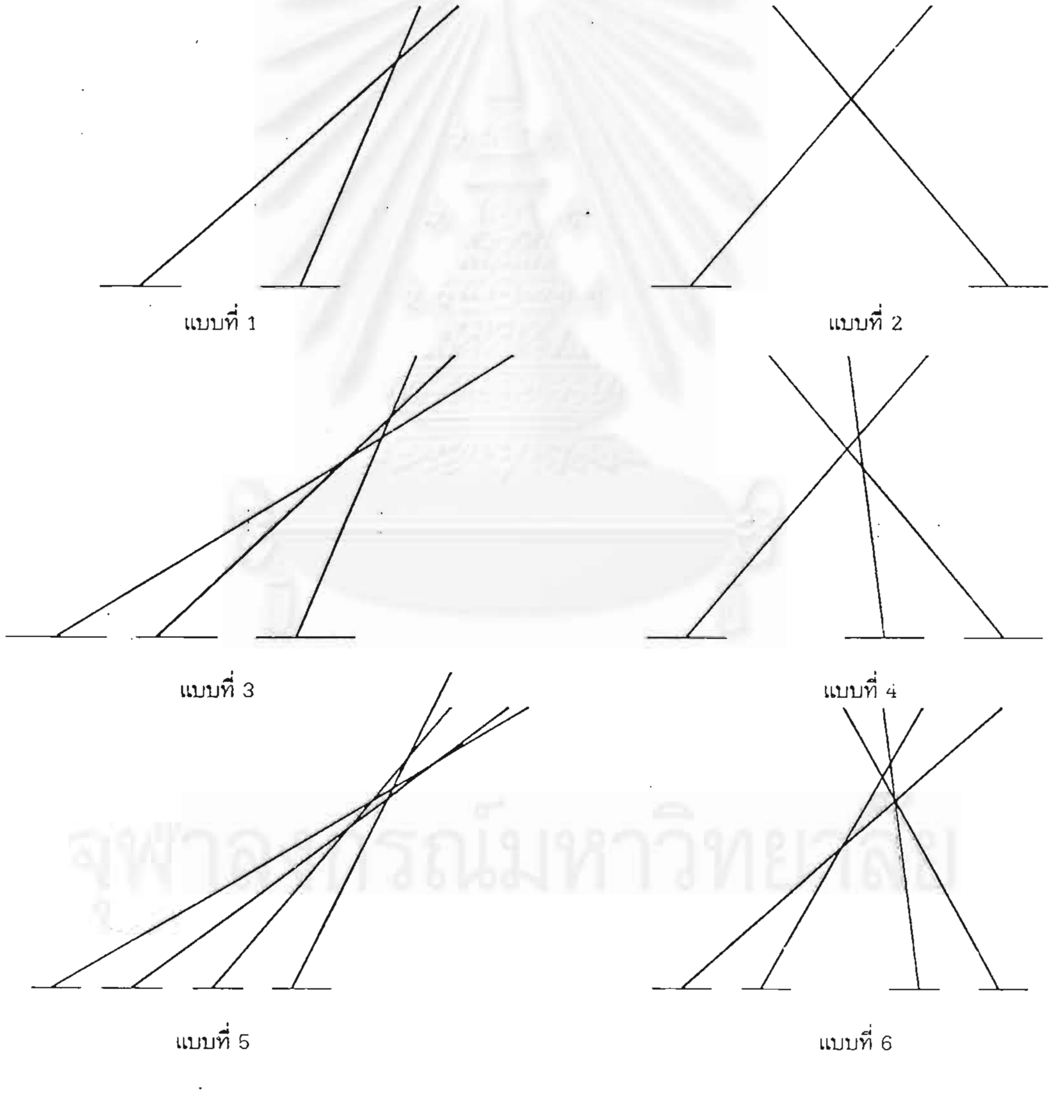


2.2) Modified Kalman filter (MKF) + Torrieri's algorithm (TR) (MKF + TR)



การดำเนินการทดสอบของทั้ง 2 กรณีในวิทยานิพนธ์ฉบับนี้นั้น จะเป็นการดำเนินการในรูปแบบตามแนวทางการทดสอบของงานวิจัยเดิม ในแต่กรณีนั่นคือ ในกรณีแรกซึ่งเป็นกรณีของ Moving observer measurements นั้น จะดำเนินการพัฒนางานในวิทยานิพนธ์ฉบับนี้ตามแนวทางการทดสอบของ KARL SPINGARN [1] กล่าวคือ จะมีการเปรียบเทียบการใช้ observer อยู่ 2 แบบ แบบแรกจะใช้ observer 2 ตัว และในแบบที่ 2 จะใช้ observer 9 ตัวในการดักรับหาทิศสัญญาณ

ส่วนในกรณีที่ 2 ที่เป็นกรณีของ Multiple fixed observers measurements นั้นในวิทยานิพนธ์ฉบับนี้จะดำเนินการพัฒนางานตามแนวทางการทดสอบของ ร.อ.สุรเดช เคารพครู [2] กล่าวคือ จะมีการเปรียบเทียบการใช้ Observer อยู่ 6 แบบดังนี้



รูปที่ 2.7 รูปแบบของ Observers แบบต่างๆ ที่ใช้ในการทดสอบ

จุฬาลงกรณ์มหาวิทยาลัย

รายละเอียดอัลกอริทึม Modified extended Kalman filter (MEKF) ที่นำเสนอ

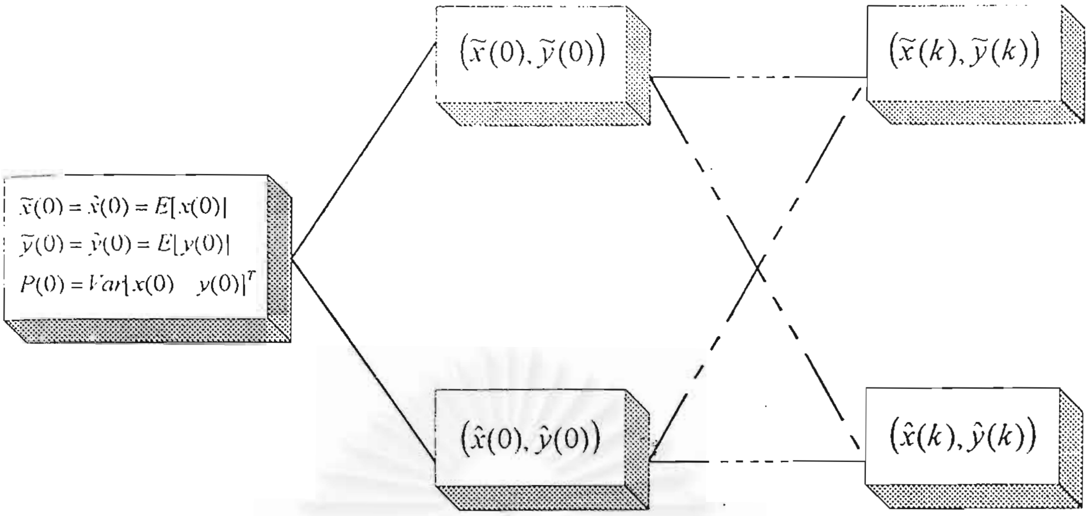
อัลกอริทึม Modified extended Kalman filter (MEKF) ที่จะกล่าวถึงนี้ เป็นการพัฒนาในส่วนของกระบวนการ linearization เพื่อให้เกิดประสิทธิภาพที่ดียิ่งขึ้น โดยในที่นี้จะนำเสนอวิธีการของ *Parallel computational scheme* ซึ่งสามารถใช้ประมาณค่าหรือ หาพารามิเตอร์ ที่เป็นแบบจำลองใน realtime ที่เราไม่สามารถทราบค่าได้ ได้ดีกว่า Extended Kalman filter (EKF) ในงานวิจัยของ CHARLES K. CHUI, GUANRONG CHEN, HERMAN C. CHUI เรื่อง " *MEKF and a Real-Time Parallel Algorithm for System Parameter Identification* " [5] นั้น ได้นำกรรมวิธีการประมวลผลแบบคู่ขนานมาปรับปรุงและ พัฒนาในส่วนของ EKF จากนั้นจึงได้นำไปประยุกต์ใช้ในเรื่องของ System parameter identification โดยมีการทดสอบและ แสดงผลการทดสอบด้วยการใช้ Numerical simulation ซึ่งผลที่ได้ปรากฏว่า วิธีการของ MEKF นั้นให้ผลที่ใกล้เคียงค่าความเป็นจริง มากกว่าวิธีการของ EKF

Modified extended Kalman filter (MEKF) [5]

จากแนวความคิดของ CHARLES K. CHUI, GUANRONG CHEN, HERMAN C. CHUI,[1] คือการพัฒนาอัลกอริทึมของ EKF โดยการใช้ Standard Kalman filter (SKF) เข้ามาเพิ่มในระบบ โดยให้ทั้ง 2 อัลกอริทึมนั้น ทำงานขนานกันไปซึ่งเรียกวิธีนี้ว่า กรรมวิธีประมวลผลแบบคู่ขนาน

ในกระบวนการของ EKF นั้นแนวความคิดหลักคือ การทำในส่วนที่เป็น nonlinear ให้เป็น linear โดยการใช้ Taylor's series ในการกระจาย ในส่วนที่เป็น nonlinear นั่นเอง ในงานด้านการประมาณหาที่ตั้งแหล่งกำเนิดสัญญาณ จากการวัดมุมทิศนี้ จากสมการที่ (2.35) และ (2.36) เราจะสังเกตเห็นว่ามีส่วนที่เป็น nonlinear function อยู่ 2 ที่ ณ. State เดียวกันในระบบนั่นคือ ในส่วนที่เป็น ฟังก์ชันของความชัน (Slope function) ใน สมการที่ (2.36) และอีกส่วนเป็น ฟังก์ชันของมุมที่วัดเข้ามาได้ในแต่ละครั้ง (Bearing angle function) ใน สมการที่ (2.35) ในอัลกอริทึมของ EKF นั้นได้จัดการในส่วนที่เป็น nonlinear function เพียงส่วนเดียว นั่นคือในส่วนของ Bearing angle function [1],[2] เท่านั้น

ดังนั้นในการประยุกต์ กรรมวิธีการประมวลผลแบบคู่ขนาน ให้เข้ากับงานของ การประมาณหาที่ตั้งแหล่งกำเนิดสัญญาณนั้น ในระบบจำเป็นต้องมี 2 อัลกอริทึมโดยที่ทั้ง 2 อัลกอริทึมนั้นจะเริ่มด้วย ค่าเริ่มต้นตัวเดียวกัน เนื่องจากข้อมูลเริ่มต้นในส่วนของ filter processing นั้น จะได้รับมาจาก DOA processing ซึ่งจะส่งข้อมูลที่ดีที่สุดมาให้และทั้ง 2 อัลกอริทึมนั้นจะใช้ค่าที่ได้นี้เป็นข้อมูลเริ่มต้นในการประมวลผล โดยในแต่ละอัลกอริทึม จะดำเนินการในส่วนที่เป็น nonlinear function ไปพร้อมๆ กันดังแสดงในรูปที่ 2.8



รูปที่ 2.8 Parallel algorithm for MEKF

โดยอัลกอริทึมที่ 1 นั้นใช้ประมาณหาค่า $\begin{bmatrix} \tilde{x}(k) \\ \tilde{y}(k) \end{bmatrix}$ จาก $\begin{bmatrix} \hat{x}(k-1) \\ \hat{y}(k-1) \end{bmatrix}$ ซึ่งเป็น Input จาก อัลกอริทึมที่ 2 และอัลกอริทึมที่ 2 นั้นใช้ประมาณหาค่า $\begin{bmatrix} \hat{x}(k) \\ \hat{y}(k) \end{bmatrix}$ จาก $\begin{bmatrix} \tilde{x}(k-1) \\ \tilde{y}(k-1) \end{bmatrix}$ ซึ่งเป็น Input จาก อัลกอริทึมที่ 1 และทำในลักษณะนี้ไปทุกๆ State รวมเรียกว่า Parallel algorithm

จาก Parallel algorithm เราสามารถแยก อัลกอริทึมของ MEKF ออกเป็น 5 กรณีนั่นคือ

ตารางที่ 2.1 MEKF ทั้ง 5 กรณี

MEKF กรณีที่	ALGORITHM # 1	ALGORITHM # 2
1	EKFT	SKF
2	EKFT	EKFS
3	EKFT	EKFT
4	EKFS	EKFS
5	EKFS	EKFT

จากตารางที่ 2.1 เราสื่อความหมายได้ว่า MEKF กรณีที่ 1 นั้นเกิดจาก กรรมวิธีการประมวลผลแบบคู่ขนาน ระหว่าง EKFT กับ SKF ซึ่งในกรณีอื่นๆ ก็พิจารณาในลักษณะเดียวกัน

โดยเรานิยามให้

EKFT = อัลกอริทึมของ EKF ที่ดำเนินการกระจายในส่วนที่เป็น nonlinear ประเภท Bearing angle function

EKFS = อัลกอริทึมของ EKF ที่ดำเนินการกระจายในส่วนที่เป็น nonlinear ประเภท Slope function

ในแต่ละอัลกอริทึมแสดงรายละเอียดได้ดังนี้

Algorithm # 1

จากตารางที่ 2.1 เราแยกพิจารณาการดำเนินการต่อส่วนที่ไม่เป็นฟังก์ชันเชิงเส้น ออกเป็น 2 กรณีนั่นคือ EKFT (กรณีที่ 1,2,3) และ EKFS (กรณีที่ 4,5)

กรณี **EKFT** (พิจารณาในส่วน Bearing angle function $(\widetilde{\beta}(k))$)

The extrapolation of the state estimate

$$\widetilde{X}(k/k-1) = \Phi\ \widetilde{X}(k-1/k-1) \qquad (2.43)$$

โดยที่

Φ = Transition matrix

$\widetilde{X}(k-1/k-1)$ = The smoothed position estimate

$\widetilde{X}(k/k-1)$ = The prediction position estimate

The error covariance extrapolation

$$P(k/k-1) = \Phi\ P(k-1/k-1)\ \Phi^T + Q \qquad (2.44)$$

โดยที่

Q = The process noise defined a small positive number

The bearing angle measurement model

$$\beta(k) = h(\tilde{X}(k)) + v(k) \tag{2.45}$$

โดยที่

$v(k)$ = Zero mean white Gaussion noise sequence

Linearization (Jacobian matrix)

$$\tilde{\beta}(k) = h(\tilde{x}(k/k-1)) = \tan^{-1}[u] \tag{2.46}$$

$$u = \left(\frac{\tilde{y}_e(k/k-1) - y(k)}{\tilde{x}_e(k/k-1) - x(k)} \right) \tag{2.47}$$

โดยที่

$\tilde{x}_e(k/k-1), \tilde{y}_e(k/k-1)$ = ตำแหน่งของแหล่งกำเนิดสัญญาณที่ประมาณค่าได้ ณ. เวลา k

$x(k), y(k)$ = ตำแหน่งของ Observer ณ. เวลา k

เรากระจายส่วนที่ไม่เป็นฟังก์ชันเชิงเส้น ให้อยู่ในเทอมของ $H(\tilde{x}(k/k-1))$ ได้ดังนี้

$$\begin{aligned} H(\tilde{x}(k/k-1)) &= \left.\frac{\partial h(x)}{\partial x}\right|_{x=\tilde{x}(k/k-1)} \\ &= \begin{bmatrix} H_{11} & H_{12} \end{bmatrix} \end{aligned} \tag{2.48}$$

$$H_{11} = \left.\frac{\partial h}{\partial x_e}\right|_{x_e=\tilde{x}_e(k/k-1)} \tag{2.49}$$

$$H_{12} = \left.\frac{\partial h}{\partial y_e}\right|_{y_e=\tilde{y}_e(k/k-1)} \tag{2.50}$$

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

แทนค่าสมการที่ (2.46) และ (2.47) ใน สมการที่ (2.49) และ (2.50) จะได้

$$H_{11} = \frac{-u}{(1+u^2)(\tilde{x}_e(k/k-1)-x(k))} \tag{2.51}$$

$$H_{12} = \frac{1}{(1+u^2)(\tilde{x}_e(k/k-1)-x(k))} \tag{2.52}$$

Kalman gain

$$K = P(k/k-1)H^T(\tilde{X}(k/k-1))[H(\tilde{X}(k/k-1))P(k/k-1) \times H^T(\tilde{X}(k/k-1))+R(k)]^{-1} \tag{2.53}$$

โดยที่

R = ค่าความแปรปรวนของสัญญาณรบกวน $v(k)$

The state estimate equation

$$\tilde{X}(k/k) = \tilde{X}(k/k-1)+K[\beta(k)-H(\tilde{X}(k/k-1))] \tag{2.54}$$

โดยที่

$\tilde{\beta}(k) = H(\tilde{X}(k/k-1))$ = มุมที่ประมาณได้จาก State ที่แล้ว

$\beta(k) = H(\tilde{X}(k))$ = มุมที่เพิ่งวัดได้

The error covariance update

$$P(k/k) = [I-KH(\tilde{X}(k/k-1))]P(k/k-1) \tag{2.55}$$

โดยที่

I = Identity matrix

The error covariance extrapolation (next state)

$$P(k+1/k) = \Phi P(k/k)\, \Phi^T + Q \tag{2.56}$$

สถาบันวิทยบริการ จุฬาลงกรณ์มหาวิทยาลัย

กรณี EKFS (พิจารณาในส่วน Slope function (u))

The extrapolation of the state estimate

$$\widetilde{X}(k/k-1) = \Phi\ \widetilde{X}(k-1/k-1) \tag{2.57}$$

โดยที่

Φ = Transition matrix

$\widetilde{X}(k-1/k-1)$ = The smoothed position estimate

$\widetilde{X}(k/k-1)$ = The prediction position estimate

The error covariance extrapolation

$$P(k/k-1) = \Phi\ P(k-1/k-1)\ \Phi^T + Q \tag{2.58}$$

โดยที่

Q = The process noise defined a small positive number

The bearing angle measurement model

$$\beta(k) = h(\widetilde{X}(k)) + \nu(k) \tag{2.59}$$

โดยที่

$\nu(k)$ = Zero mean white Gaussion noise sequence

Linearization (Jacobian matrix)

$$\widetilde{\beta}(k) = h(\widetilde{x}(k/k-1)) = u \tag{2.60}$$

$$u = \left(\frac{\widetilde{y}_e(k/k-1) - y(k)}{\widetilde{x}_e(k/k-1) - x(k)} \right) \tag{2.61}$$

โดยที่

$\widetilde{x}_e(k/k-1), \widetilde{y}_e(k/k-1)$ = ตำแหน่งของแหล่งกำเนิดสัญญาณที่ประมาณค่าได้ ณ. เวลา k

$x(k),\ y(k)$ = ตำแหน่งของ Observer ณ. เวลา k

เรากระจายส่วนที่ไม่เป็นฟังก์ชันเชิงเส้น ให้อยู่ในเทอมของ $H(\tilde{x}(k/k-1))$ ได้ดังนี้

$$H(\tilde{x}(k/k-1)) = \left.\frac{\partial h(x)}{\partial x}\right|_{x=\tilde{x}(k/k-1)} = \begin{bmatrix} H_{11} & H_{12} \end{bmatrix} \tag{2.62}$$

$$H_{11} = \left.\frac{\partial h}{\partial x_e}\right|_{x_e=\tilde{x}_e(k/k-1)} \tag{2.63}$$

$$H_{12} = \left.\frac{\partial h}{\partial y_e}\right|_{y_e=\tilde{y}_e(k/k-1)} \tag{2.64}$$

แทนค่าสมการที่ (2.60) และ (2.61) ใน สมการที่ (2.63) และ (2.64) จะได้

$$H_{11} = \frac{-u/(\tilde{x}_e(k/k-1)-x(k))}{\tilde{x}_e(k/k-1)} \tag{2.65}$$

$$H_{12} = \frac{1/(\tilde{x}(k/k-1)-x(k))}{\tilde{y}_e(k/k-1)} \tag{2.66}$$

Kalman gain

$$K = P(k/k-1)H^T(\tilde{X}(k/k-1))[H(\tilde{X}(k/k-1))P(k/k-1) \times H^T(\tilde{X}(k/k-1)) + R(k)]^{-1} \tag{2.67}$$

โดยที่

R = ค่าความแปรปรวนของสัญญาณรบกวน $\nu(k)$

The state estimate equation

$$\tilde{X}(k/k) = \tilde{X}(k/k-1) + K[\beta(k) - \tan^{-1}(h(\tilde{X}(k/k-1)))] \tag{2.68}$$

โดยที่

$\tan^{-1}(\tilde{X}(k/k-1))$ = มุมที่ประมาณได้จาก State ที่แล้ว

$\beta(k) = H(\tilde{X}(k))$ = มุมที่เพิ่งวัดได้

The error covariance update

$$P(k/k) = [I - KH(\widetilde{X}(k/k-1))]P(k/k-1) \tag{2.69}$$

โดยที่

I = Identity matrix

The error covariance extrapolation (next state)

$$P(k+1/k) = \Phi P(k/k)\, \Phi^T + Q \tag{2.70}$$

Algorithm # 2

จากตารางที่ 2.1 เราแยกพิจารณาการดำเนินการต่อส่วนที่ไม่เป็นเป็นฟังก์ชันเชิงเส้น ออกเป็น 3 กรณีนั่นคือ SKF, EKFS และ EKFT

กรณี SKF

The extrapolation of the state estimate

$$\hat{X}(k/k-1) = \Phi\, \hat{X}(k-1/k-1) \tag{2.71}$$

โดยที่

Φ = Transition matrix

$\hat{X}(k-1/k-1)$ = The smoothed position estimate

$\hat{X}(k/k-1)$ = The prediction position estimate

The error covariance extrapolation

$$P(k/k-1) = \Phi\, P(k-1/k-1)\, \Phi^T + Q \tag{2.72}$$

โดยที่

Q = The process noise defined a small positive number

The bearing angle measurement model

$$\beta(k) = h(\hat{X}(k)) + v(k) \tag{2.73}$$

โดยที่

$v(k)$ = Zero mean white Gaussion noise sequence

Kalman gain

$$K = P(k/k-1)h^T(\hat{X}(k/k-1))[h(\hat{X}(k/k-1))P(k/k-1) \times h^T(\hat{X}(k/k-1)) + R(k)]^{-1} \quad (2.74)$$

โดยที่

R = ค่าความแปรปรวนของสัญญาณรบกวน $v(k)$

The state estimate equation

$$\hat{X}(k/k) = \hat{X}(k/k-1) + K[\beta(k) - h(\hat{X}(k/k-1))] \quad (2.75)$$

โดยที่

$\hat{\beta}(k) = h(\hat{X}(k/k-1))$ = มุมที่ประมาณได้จาก state ที่แล้ว

$\beta(k) = h(\hat{X}(k))$ = มุมที่เพิ่งวัดได้

The error covariance update

$$P(k/k) = [I - Kh(\hat{X}(k/k-1))]P(k/k-1) \quad (2.76)$$

โดยที่

I = Identity matrix

The error covariance extrapolation (next state)

$$P(k+1/k) = \Phi P(k/k)\Phi^T + Q \quad (2.77)$$

กรณี EKFS

The extrapolation of the state estimate

$$\hat{X}(k/k-1) = \Phi\hat{X}(k-1/k-1) \quad (2.78)$$

โดยที่

Φ = Transition matrix

$\hat{X}(k-1/k-1)$ = The smoothed position estimate

$\hat{X}(k/k-1)$ = The prediction position estimate

The error covariance extrapolation

$$P(k/k-1) = \Phi\ P(k-1/k-1)\ \Phi^T + Q \tag{2.79}$$

โดยที่

Q = The process noise defined a small positive number

The bearing angle measurement model

$$\beta(k) = h(\hat{X}(k)) + v(k) \tag{2.80}$$

โดยที่

$v(k)$ = Zero mean white Gaussion noise sequence

Linearization (Jacobian matrix)

$$\hat{\beta}(k) = h(\hat{x}(k/k-1)) = u \tag{2.81}$$

$$u = \left(\frac{\hat{y}_e(k/k-1) - y(k)}{\hat{x}_e(k/k-1) - x(k)} \right) \tag{2.82}$$

โดยที่

$\hat{x}_e(k/k-1), \hat{y}_e(k/k-1)$ = ตำแหน่งของแหล่งกำเนิดสัญญาณที่ประมาณค่าได้ ณ. เวลา k

$x(k),\ y(k)$ = ตำแหน่งของ Observer ณ. เวลา k

เรากระจายส่วนที่ไม่เป็น ฟังก์ชันเชิงเส้น ให้อยู่ในเทอมของ $H(\hat{x}(k/k-1))$ ได้ดังนี้

$$H(\hat{x}(k/k-1)) = \left.\frac{\partial h(x)}{\partial x}\right|_{x=\hat{x}(k/k-1)} = \begin{bmatrix} H_{11} & H_{12} \end{bmatrix} \tag{2.83}$$

$$H_{11} = \left.\frac{\partial h}{\partial x_e}\right|_{x_e=\hat{x}_e(k/k-1)} \tag{2.84}$$

$$H_{12} = \left.\frac{\partial h}{\partial y_e}\right|_{y_e=\hat{y}_e(k/k-1)} \tag{2.85}$$

แทนค่าสมการที่ (2.81) และ (2.82) ใน สมการที่ (2.84) และ (2.85) จะได้

$$H_{11} = \frac{-u/(\hat{x}_e(k/k-1)-x(k))}{\hat{x}_e(k/k-1)} \tag{2.86}$$

$$H_{12} = \frac{1/(\hat{x}(k/k-1)-x(k))}{\tilde{y}_e(k/k-1)} \tag{2.87}$$

Kalman gain

$$\begin{aligned} K = {} & P(k/k-1)H^T(\hat{X}(k/k-1))[H(\hat{X}(k/k-1))P(k/k-1) \\ & \times H^T(\hat{X}(k/k-1)) + R(k)]^{-1} \end{aligned} \tag{2.88}$$

โดยที่

R = ค่าความแปรปรวนของสัญญาณรบกวน $v(k)$

The state estimate equation

$$\hat{X}(k/k) = \hat{X}(k/k-1) + K[\beta(k) - \tan^{-1}(h(\hat{X}(k/k-1)))] \tag{2.89}$$

โดยที่

$\tan^{-1}(\hat{X}(k/k-1))$ = มุมที่ประมาณได้จาก State ที่แล้ว

$\beta(k) = H(\hat{X}(k))$ = มุมที่เพิ่งวัดได้

The error covariance update

$$P(k/k) = [I - KH(\hat{X}(k/k-1))]P(k/k-1) \tag{2.90}$$

โดยที่

I = Identity matrix

The error covariance extrapolation (next state)

จุฬาลงกรณ์มหาวิทยาลัย

$$P(k+1/k) = \Phi P(k/k)\, \Phi^T + Q \tag{2.91}$$

กรณี EKFT: ตามสมการที่ที่ (2.28) - (2.42)

The extrapolation of the state estimate (2.28)

The error covariance extrapolation (2.29)

The bearing angle measurement model (2.30)

Linearization (Jacobean matrix) (2.31,2.32,2.33,2.34,2.35,2.36,2.37,2.38)

Kalman gain (2.39)

The state estimate equation (2.40)

The error covariance update (2.41)

The error covariance extrapolation (next state) (2.42)

ในการปฏิบัติงานจริงนั้น จะเป็นการปฏิบัติทางยุทธวิธีเป็นหลัก หมายถึงต้องใช้ความรวดเร็วในการประมาณหาที่ตั้งแหล่งกำเนิดสัญญาณค่อนข้างสูงเป็นอันดับแรก และความแม่นยำที่สูงรองลงมา เนื่องจากอำนาจการทำลายล้างของกระสุนปืนใหญ่หรือ อาวุธหนักของกองทัพนั้นสามารถครอบคลุมพื้นที่ได้กว้างตั้งแต่ 0 - 10 กิโลเมตร โดยประมาณ ดังนั้นความรวดเร็วในการหาที่ตั้งแหล่งกำเนิดสัญญาณจึงมีความสำคัญค่อนข้างมาก เพราะหากว่าการประมวลผลใช้เวลานานเกินไป ฝ่ายเราจะเสียเปรียบเชิงยุทธ อาจเป็นอันตรายได้ หากฝ่ายตรงข้ามมีขีดความสามารถสูงในการประมาณหาที่ตั้งแหล่งกำเนิดสัญญาณเช่นเดียวกัน

จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 3

การจำลองบนคอมพิวเตอร์

การสร้างแบบจำลองบนคอมพิวเตอร์

วัตถุประสงค์ในการสร้างแบบจำลองบนคอมพิวเตอร์ คือการผลการเปรียบเทียบค่าความคลาดเคลื่อนไปจากเป้าหมายจริง ระหว่างอัลกอริทึมเดิม (EKF) และอัลกอริทึมที่นำเสนอขึ้นมาใหม่ในวิทยานิพนธ์ฉบับนี้ (MEKF) ทั้งกรณีเครื่องดักรับหาทิศสัญญาณเคลื่อนที่ (Moving observer) และ เครื่องดักรับหาทิศสัญญาณอยู่กับที่ (Multiple fixed observers) โดยในทั้ง 2 กรณีนั้น จะดำเนินการ simulate เพื่อทำการเปรียบเทียบผลการทดสอบตามแนวทางการ simulate ของงานวิจัยเดิม ซึ่งสามารถแยกพิจารณาได้ดังนี้

กรณีเครื่องดักรับหาทิศสัญญาณเคลื่อนที่ (Moving observer)

การ Simulate จะใช้หลักการของ K. SPINGARN [2] นั่นคือเราจะแบ่งการ Simulate ออกเป็น 2 กรณีตาม observations คือ

- กรณีที่ 1 : จำนวน Observation = 2 Observations
- กรณีที่ 2 : จำนวน Observation = 9 Observations

โดยในทั้ง 2 กรณีนั้น การทดสอบจะเริ่มต้นโดยกำหนด แหล่งกำเนิดสัญญาณอยู่ที่พิกัด (141,141) และตำแหน่งที่อากาศยานทำการวัดและประมวลผลครั้งแรกอยู่ที่พิกัด (130,11) ซึ่งในการทดสอบนี้จะกำหนดให้อากาศยานมี อัตราเร็ว 0.1 km/sec และทำการวัดทุกๆ 300 วินาที เรียงกันไป ดังนั้นพิกัดที่อากาศยาน จะทำการวัดในครั้งถัดๆไปคือ (160,41), (190,71), (220,101), (250,131), (280,161), (310,191), (340,221) และ (370,251) ตามลำดับ โดยที่ความคลาดเคลื่อนของมุมที่วัดเข้ามาได้นั้น เกิดจาก Zero mean white Gaussion noise เหมือนกัน

จุฬาลงกรณ์มหาวิทยาลัย

กรณีเครื่องดักรับหาทิศสัญญาณอยู่กับที่ (Multiple foxed observers)

การ Simulate จะใช้หลักการของ ร.อ.สุรเดช เคารพครู [2] นั่นคือเราจะแบ่งการ simulate ออกเป็น 6 กรณีตามตำแหน่งและจำนวน observers ดังนี้ (รูปที่ 2.6)

- กรณีที่ 1 : 2 observers , ตำแหน่งของ emitter อยู่ในพื้นที่ระหว่างตำแหน่งของ observer แรกกับ observer ที่สอง
- กรณีที่ 2 : 2 observers , ตำแหน่งของ emitter อยู่นอกพื้นที่ระหว่างตำแหน่งของ observer แรกกับ observer ที่สอง
- กรณีที่ 3 : 3 observers , ตำแหน่งของ emitter อยู่ในพื้นที่ระหว่างตำแหน่งของ observer แรกกับ observer ที่สาม
- กรณีที่ 4 : 3 observers , ตำแหน่งของ emitter อยู่นอกพื้นที่ระหว่างตำแหน่งของ observer แรกกับ observer ที่สาม
- กรณีที่ 5 : 4 observers , ตำแหน่งของ emitter อยู่ในพื้นที่ระหว่างตำแหน่งของ observer แรกกับ observer ที่สี่
- กรณีที่ 6 : 4 observers , ตำแหน่งของ emitter อยู่นอกพื้นที่ระหว่างตำแหน่งของ observer แรกกับ observer ที่สี่

โดยในทั้ง 6 กรณีนั้น เราจะทำการผลการทดลองทั้ง 2 block (หน้า 21) นั่นคือทั้ง block ของ MEKF และ block ของ MKF + TR โดยค่าพารามิเตอร์ต่างๆ ที่ใช้ในการ Simulate นั้นจะใช้จาก seed เดียวกัน นั่นหมายความว่า ตำแหน่งของแหล่งกำเนิดสัญญาณ (Emitter) จะอยู่ที่พิกัดเดียวกันคือ (15,15) และ ตำแหน่งต่างๆ ของ observers ในแต่ละกรณี สามารถได้ดังนี้ (0,0), (20,0), (30,0), (40,0), (60,0)และ (80,0) โดยที่ความคลาดเคลื่อนของมุมที่วัดเข้ามาได้นั้น เกิดจาก Zero mean white Gaussion noise เหมือนกัน

และสำหรับค่าเริ่มต้น (Initial state) ที่ใช้นั้นในทั้ง 2 กรณีจะดำเนินการตามแนวทางการ simulate ของ K. SPINGARN [1] นั่นคือเราจะสร้างเป็น seed เก็บไว้เหมือนๆกัน เพื่อให้ในทุกๆกรณี ได้ใช้ค่าเริ่มต้นเหมือนๆกัน

รูปแบบของการผลที่ได้จากการทดลอง

1. รูปกราฟเปรียบเทียบของอัลกอริทึมของทั้ง 2 วิธีด้วยค่า Mean square error (MSE) ทั้งแกน x และ แกน y ดังนี้

$$Mean\ square\ error\ (MSE) = \sum_{i=1}^{N} \frac{[(x(k), y(k)) - (\hat{x}_e(k/k), \hat{y}_e(k/k))]^2}{N}$$

2. รูปของ Circular error probability (CEP) of 50% ของวิธี Modified extended Kalman filter (MEKF) โดยการนำค่า ระยะทางจากจุดของตำแหน่งจริงถึง จุดของตำแหน่งที่หาได้ มาจัดลำดับมีทั้งหมด 100 ค่า เนื่องจากการทำ 100 Monte Carlo runs โดยค่า CEP of 50% คือค่ากลางของค่าที่ 50-51 นั่นหมายความว่าค่า CEP of 50% เป็นตัวกำหนดรัศมีของวงกลม โดยจะมีค่าที่น้อยกว่า CEP of 50% อยู่ในวงกลม

$$R = \sqrt{[x(k) - \hat{x}_e(k/k)]^2 + [y(k) - \hat{y}_e(k/k)]^2}$$

โดยที่ R = ระยะทางจาก Emitter ถึง ตำแหน่งที่หาได้

$[x(k), y(k)]$ = พิกัดของตำแหน่งจริง

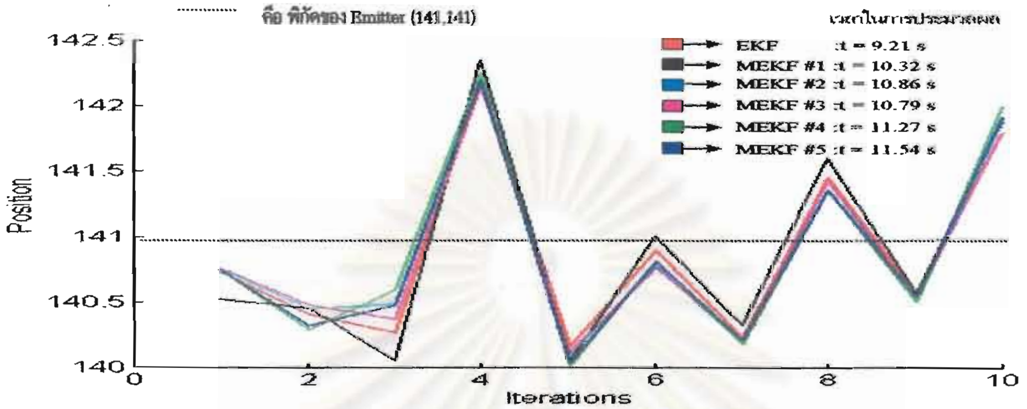
$[\hat{x}_e(k/k), \hat{y}_e(k/k)]$ = พิกัดของตำแหน่งที่หาได้

หอสมุดกลาง สถาบันวิทยบริการ จุฬาลงกรณ์มหาวิทยาลัย
สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

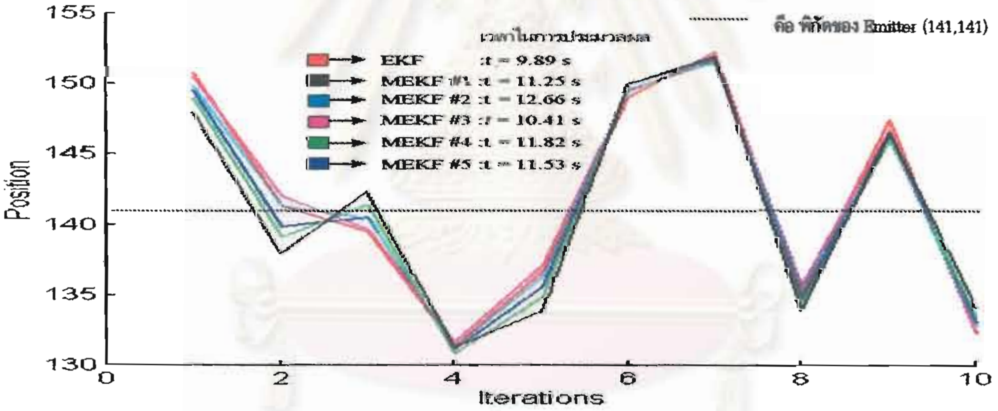
ผลการทดสอบที่ได้จากการจำลองบนคอมพิวเตอร์

กรณีเครื่องดักรับหาทิศสัญญาณเคลื่อนที่ (Moving observer)

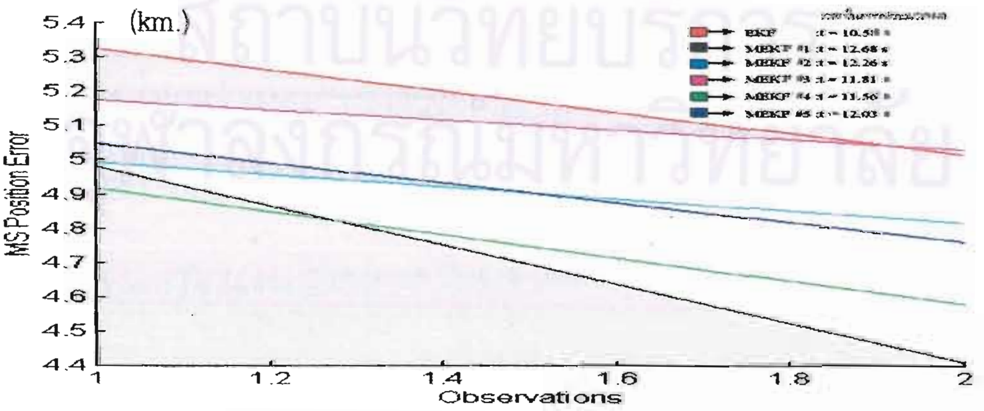
กรณีที่ 1 – จำนวน 2 Observations



รูปที่ 3.1 Position estimates บนแกน X ที่ได้จากอัลกอริทึม EKF และ MEKF ประเภทต่างๆ



รูปที่ 3.2 Position estimates บนแกน Y ที่ได้จากอัลกอริทึม EKF และ MEKF ประเภทต่างๆ



รูปที่ 3.3 MS Position error บนระนาบ XY ที่ได้จากอัลกอริทึม EKF และ MEKF ประเภทต่างๆ

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ตารางที่ 3.1 การเปรียบเทียบ Position estimates ของรูปที่ 3.1

Mon	EKF	MEKF #1	MEKF #2	MEKF #3	MEKF #4	MEKF #5	Mon	EKF	MEKF #1	MEKF #2	MEKF #3	MEKF #4	MEKF #5
1	140.72	140.52	140.75	140.75	140.74	140.74	6	140.89	141.00	140.78	140.76	140.81	140.80
2	140.39	140.45	140.44	140.47	140.28	140.31	7	140.22	140.32	140.18	140.21	140.17	140.20
3	140.27	140.05	140.48	140.36	140.59	140.47	8	141.45	141.60	141.42	141.42	141.35	141.35
4	142.19	142.35	142.20	142.15	142.25	142.19	9	140.55	140.56	140.50	140.53	140.49	140.53
5	140.16	140.05	140.09	140.12	140.00	140.03	10	141.79	141.88	141.88	141.79	141.99	141.91

ตารางที่ 3.2 การเปรียบเทียบ Position estimates ของรูปที่ 3.2

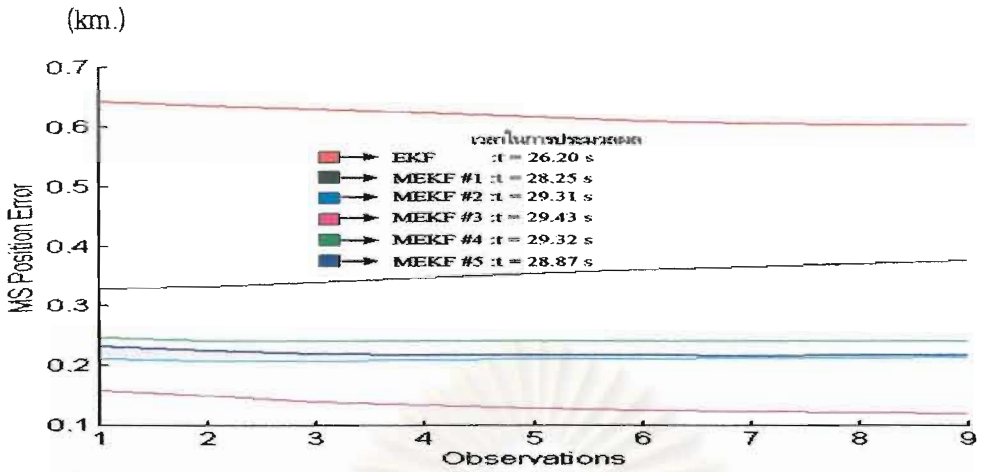
Mon	EKF	MEKF #1	MEKF #2	MEKF #3	MEKF #4	MEKF #5	Mon	EKF	MEKF #1	MEKF #2	MEKF #3	MEKF #4	MEKF #5
1	150.76	147.94	149.76	150.45	148.90	149.51	6	149.06	149.97	149.60	149.51	150.01	149.96
2	141.27	137.91	141.31	141.99	139.12	139.79	7	152.17	151.94	151.48	151.71	151.53	151.77
3	139.46	142.29	140.40	139.58	141.37	140.48	8	135.01	133.90	135.34	135.67	134.37	134.66
4	131.30	131.21	131.23	131.61	130.80	131.12	9	147.40	146.42	146.11	146.63	146.01	146.52
5	136.67	133.78	136.28	137.07	134.81	135.56	10	132.10	134.06	132.70	132.26	133.39	132.94

ตารางที่ 3.3 การเปรียบเทียบ MS Position error ของรูปที่ 3.3

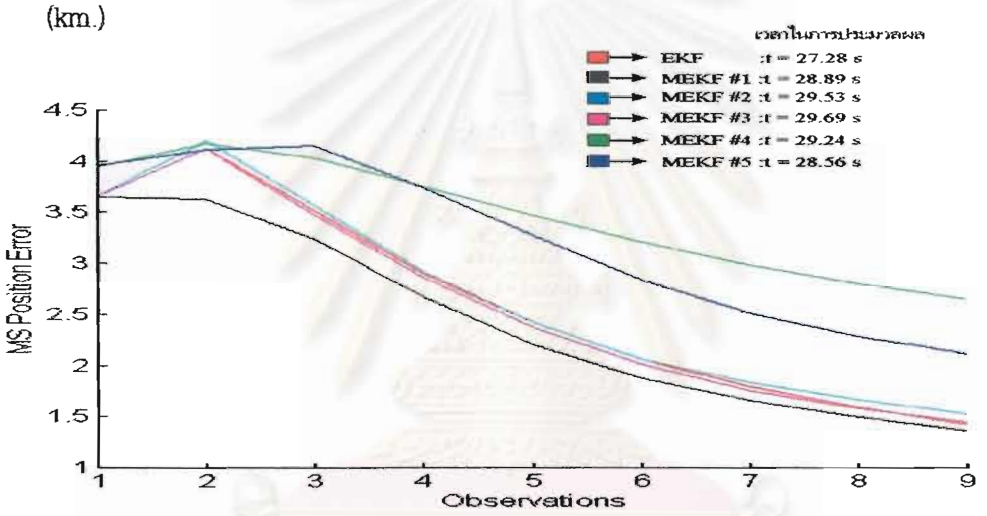
Obsv	EKF	MEKF #1	MEKF #2	MEKF #3	MEKF #4	MEKF #5	Obsv	EKF	MEKF #1	MEKF #2	MEKF #3	MEKF #4	MEKF #5
1	5.3225	4.9796	4.9922	5.1735	4.9169	5.0495	2	5.0131	4.4082	4.8149	5.0264	4.5747	4.7578

กำหนดให้ Mon คือ Monte Carlo's run [2] และ Obsv คือ Observations [3]

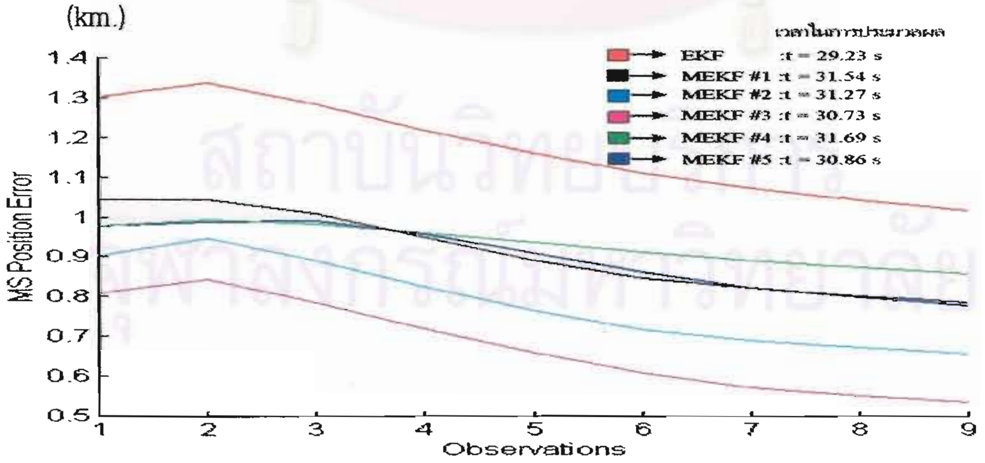
กรณีที่ 2 – จำนวน 9 Observations



รูปที่ 3.4 MS Position error บนแกน X ที่ได้จากอัลกอริทึม EKF และ MEKF ประเภทต่างๆ



รูปที่ 3.5 MS Position error บนแกน X ที่ได้จากอัลกอริทึม EKF และ MEKF ประเภทต่างๆ



รูปที่ 3.6 MS Position error บนระนาบ XY ที่ได้จากอัลกอริทึม EKF และ MEKF ประเภทต่างๆ

ตารางที่ 3.4 การเปรียบเทียบ MS Position errors ของรูปที่ 3.4

Obsv	EKF	MEKF #1	MEKF #2	MEKF #3	MEKF #4	MEKF #5
1	0.6407	0.3331	0.2126	0.1577	0.2458	0.2330
2	0.6327	0.3354	0.2075	0.1496	0.2417	0.2256
3	0.6274	0.3437	0.2086	0.1400	0.2404	0.2199
4	0.6204	0.3496	02096	0.1337	0.2402	0.2188
5	0.6133	0.3577	0.2110	0.1285	0.2401	0.2177
6	0.6075	0.3631	0.2119	0.1252	0.2402	0.2177
7	0.6041	0.3695	0.2131	0.1227	0.2403	0.2174
8	0.6023	0.3753	0.2141	0.1209	0.2405	0.2177
9	0.6015	0.3805	0.2150	0.1195	0.2406	0.2175

ตารางที่ 3.5 การเปรียบเทียบ MS Position errors ของรูปที่ 3.5

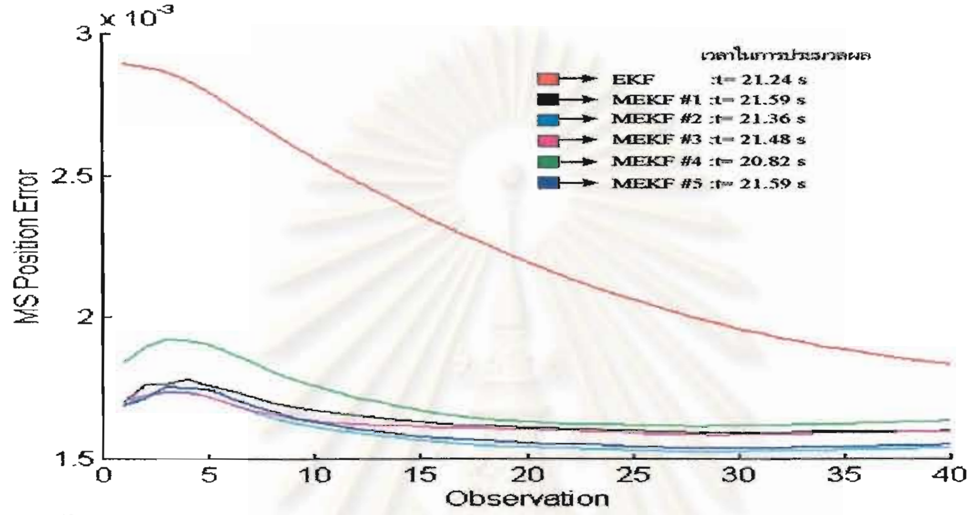
Obsv	EKF	MEKF #1	MEKF #2	MEKF #3	MEKF #4	MEKF #5
1	3.6631	3.6513	3.6596	3.6628	3.9435	3.9566
2	4.1235	3.6266	4.2042	4.1232	4.1728	4.1083
3	3.5115	3.2332	3.5618	3.4744	4.0276	4.1497
4	2.9000	2.6681	2.9199	2.8545	3.7511	3.7405
5	2.4207	2.2041	2.4164	2.3658	3.4636	3.2654
6	2.0608	1.8711	2.0552	2.0038	3.2034	2.8403
7	1.7907	1.6582	1.8275	1.7490	2.9844	2.5060
8	1.5821	1.4899	1.6598	1.5758	2.7994	2.2805
9	1.4164	1.3614	1.5252	1.4422	2.6417	2.1048

ตารางที่ 3.6 การเปรียบเทียบ MS Position errors ของรูปที่ 3.6

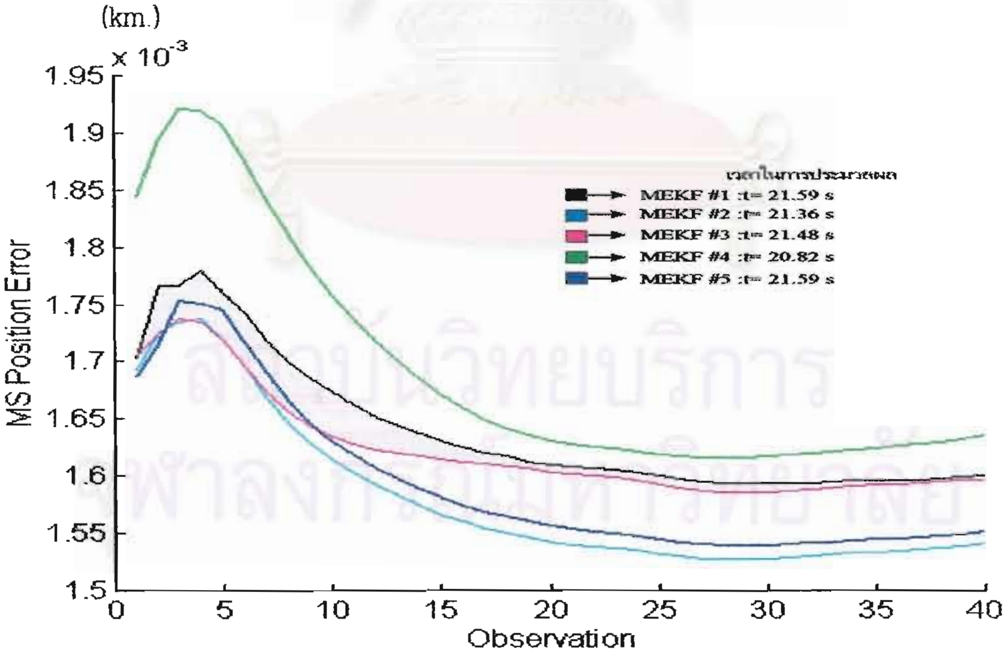
Obsv	EKF	MEKF #1	MEKF #2	MEKF #3	MEKF #4	MEKF #5
1	1.3021	1.0448	0.9045	0.8082	0.9776	0.9781
2	1.3369	1.0430	0.9457	0.8430	0.9948	0.9892
3	1.2830	1.0087	0.8902	0.7848	0.9834	0.9920
4	1.2185	0.9482	0.8235	0.7187	0.9603	0.9567
5	1.1597	0.8908	0.7626	0.6574	0.9348	0.9096
6	1.1106	0.8469	0.7160	0.6069	0.9106	0.8621
7	1.0717	0.8219	0.6898	0.5721	0.8894	0.8222
8	1.0407	0.8015	0.6707	0.5507	0.8711	0.7972
9	1.0156	0.7863	0.6551	0.5343	0.8551	0.7777

กรณีเครื่องดักรับหาทิศสัญญาณอยู่กับที่ (Multiple fixed observers)

<u>กรณีที่ 1 - จำนวน 2 Observers และ Emitter อยู่ใน Observers area</u>



รูปที่ 3.7 MS Position error บนแกน X ที่ได้จากอัลกอริทึม EKF และ MEKF ประเภทต่างๆ

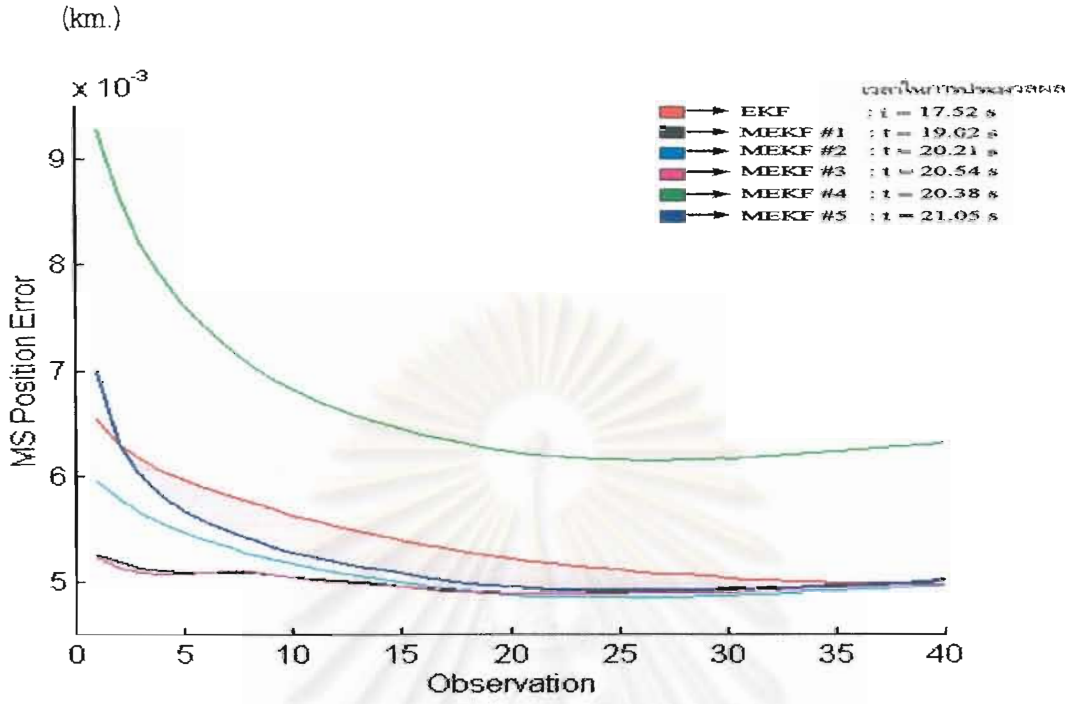


รูปที่ 3.7.1 MS Position error บนแกน X ที่ได้จากอัลกอริทึม MEKF ประเภทต่างๆ

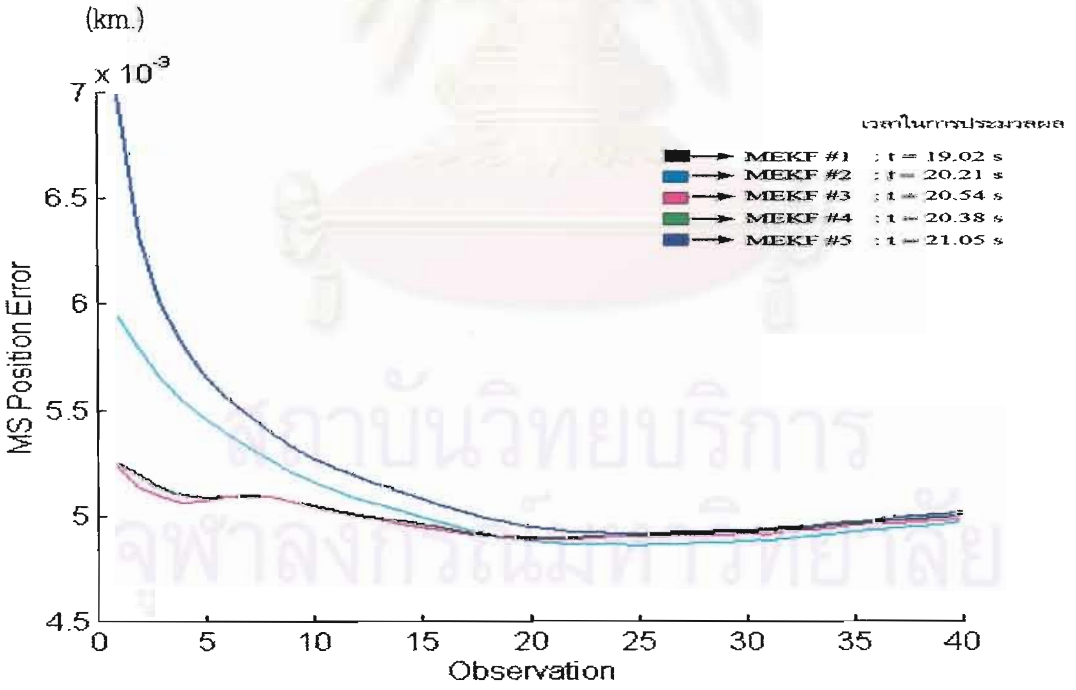
ตารางที่ 3.7 ค่าความคลาดเคลื่อนในรูป Mean square error ของรูปที่ 3.7

Obsv	EKF (10^{-3})	MEKF # 1 (10^{-3})	MEKF # 2 (10^{-3})	MEKF # 3 (10^{-3})	MEKF # 4 (10^{-3})	MEKF # 5 (10^{-3})
1	2.8951	1.6873	1.6926	1.7062	1.8443	1.6873
2	2.8819	1.7170	1.7262	1.7245	1.8951	1.7170
3	2.8690	1.7530	1.7352	1.7376	1.9213	1.7530
4	2.8388	1.7508	1.7376	1.7344	1.9197	1.7508
5	2.7982	1.7452	1.7195	1.7195	1.9061	1.7452
6	2.7507	1.7170	1.6950	1.6954	1.8751	1.7170
7	2.7017	1.6904	1.6678	1.6735	1.8409	1.6904
8	2.6534	1.6656	1.6458	1.6563	1.8095	1.6656
9	2.6075	1.6471	1.6292	1.6440	1.7831	1.6471
10	2.5635	1.6312	1.6150	1.6343	1.7589	1.6312
11	2.5210	1.6187	1.6025	1.6274	1.7375	1.6187
12	2.4797	1.6068	1.5917	1.6229	1.7176	1.6068
13	2.4398	13.5976	1.5826	1.6202	1.7002	1.5976
14	2.4013	1.5887	1.5745	1.6174	1.6846	1.5887
15	2.3641	1.5813	1.5665	1.6146	1.6709	1.5813
16	2.3281	1.5741	1.5601	1.6120	1.6592	1.5741
17	2.2933	1.5689	1.5546	1.6102	1.6496	1.5689
18	2.2600	1.5641	1.5505	1.6087	1.6418	1.5641
19	2.2279	1.5608	1.5462	1.6066	1.6359	1.5608
20	2.1969	1.5566	1.5426	1.6034	1.6309	1.5566
21	2.1673	1.5539	1.5400	1.6013	1.6276	1.5539
22	2.1390	1.5517	1.5383	1.5998	1.6251	1.5517
23	2.1122	1.5506	1.5372	1.5989	1.6237	1.5506
24	2.0866	1.5482	1.5351	1.5962	1.6217	1.5482
25	2.0623	1.5450	1.5320	1.5923	1.6193	1.5450
26	2.0394	1.5425	1.5300	1.5895	1.6176	1.5425
27	2.0176	1.5411	1.5283	1.5874	1.6166	1.5411
28	1.9971	1.5399	1.5275	1.5860	1.6163	1.5399
29	1.9778	1.5398	1.5274	1.5857	1.6167	1.5398
30	1.9596	1.5400	1.5279	1.5860	1.6175	1.5400
31	1.9426	1.5410	1.5290	1.5873	1.6189	.5410
32	1.9267	1.5417	1.5299	1.5882	1.6202	1.5417
33	1.9119	1.5426	1.5314	1.5896	1.6217	1.5426
34	1.8980	1.5437	1.5326	1.5911	1.6232	1.5437
35	1.8851	1.5446	1.5335	1.5920	1.6246	1.5446
36	1.8731	1.5454	1.5345	1.5929	1.6261	1.5454
37	1.8620	1.5466	1.5355	1.5938	1.6279	1.5466
38	1.8517	1.5476	1.5368	1.5946	1.6300	1.5476
39	1.8422	1.5493	1.5384	1.5957	1.6326	1.5493
40	1.8335	1.5510	1.5403	1.5969	1.6355	1.5510

จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ 3.8 MS Position error บนแกน Y ที่ได้จากอัลกอริทึม EKF และ MEKF ประเภทต่างๆ



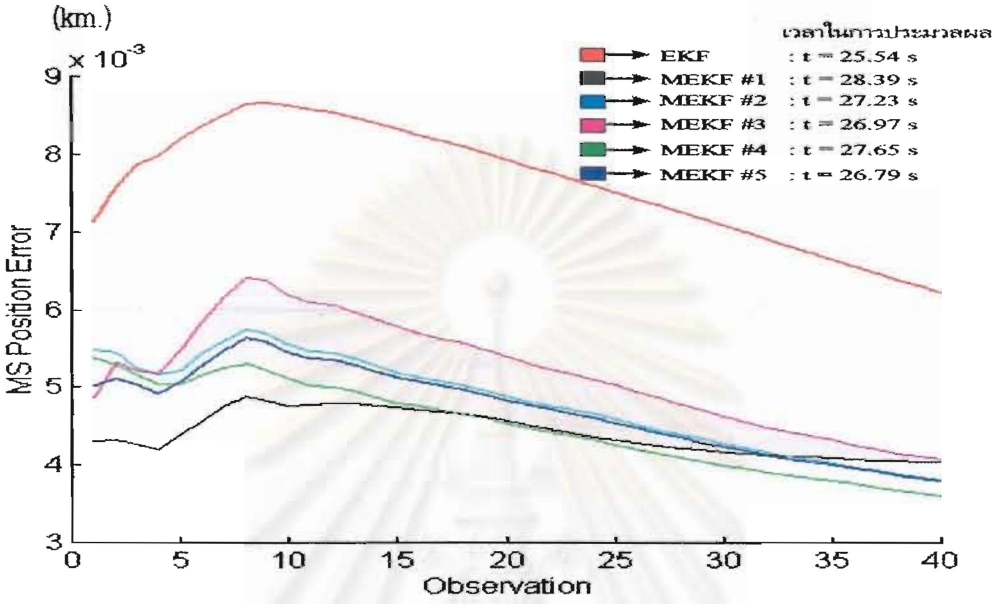
รูปที่ 3.8.1 MS Position error บนแกน Y ที่ได้จากอัลกอริทึม MEKF ประเภทต่างๆ

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

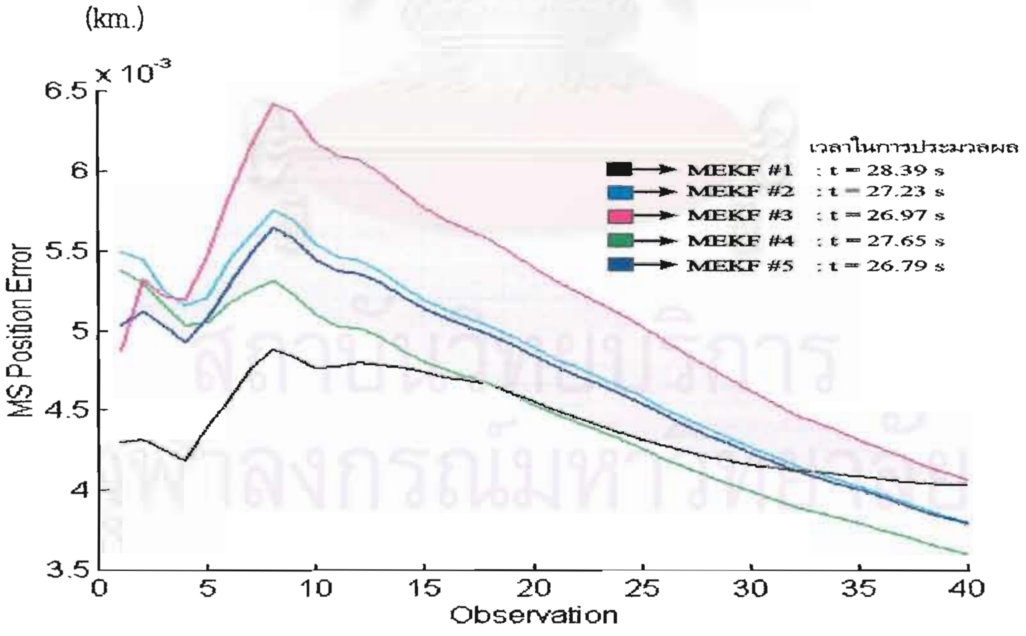
ตารางที่ 3.8 ค่าความคลาดเคลื่อนในรูป Mean square error ของรูปที่ 3.8

Obsv	EKF (10^{-3})	MEKF # 1 (10^{-3})	MEKF # 2 (10^{-3})	MEKF # 3 (10^{-3})	MEKF # 4 (10^{-3})	MEKF # 5 (10^{-3})
1	6.5354	5.2442	5.9460	5.2350	9.2774	6.9950
2	6.3072	5.1867	5.7938	5.1288	8.6593	6.3234
3	6.1693	5.1267	5.6595	5.0914	8.2305	6.0200
4	6.0555	5.0986	5.5489	5.0673	7.9046	5.8176
5	5.9649	5.0856	5.4651	5.0730	7.6301	5.6705
6	5.8898	5.0934	5.3953	5.0193	7.4140	5.5649
7	5.8218	5.0929	5.3290	5.0986	7.2325	5.4838
8	5.7588	5.0920	5.2683	5.0974	7.0721	5.4032
9	5.6959	5.0713	5.2114	5.0721	6.9360	5.3360
10	5.6351	5.0469	5.1675	5.0406	6.8252	5.2786
11	5.5796	5.0258	5.1254	5.0169	6.7294	5.2334
12	5.5279	5.0092	5.0895	4.9980	6.6458	5.1881
13	5.4793	4.9914	5.0578	4.9806	6.5743	5.1510
14	5.4338	4.9768	5.0289	4.9652	6.5091	5.1153
15	5.3918	4.9601	4.9990	4.9494	6.4491	5.0831
16	5.3527	4.9451	4.9711	4.9338	6.3937	5.0498
17	5.3160	4.9279	4.9421	4.9171	6.3425	5.0193
18	5.2818	4.9134	4.9180	4.9039	6.2973	4.9905
19	5.2500	4.9035	4.8989	4.8946	6.2588	4.9678
20	5.2206	4.8977	4.8849	4.8905	6.2267	4.9500
21	5.1937	4.8953	4.8733	4.8892	6.2002	4.9372
22	5.1697	4.8967	4.8675	4.8920	6.1807	4.9282
23	5.1479	4.9007	4.8642	4.8960	6.1677	4.9246
24	5.1279	4.9056	4.8631	4.8999	6.1574	4.9210
25	5.1098	4.9095	4.8621	4.9031	6.1500	4.9198
26	5.0936	4.9140	4.8644	4.9065	6.1474	4.9197
27	5.0787	4.9169	4.8661	4.9085	6.1487	4.9224
28	5.0653	4.9200	4.8704	4.9106	6.1534	4.9245
29	5.0529	4.9227	4.8745	4.9129	6.1608	4.9291
30	5.0415	4.9264	4.8805	4.9159	6.1704	4.9332
31	5.0314	4.9315	4.8887	4.9213	6.1829	4.9398
32	5.0225	4.9395	4.8983	4.9296	6.1962	4.9479
33	5.0147	4.9480	4.9080	4.9385	6.2102	4.9573
34	5.0080	4.9586	4.9195	4.9492	6.2253	4.9674
35	5.0021	4.9685	4.9290	4.9586	6.2409	4.9787
36	4.9970	4.9776	4.9385	4.9659	6.2559	4.9869
37	4.9926	4.9843	4.9459	4.9715	6.2705	4.9953
38	4.9891	4.9911	4.9548	4.9772	6.2859	4.9952
39	4.9864	4.9981	4.9639	4.9831	6.3023	4.9851
40	4.9847	4.9851	4.9759	4.9911	6.3210	4.9849

กรณีที่ 2 – จำนวน 2 Observers และ Emitter อยู่นอก Observers area



รูปที่ 3.9 MS Position error บนแกน X ที่ได้จากอัลกอริทึม EKF และ MEKF ประเภทต่างๆ

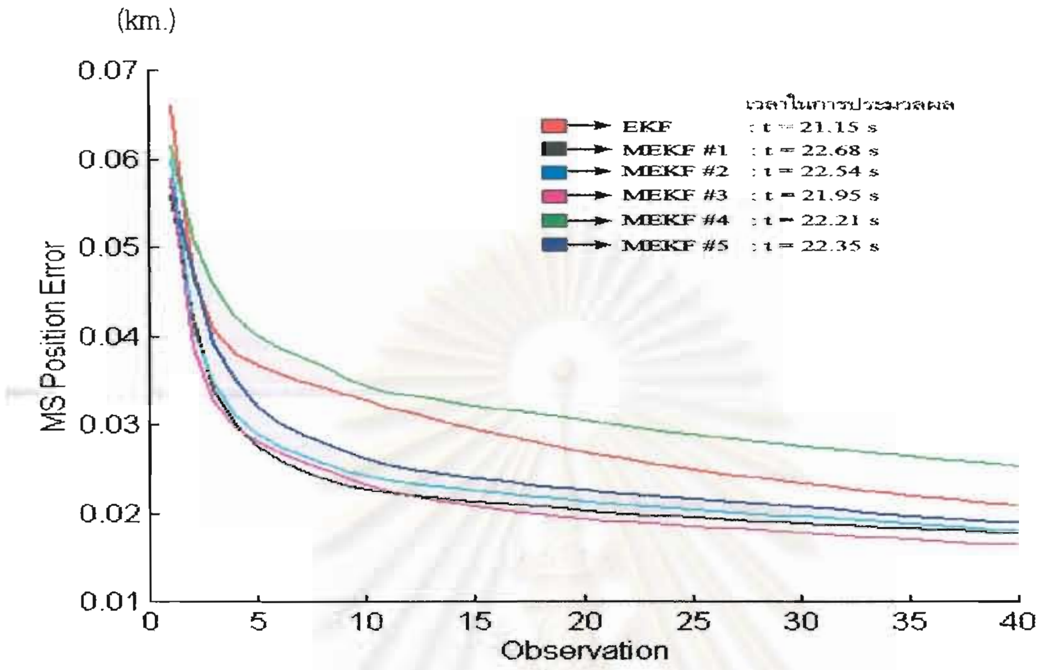


รูปที่ 3.9.1 MS Position error บนแกน X ที่ได้จากอัลกอริทึม MEKF ประเภทต่างๆ

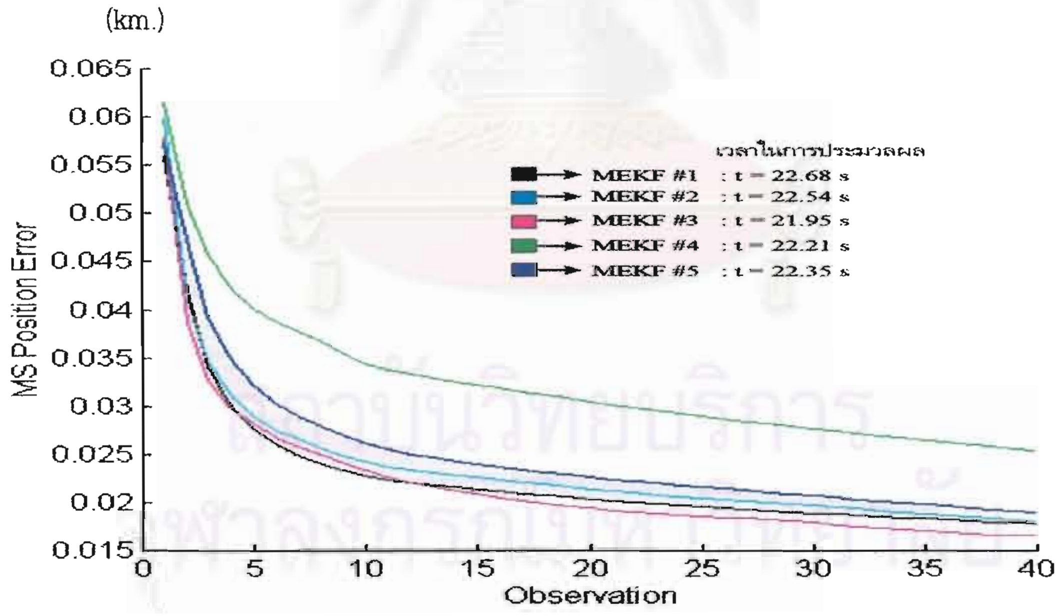
ตารางที่ 3.9 ค่าความคลาดเคลื่อนในรูป Mean square error ของรูปที่ 3.9

Obsv	EKF (10^{-3})	MEKF # 1 (10^{-3})	MEKF # 2 (10^{-3})	MEKF # 3 (10^{-3})	MEKF # 4 (10^{-3})	MEKF # 5 (10^{-3})
1	7.1335	4.3272	5.5325	4.9524	5.4546	5.0245
2	7.6354	4.3595	5.4751	5.3798	5.3302	5.1410
3	7.9268	4.3041	5.3578	5.2023	5.2034	5.0235
4	8.0012	4.5732	5.2548	5.2754	5.0782	4.9682
5	8.2120	4.4144	5.2120	5.5023	5.0655	5.3501
6	8.4389	4.6488	5.4247	5.8354	5.2315	5.3355
7	8.5487	4.8368	5.6564	6.2201	5.3025	5.5985
8	8.6158	4.9750	5.7253	6.4547	5.3872	5.6325
9	8.7024	4.8689	5.7487	6.4568	5.2201	5.6364
10	8.6321	4.8547	5.5752	6.2201	5.1358	5.4842
11	8.5076	4.8333	5.5245	6.1354	5.0828	5.4720
12	8.5054	4.8152	5.4124	6.1853	5.0568	5.4558
13	8.5001	4.8057	5.4457	6.0465	4.9654	5.4355
14	8.4859	4.8005	5.3213	5.9984	4.9548	5.2321
15	8.3265	4.7789	5.2524	5.8011	4.8687	5.1205
16	8.2645	4.7558	5.1874	5.7206	4.8652	5.1025
17	8.9871	4.7358	5.1754	5.6854	4.7203	5.0548
18	8.0541	4.7163	5.0258	5.6751	4.7231	5.0982
19	8.2150	4.6852	5.0245	5.5248	4.6908	4.9205
20	7.9213	4.6448	4.9245	5.4251	4.5650	4.8350
21	7.8587	4.5631	4.8455	5.3578	4.5205	4.8254
22	7.6548	4.5215	4.8354	5.2125	4.4879	4.7652
23	7.7132	4.4335	4.7520	5.2487	4.4284	4.7854
24	7.6875	4.4267	4.6254	5.1856	4.3259	4.6201
25	7.5123	4.3756	4.6879	5.0025	4.3321	4.5028
26	7.4687	4.3261	4.5235	4.9359	4.2287	4.5852
27	7.3851	4.2892	4.5578	4.4756	4.1021	4.4244
28	7.3023	4.2754	4.4748	4.8201	4.0205	4.2583
29	7.2865	4.2566	4.3963	4.7020	4.0028	4.3985
30	7.1028	4.2002	4.3186	4.6651	3.9875	4.3544
31	7.0498	4.1865	4.2320	4.5858	3.9874	4.2154
32	6.9856	4.1753	4.2022	4.5910	3.9321	4.1252
33	6.8002	4.1650	4.2201	4.4543	3.9215	4.0987
34	6.7568	4.1241	4.1357	4.4023	3.8201	4.0875
35	6.6256	4.1114	4.1785	4.3248	3.8010	4.0548
36	6.5332	4.1109	4.0201	4.3023	3.7328	4.0562
37	6.4514	4.1093	3.9854	4.2354	3.7218	3.9658
38	6.3269	4.0585	3.9584	4.2032	3.7125	3.9558
39	6.2043	4.0387	3.8485	4.1879	3.6548	3.8521
40	6.2836	4.0231	3.8254	4.1025	3.6254	3.8025

จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ 3.10 MS Position error บนแกน Y ที่ได้จากอัลกอริทึม EKF และ MEKF ประเภทต่างๆ



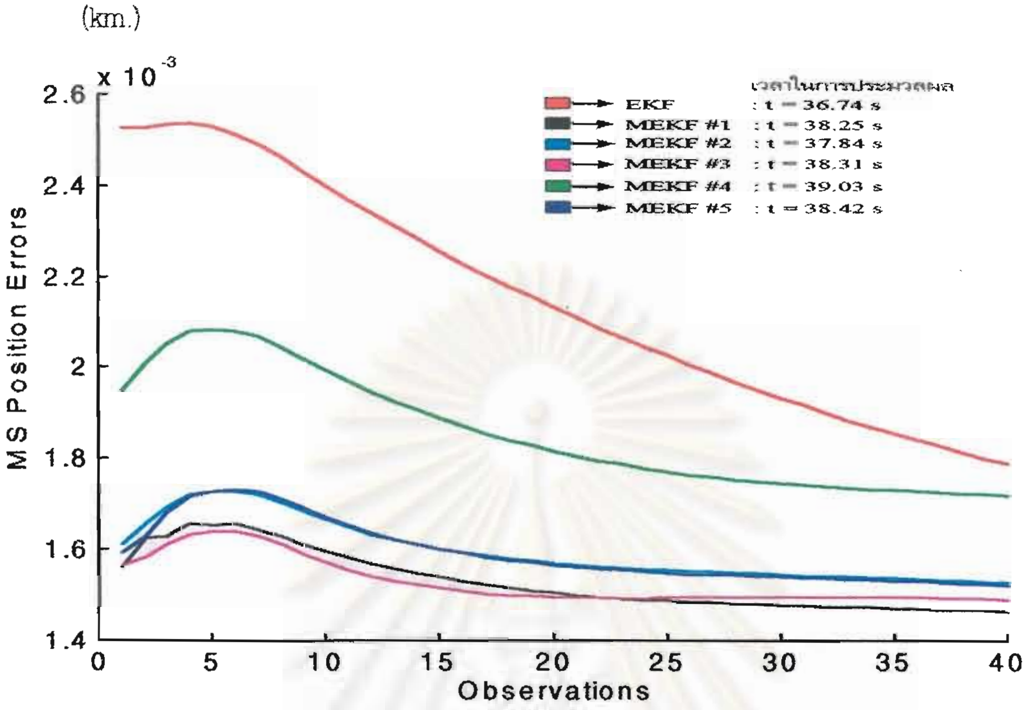
รูปที่ 3.10.1 MS Position error บนแกน Y ที่ได้จากอัลกอริทึม MEKF ประเภทต่างๆ

ตารางที่ 3.10 ค่าความคลาดเคลื่อนในรูป Mean square error ของรูปที่ 3.10

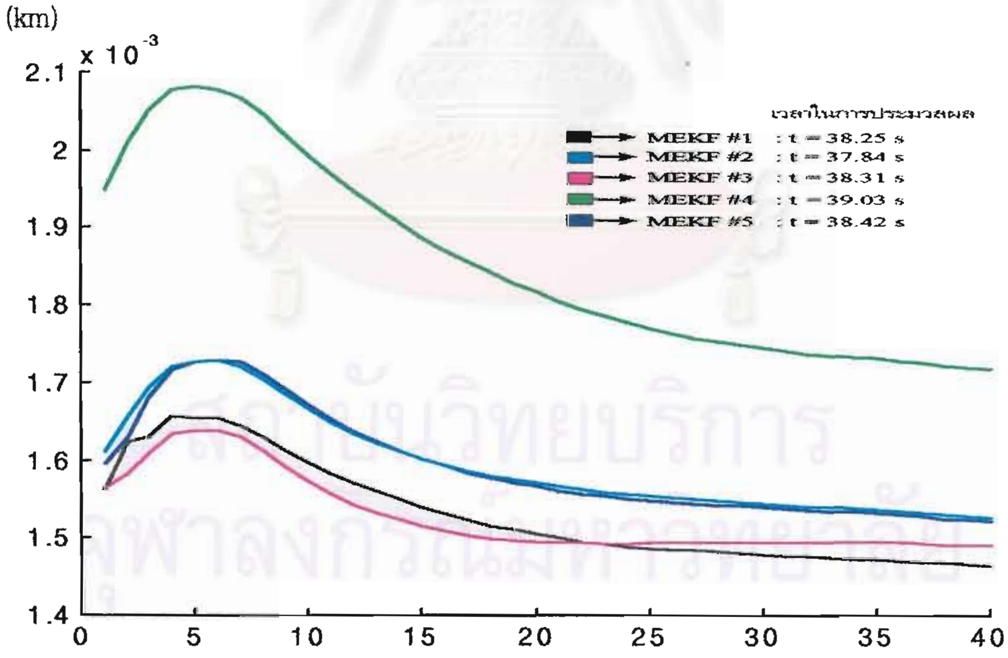
Obsv	EKF (10^{-2})	MEKF # 1 (10^{-2})	MEKF # 2 (10^{-2})	MEKF # 3 (10^{-2})	MEKF # 4 (10^{-2})	MEKF # 5 (10^{-2})
1	6.5945	5.5920	5.9925	5.7823	6.1502	5.7287
2	4.6787	4.2048	4.0936	3.8778	5.1154	4.7302
3	4.0732	3.4052	3.4614	3.2646	4.5821	3.9254
4	3.8054	3.0036	3.1185	2.9734	4.2285	3.5008
5	3.6748	2.7658	2.8920	2.8118	4.0169	3.2194
6	3.5790	2.6094	2.7500	2.6847	3.8700	3.0293
7	3.4901	2.4970	2.6514	2.5898	3.7735	2.8926
8	3.4272	2.4015	2.5601	2.4919	3.6725	2.7957
9	3.3529	2.3320	2.4805	2.4139	3.5568	2.7037
10	3.2720	2.2714	2.4268	2.3367	3.4491	2.6217
11	3.2087	2.2324	2.3763	2.2627	3.7220	2.5514
12	3.1368	2.2022	2.3487	2.2020	3.3324	2.5020
13	3.0654	2.1836	2.3195	2.1624	3.2957	2.4651
14	3.0018	2.1666	2.2854	2.1258	3.2584	2.4336
15	2.9424	2.1395	2.2658	2.0865	3.2168	2.3935
16	2.8969	2.1155	2.2332	2.0532	3.1893	2.3787
17	2.8420	2.0987	2.2135	2.0192	3.1527	2.3420
18	2.7945	2.0747	2.1837	1.9822	3.1258	2.3190
19	2.7448	2.0524	2.1692	1.9698	3.0869	2.2987
20	2.7010	2.0320	2.1318	1.9399	3.0490	2.2625
21	2.6587	2.0105	2.1154	1.9168	3.0133	2.2460
22	2.6152	2.0254	2.0985	1.9058	2.9828	2.2352
23	2.5736	1.9898	2.0863	1.8924	2.9565	2.2085
24	2.5421	1.9765	2.0625	1.8723	2.9221	2.1894
25	2.5015	1.9578	2.0498	1.8669	2.8936	2.1646
26	2.4784	1.9411	2.0336	1.8435	2.8731	2.1435
27	2.4381	1.9224	2.0134	1.8320	2.8436	2.1266
28	2.4038	1.9102	2.0017	1.8260	2.8195	2.1187
29	2.3725	1.9035	1.9819	1.8054	2.7942	2.0981
30	2.3418	1.8936	1.9735	1.7998	2.7725	2.0721
31	2.3124	1.8887	1.9565	1.7765	2.7416	2.0502
32	2.2926	1.8665	1.9334	1.7538	2.7265	2.0311
33	2.2635	1.8529	1.9227	1.7491	2.6985	2.0258
34	2.2329	1.8445	1.9069	1.7217	2.6794	2.0059
35	2.2173	1.8325	1.8884	1.7128	2.6532	1.9863
36	2.1864	1.8269	1.8726	1.6946	2.6267	1.9682
37	2.1658	1.8135	1.8504	1.6838	2.6075	1.9486
38	2.1388	1.8069	1.8450	1.6736	2.5811	1.9334
39	2.1124	1.7955	1.8224	1.6656	2.5688	1.9125
40	2.0966	1.7864	1.8115	1.6670	2.5494	1.9069

จุฬาลงกรณ์มหาวิทยาลัย

กรณีที่ 3 – จำนวน 3 Observers และ Emitter อยู่ใน Observers area



รูปที่ 3.11 MS Position error บนแกน X ที่ได้จากอัลกอริทึม EKF และ MEKF ประเภทต่างๆ

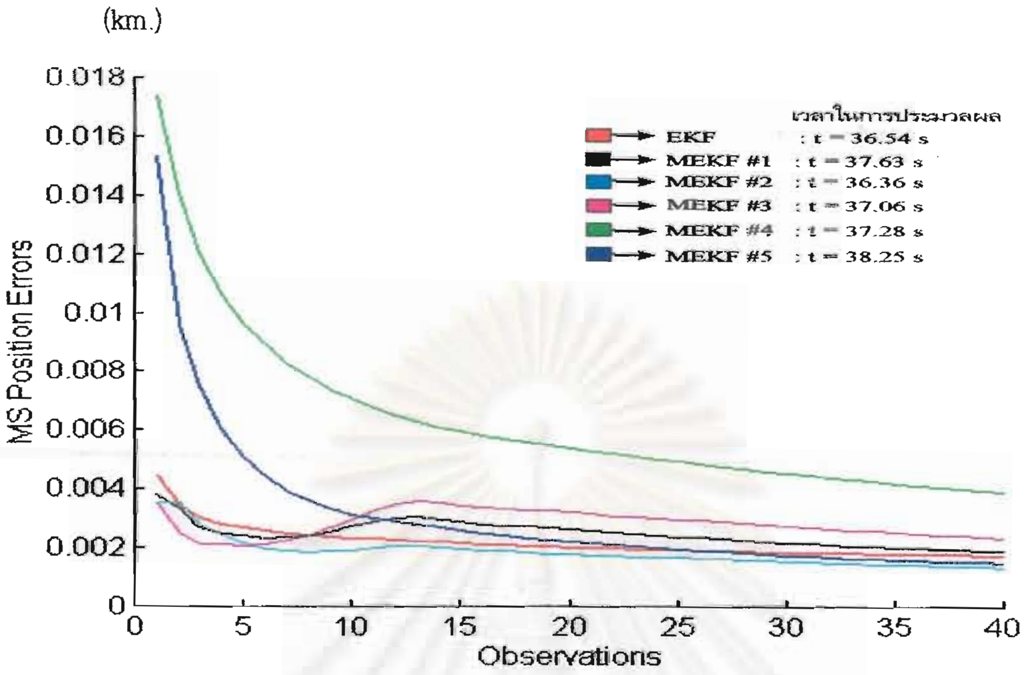


รูปที่ 3.11.1 MS Position error บนแกน X ที่ได้จากอัลกอริทึม MEKF ประเภทต่างๆ

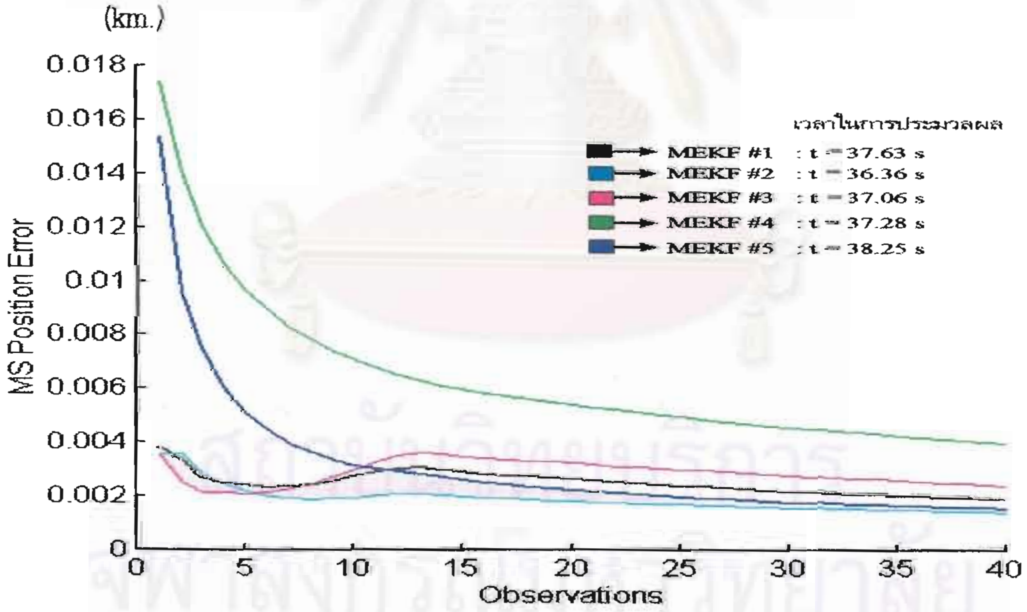
ตารางที่ 3.11 ค่าความคลาดเคลื่อนในรูป Mean square error ของรูปที่ 3.11

Obsv	EKF (10^{-3})	MEKF # 1 (10^{-3})	MEKF # 2 (10^{-3})	MEKF # 3 (10^{-3})	MEKF # 4 (10^{-3})	MEKF # 5 (10^{-3})	Obsv	EKF (10^{-3})	MEKF # 1 (10^{-3})	MEKF # 2 (10^{-3})	MEKF # 3 (10^{-3})	MEKF # 4 (10^{-3})	MEKF # 5 (10^{-3})
1	2.5261	1.5612	1.6114	1.5650	1.9478	1.5946	21	2.1092	1.4998	1.5656	1.4942	1.8047	1.5608
2	2.5244	1.6328	1.6571	1.5819	2.0089	1.6304	22	2.0873	1.9454	1.5619	1.4934	1.7947	1.5569
3	2.5339	1.6299	1.6930	1.6094	2.0511	1.6803	23	2.0660	1.9242	1.5585	1.4930	1.7855	1.5533
4	2.5375	1.6559	1.7195	1.6340	2.0779	1.7159	24	2.0453	1.4892	1.5557	1.4929	1.7773	1.5503
5	2.5300	1.6535	1.7258	1.6391	2.0817	1.7271	25	2.0251	1.4872	1.5535	1.4935	1.7701	1.5480
6	2.5138	1.6555	1.7277	1.6385	2.0776	1.7293	26	2.0056	1.4850	1.5514	1.4940	1.7636	1.5459
7	2.4904	1.6443	1.7201	1.6290	2.0676	1.7270	27	1.9866	1.4834	1.5498	1.4948	1.7580	1.5440
8	2.4620	1.6302	1.7041	1.6170	2.0457	1.7102	28	1.9682	1.4814	1.5481	1.4953	1.7532	1.5426
9	2.4315	1.6129	1.6852	1.5905	2.0200	1.6905	29	1.9503	1.4800	1.5462	1.4949	1.7486	1.5407
10	2.4002	1.5977	1.6663	1.5719	1.9939	1.6711	30	1.9328	1.4781	1.5440	1.4950	1.7444	1.5391
11	2.3692	1.5829	1.6487	1.5554	1.9687	1.6528	31	1.9157	1.4769	1.5424	1.4948	1.7409	1.5372
12	2.3389	1.5698	1.6336	1.5418	1.9455	1.6364	32	1.8992	1.4751	1.5406	1.4942	1.7377	1.5356
13	2.3098	1.5598	1.6217	1.5315	1.9249	1.6230	33	1.8832	1.4740	1.5393	1.4941	1.7348	1.5339
14	2.2816	1.5491	1.6107	1.5224	1.9053	1.6112	34	1.8678	1.4728	1.5381	1.4953	1.7325	1.5329
15	2.2546	1.5405	1.6014	1.5149	1.8874	1.6008	35	1.8528	1.4714	1.5363	1.4947	1.7298	1.5310
16	2.2285	1.5314	1.5932	1.5086	1.8711	1.5920	36	1.8384	1.4698	1.5339	1.4936	1.7269	1.5289
17	2.2032	1.5234	1.5857	1.5033	1.8553	1.5836	37	1.8245	1.4681	1.5312	1.4920	1.7238	1.5263
18	2.1787	1.5159	1.5796	1.4994	1.8408	1.5763	38	1.8111	1.4668	1.5288	1.4909	1.7211	1.5240
19	2.1549	1.5099	1.5744	1.4972	1.8279	1.5707	39	1.7982	1.4656	1.5268	1.4902	1.7187	1.5222
20	2.1317	1.5043	1.5698	1.4953	1.8157	1.5653	40	1.7858	1.4644	1.5250	1.4896	1.7166	1.5206

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ 3.12 MS Position error บนแกน Y ที่ได้จากอัลกอริทึม EKF และ MEKF ประเภทต่างๆ



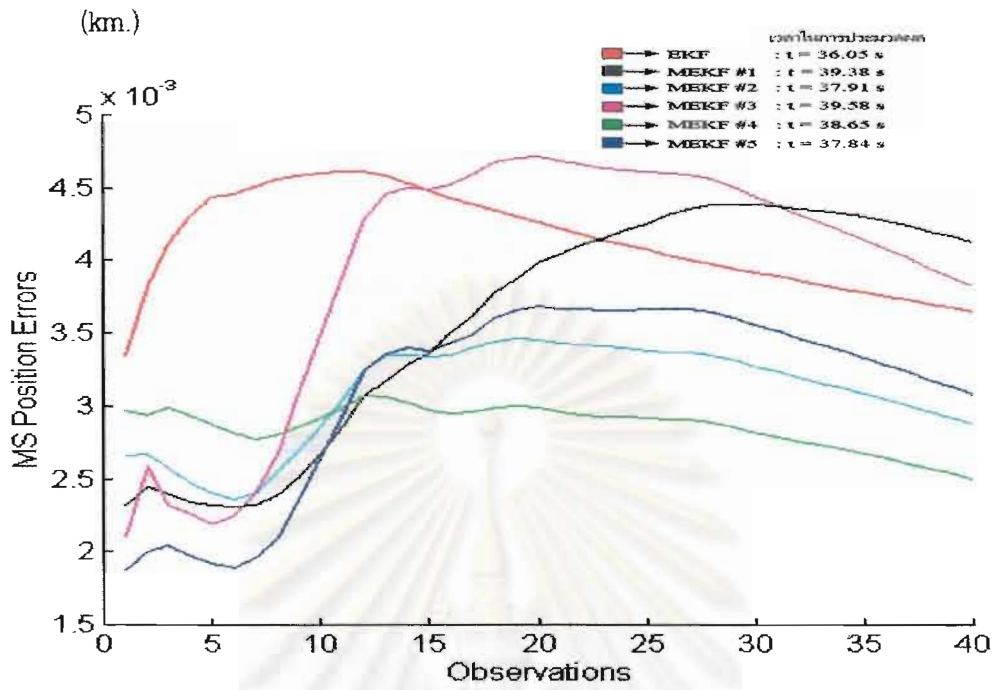
รูปที่ 3.12.1 MS Position error บนแกน Y ที่ได้จากอัลกอริทึม MEKF ประเภทต่างๆ

ตารางที่ 3.12 ค่าความคลาดเคลื่อนในรูป Mean square error ของรูปที่ 3.12

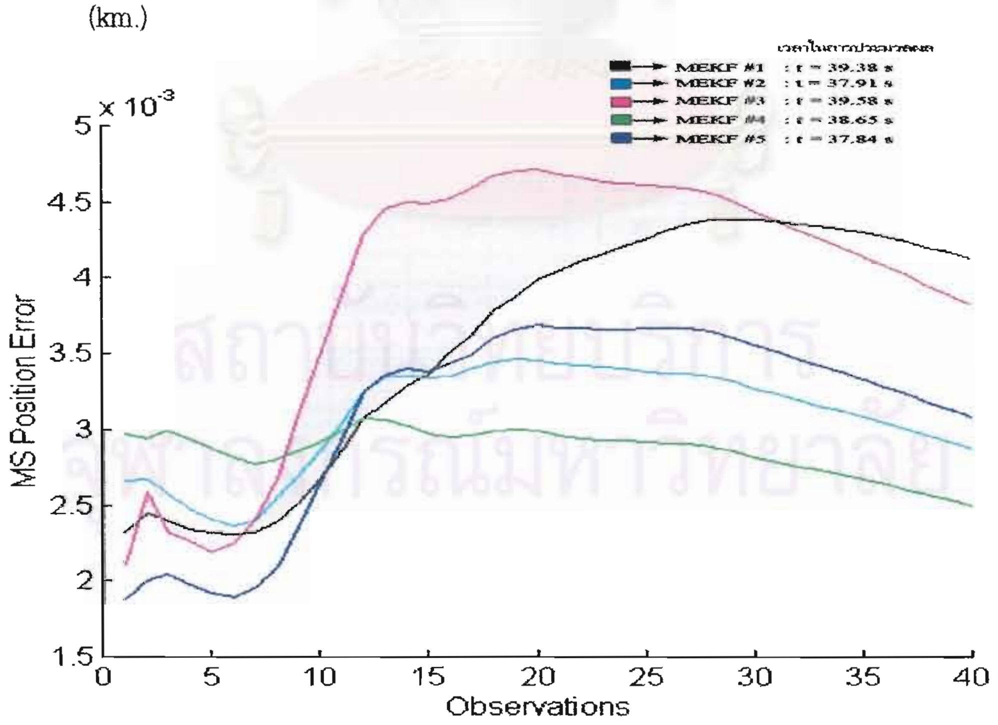
Obsv	EKF (10^{-3})	MEKF # 1 (10^{-3})	MEKF # 2 (10^{-3})	MEKF # 3 (10^{-3})	MEKF # 4 (10^{-3})	MEKF # 5 (10^{-3})
1	4.4428	3.7972	3.5552	3.4870	17.370	15.303
2	3.4242	3.3011	3.5469	2.4689	14.031	9.5996
3	3.0117	2.7170	2.8163	1.1260	12.031	7.5096
4	2.8057	2.4847	2.4041	2.0977	10.593	6.0091
5	2.6795	2.4097	2.1409	2.0839	9.6284	5.0739
6	2.5708	2.3032	1.9678	2.1093	8.8958	4.4079
7	2.4974	2.3686	1.9142	2.2659	8.2535	3.9252
8	2.4425	2.4354	1.8463	2.4175	7.7907	3.5737
9	2.3923	2.5474	1.8788	2.6696	7.4113	3.3009
10	2.3482	2.7246	1.9130	2.9725	7.0757	3.1240
11	2.3090	2.9049	2.0077	3.2768	6.7727	2.9849
12	2.2720	3.0301	2.0560	3.5043	6.5031	2.9058
13	2.2339	3.0375	2.0472	3.5746	6.2754	2.8127
14	2.1974	2.9799	2.0079	3.5285	6.0944	2.6996
15	2.1644	2.8863	1.9530	3.4387	5.5907	2.5889
16	2.1322	2.8180	1.9212	3.3699	5.8175	2.4875
17	2.1037	2.7619	1.8831	3.3280	5.6974	2.4069
18	2.0764	2.7224	1.8587	3.2970	5.5863	2.3301
19	2.0510	2.6714	1.8205	3.2521	5.4871	2.2641
20	2.0274	2.6189	1.7943	3.1994	5.3919	2.1985
21	2.0053	2.5617	1.7590	3.1412	5.2995	2.1436
22	1.9845	2.5101	1.7360	3.0878	5.2071	2.0870
23	1.9648	2.4631	1.7057	3.0404	5.1174	2.0406
24	1.9468	2.4243	1.6870	3.0033	5.0283	1.9916
25	1.9298	2.3914	1.6650	2.9732	4.9394	1.9527
26	1.9136	2.3615	1.6512	2.9447	43853	1.9118
27	1.8986	2.3254	1.6264	2.9064	4.7690	1.8778
28	1.8846	2.2804	1.5986	2.8541	4.6918	1.8369
29	1.8712	2.2334	1.5673	2.7977	4.6147	1.7983
30	1.8584	2.1900	1.5392	2.7432	4.5380	1.7586
31	1.8460	2.1508	1.5129	2.6939	4.4610	1.7232
32	1.8342	2.1154	1.4884	2.6472	4.3878	1.6889
33	1.8231	2.0828	1.4668	2.6045	4.3186	1.6576
34	1.8126	2.0519	1.4439	2.5623	4.2538	1.6296
35	1.8027	2.0213	1.4231	2.5204	4.1897	1.6013
36	1.7934	1.9935	1.4043	2.4819	4.1272	1.5759
37	1.7846	1.9664	1.3849	2.4442	4.0667	1.5528
38	1.7761	1.9393	1.3662	2.4055	4.0094	1.5290
39	1.7680	1.9107	1.3465	2.3658	3.9553	1.5065
40	1.7604	1.8813	1.3261	2.3251	3.9042	1.4836

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

กรณีที่ 4- จำนวน 3 Observers และ Emitter อยู่นอก Observers area



รูปที่ 3.13 MS Position error บนแกน X ที่ได้จากอัลกอริทึม EKF และ MEKF ประเภทต่างๆ

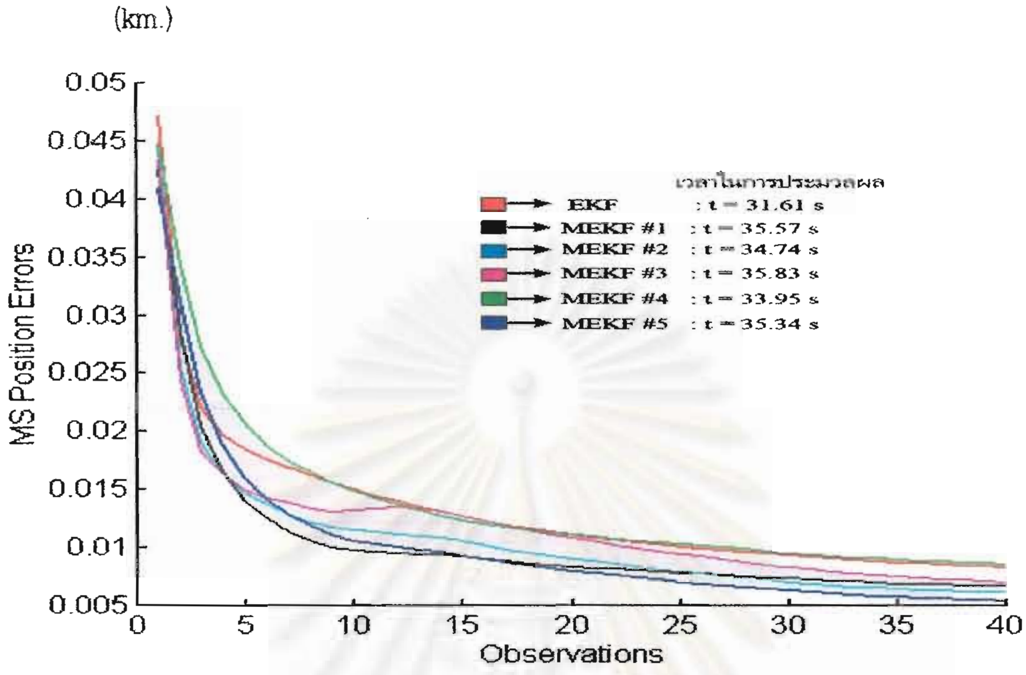


รูปที่ 3.13.1 MS Position error บนแกน X ที่ได้จากอัลกอริทึม MEKF ประเภทต่างๆ

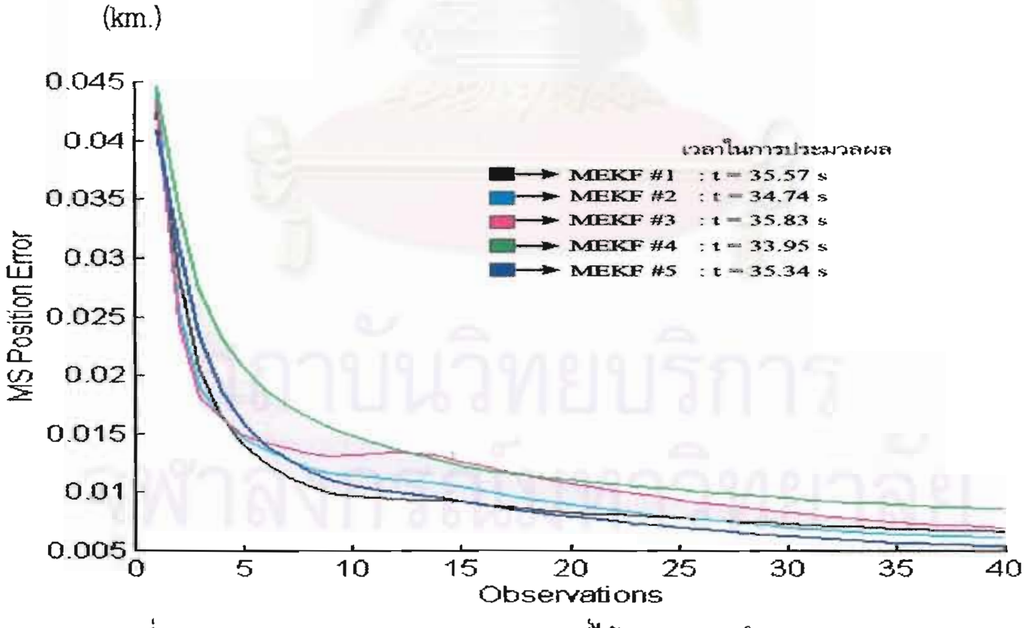
ตารางที่ 3.13 ค่าความคลาดเคลื่อนในรูป Mean square error ของรูปที่ 3.13

Obsv	EKF (10 [illegible])	MEKF # 1 (10 [illegible])	MEKF # 2 (10 [illegible])	MEKF # 3 (10 [illegible])	MEKF # 4 (10 [illegible])	MEKF # 5 (10 [illegible])
1	3.3445	2.3249	2.6624	2.1099	2.9702	1.8797
2	3.8220	2.4489	2.6742	2.5916	2.9399	2.0042
3	4.1038	2.4076	2.5762	2.3259	2.9879	2.0442
4	4.3044	2.3393	2.4707	2.2592	2.9352	1.9720
5	4.4349	2.3248	2.4003	2.1852	2.8724	1.9192
6	4.4590	2.3136	2.3617	2.2489	2.8119	1.8940
7	4.5066	2.3202	2.4023	2.4173	2.7769	1.9599
8	4.5547	2.3881	2.5599	2.6800	2.8001	2.0986
9	4.5827	2.5105	2.6931	3.0763	2.8538	2.3500
10	4.6006	2.6680	2.8531	3.4814	2.9198	2.6401
11	4.6128	2.8541	3.0384	3.8855	2.9901	2.9242
12	4.6150	3.0713	3.2493	4.2562	3.0671	3.2435
13	4.5842	3.1739	3.3494	4.4608	3.0629	3.3553
14	4.4325	3.2832	3.3488	4.5000	3.0184	3.3940
15	4.4474	2.3588	3.3384	4.4885	2.9657	3.3783
16	4.4240	3.5006	3.3438	4.5197	2.9448	3.4299
17	4.3801	3.6187	3.4011	4.5861	2.9589	3.4948
18	4.3405	3.7832	3.4397	4.6572	2.9410	3.6085
19	4.2995	3.8727	3.4611	4.7064	2.9987	3.6534
20	4.2584	3.9803	3.4453	4.7093	2.9841	3.6816
21	4.2170	4.0321	3.4319	4.6797	2.9589	3.6626
22	4.1776	4.1051	3.4158	4.6576	2.9410	3.6631
23	4.1401	4.1470	3.4097	4.6351	2.9307	3.6540
24	4.1049	4.2115	3.3948	4.6250	2.9244	3.6595
25	4.0713	4.2532	3.3818	4.6130	2.9171	3.6607
26	4.0389	4.3137	3.3686	4.6018	2.9087	3.6631
27	4.0083	4.3581	3.3642	4.5906	2.9027	3.6660
28	3.9775	4.3831	3.3446	4.5538	2.8827	3.6245
29	3.9469	4.3890	3.3118	4.5033	2.8536	3.6078
30	3.9165	4.3801	3.2705	4.4358	2.8180	3.5550
31	3.8870	4.3705	3.2298	4.3741	2.7856	3.5099
32	3.8584	4.3561	3.1940	4.3115	2.7561	3.4616
33	3.8306	4.3452	3.1575	4.2578	2.7287	3.4226
34	3.8032	4.3240	3.1194	4.1977	2.6986	3.3764
35	3.7762	4.2698	3.0796	4.1377	2.6350	3.3308
36	3.7498	4.2373	3.0394	4.0741	2.6671	3.2805
37	3.7236	4.3890	3.0257	4.0104	2.6015	3.2303
38	3.6976	4.1994	2.9025	3.9438	2.5670	3.1775
39	3.6719	4.1638	2.9102	3.8805	2.5336	3.1280
40	3.6462	4.1232	2.7602	3.8147	2.5004	3.0764

จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ 3.14 MS Position error บนแกน Y ที่ได้จากอัลกอริทึม EKF และ MEKF ประเภทต่างๆ



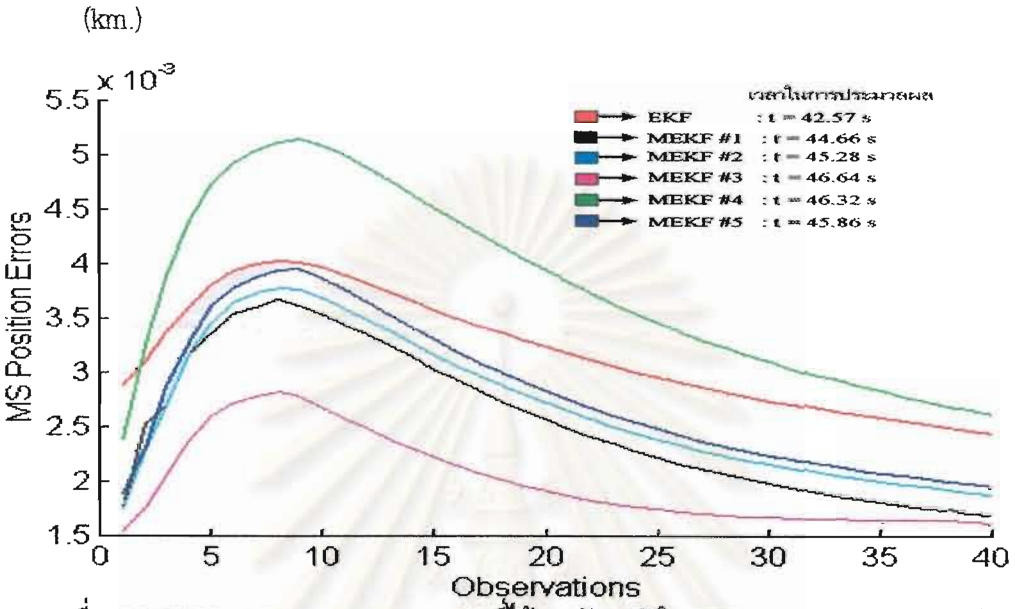
รูปที่ 3.14.1 MS Position error บนแกน Y ที่ได้จากอัลกอริทึม MEKF ประเภทต่างๆ

ตารางที่ 3.14 ค่าความคลาดเคลื่อนในรูป Mean square error ของรูปที่ 3.14

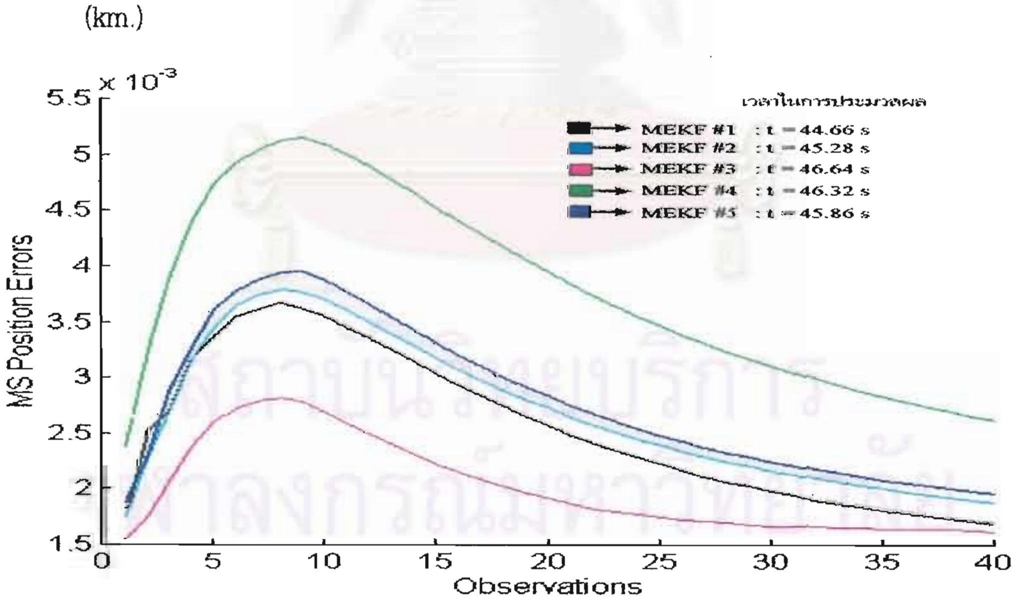
Obsv	EKF (10^{-3})	MEKF # 1 (10^{-3})	MEKF # 2 (10^{-3})	MEKF # 3 (10^{-3})	MEKF # 4 (10^{-3})	MEKF # 5 (10^{-3})
1	47.118	42.456	44.412	43.458	44.673	40.893
2	28.050	28.622	25.446	24.380	34.311	31.146
3	21.992	20.505	19.044	18.194	27.315	23.387
4	19.637	16.496	16.379	16.385	23.275	18.763
5	18.502	13.987	14.599	14.879	20.681	15.956
6	17.553	12.468	13.731	14.235	18.761	14.073
7	16.836	11.382	12.846	13.850	17.410	12.821
8	16.187	10.544	12.072	13.319	16.419	11.826
9	15.551	9.9735	11.692	13.073	15.585	11.064
10	14.963	9.7264	11.452	13.193	14.875	10.605
11	14.428	9.6087	11.289	13.359	14.246	10.271
12	13.930	9.4916	11.057	13.447	13.685	10.046
13	13.461	9.3548	10.990	13.355	13.169	9.7720
14	13.020	9.3214	10.750	13.075	12.713	9.5802
15	12.619	9.1649	10.495	12.676	12.312	9.0730
16	12.255	9.0120	10.123	12.252	11.976	8.9985
17	11.924	8.7808	9.7905	11.833	11.704	8.6790
18	11.622	8.5972	9.4739	11.455	11.478	8.4128
19	11.343	8.4204	9.2206	11.103	11.265	8.1576
20	11.087	8.3262	8.9883	10.798	11.070	7.9578
21	10.849	8.2149	8.7646	10.500	10.888	7.7657
22	10.627	8.1217	8.5373	10.210	10.722	7.5816
23	10.418	7.9915	8.3092	9.9127	10.563	7.3904
24	10.225	7.8861	8.1001	9.6391	10.406	7.2036
25	10.044	7.7838	7.9006	9.9315	10.251	7.0320
26	9.8737	7.7125	7.7134	9.1621	10.101	6.8708
27	9.7146	7.6179	7.5266	8.9347	9.9606	6.7191
28	9.5664	7.5210	7.3516	8.7077	9.8176	6.5681
29	9.4275	7.4241	7.1841	8.4962	9.6719	6.4265
30	9.2961	7.3337	7.0288	8.2976	9.5265	6.2924
31	9.1717	7.2452	6.8807	8.1131	9.3902	6.1656
32	9.0543	7.1537	6.7428	7.9331	9.2656	6.0451
33	8.9443	7.0635	6.6262	7.7709	9.1545	5.9311
34	8.8412	6.9809	6.5270	7.6303	9.0544	5.8368
35	8.7430	6.9098	6.4308	7.4969	8.9564	5.7487
36	8.6497	6.8439	6.3541	7.3788	8.8641	5.6659
37	8.5609	6.7909	6.2894	7.2806	8.7826	5.6002
38	8.4768	6.7456	6.2344	7.1925	8.7121	5.5450
39	8.3967	6.7057	6.1812	7.1114	8.6495	5.4991
40	8.3205	6.6727	6.1359	7.0360	8.5907	5.4580

จุฬาลงกรณ์มหาวิทยาลัย

<u>กรณีที่ 5 - จำนวน 4 Observers และ Emitter อยู่ใน Observers area</u>



รูปที่ 3.15 MS Position error บนแกน X ที่ได้จากอัลกอริทึม EKF และ MEKF ประเภทต่างๆ

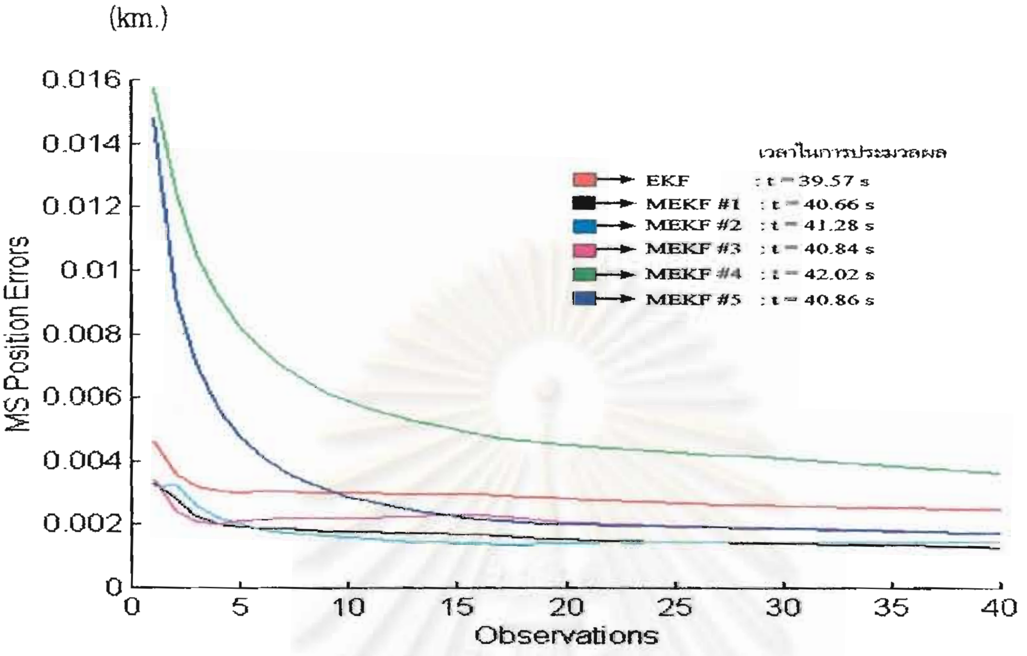


รูปที่ 3.15.1 MS Position error บนแกน X ที่ได้จากอัลกอริทึม MEKF ประเภทต่างๆ

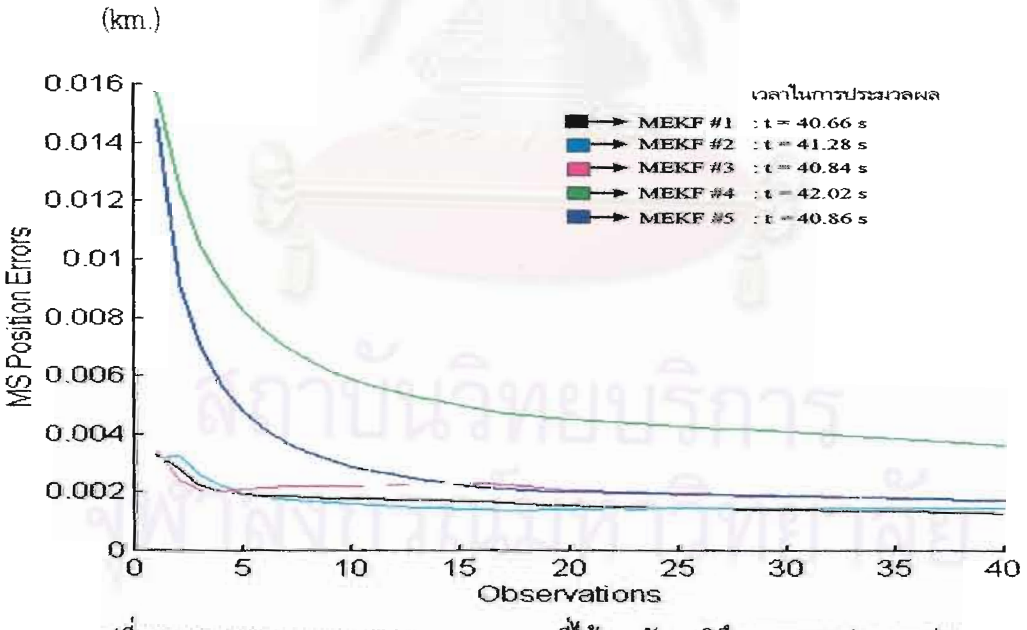
ตารางที่ 3.15 ค่าความคลาดเคลื่อนในรูป Mean square error ของรูปที่ 3.15

Obsv	EKF (10^{-3})	MEKF # 1 (10^{-3})	MEKF # 2 (10^{-3})	MEKF # 3 (10^{-3})	MEKF # 4 (10^{-3})	MEKF # 5 (10^{-3})
1	2.8892	1.7473	1.7458	1.5418	2.3832	1.8782
2	3.0957	2.5229	2.2931	1.7634	3.2431	2.3122
3	3.3616	2.6908	2.6867	2.0633	3.8832	2.8723
4	3.6094	3.1621	3.1562	2.3740	4.3802	3.2608
5	3.8131	3.3564	3.4510	2.6042	4.7327	3.6157
6	3.9399	3.5381	3.6420	2.7244	4.9292	3.7769
7	4.0015	3.6067	3.7346	2.7856	5.0471	3.8870
8	4.0289	3.6743	3.7846	2.8195	5.1215	3.9444
9	4.0167	3.6192	3.7671	2.7736	5.1489	3.9498
10	3.9679	3.5514	3.6974	2.6837	5.0920	3.8650
11	3.8983	3.4477	3.5944	2.5798	4.9963	3.7602
12	3.8206	3.3586	3.4870	2.4819	4.8849	3.6467
13	3.7400	3.2455	3.3760	2.3873	4.7623	3.5290
14	3.6592	3.1443	3.2711	2.3003	4.6381	3.4124
15	3.5797	3.0360	3.1638	2.2204	4.5140	3.3026
16	3.5026	2.9342	3.0628	2.1478	4.3921	3.1933
17	3.4302	2.8354	2.9679	2.0802	4.2752	3.0913
18	3.3609	2 7407	2.8783	2.0180	4.1595	2.9949
19	3.2950	2.6515	2.7944	1.9611	4.0470	2.9048
20	3.2323	2.5690	2.7175	1.9084	3.9392	2.8218

Obsv	EKF (10^{-3})	MEKF # 1 (10^{-3})	MEKF # 2 (10^{-3})	MEKF # 3 (10^{-3})	MEKF # 4 (10^{-3})	MEKF # 5 (10^{-3})
21	3.1718	2.4896	2.6424	1.8611	3.8348	2.7442
22	3.1138	2.4163	2.5720	1.8213	3.7336	2.6700
23	3.0583	2.3467	2.5052	1.7891	3.6367	2.6010
24	3.0054	2.2822	2.4436	1.7648	3.5446	2.5368
25	2.9553	2.2228	2.3871	1.7454	3.4579	2.4781
26	2.9080	2.1673	2.3347	1.7272	3.3768	2.4238
27	2.8632	2.1158	2.2859	1.7101	3.3009	2.3733
28	2.8206	2.0677	2.2407	1.6940	3.2296	2.3262
29	2.7802	2.0234	2.1985	1.6807	3.1625	2.2826
30	2.7415	1.9811	2.1581	1.6692	3.0980	2.2411
31	2.7047	1.9420	2.1199	1.6595	3.0362	2.2015
32	2.6694	1.9046	2.0833	1.6506	2.9769	2.1640
33	2.6356	1.8695	2.0485	1.6427	2.9200	2.1282
34	2.6032	1.8371	2.0163	1.6364	2.8665	2.0952
35	2.5721	1.8076	1.9872	1.6324	2.8162	2.0649
36	2.5423	1.7812	1.9605	1.6315	2.7691	2.0377
37	2.5135	1.7561	1.9356	1.6297	2.7249	2.0107
38	2.4856	1.7326	1.9116	1.6256	2.6830	1.9880
39	2.4586	1.7091	1.8875	1.6186	2.6433	1.9641
40	2.4324	1.6861	1.8643	1.6089	2.6057	1.9399



รูปที่ 3.16 MS Position error บนแกน Y ที่ได้จากอัลกอริทึม EKF และ MEKF ประเภทต่างๆ

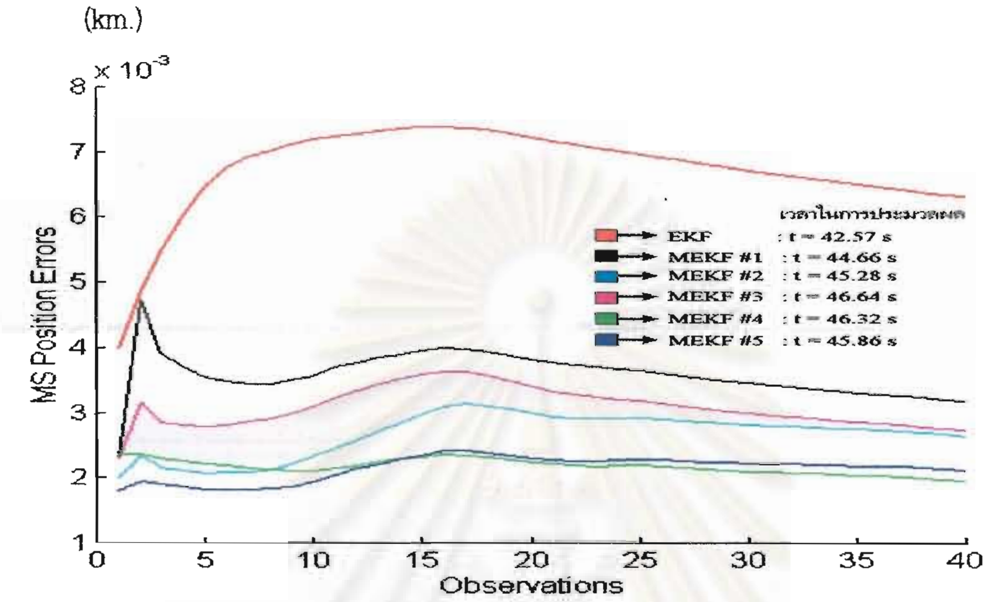


รูปที่ 3.16.1 MS Position error บนแกน Y ที่ได้จากอัลกอริทึม MEKF ประเภทต่างๆ

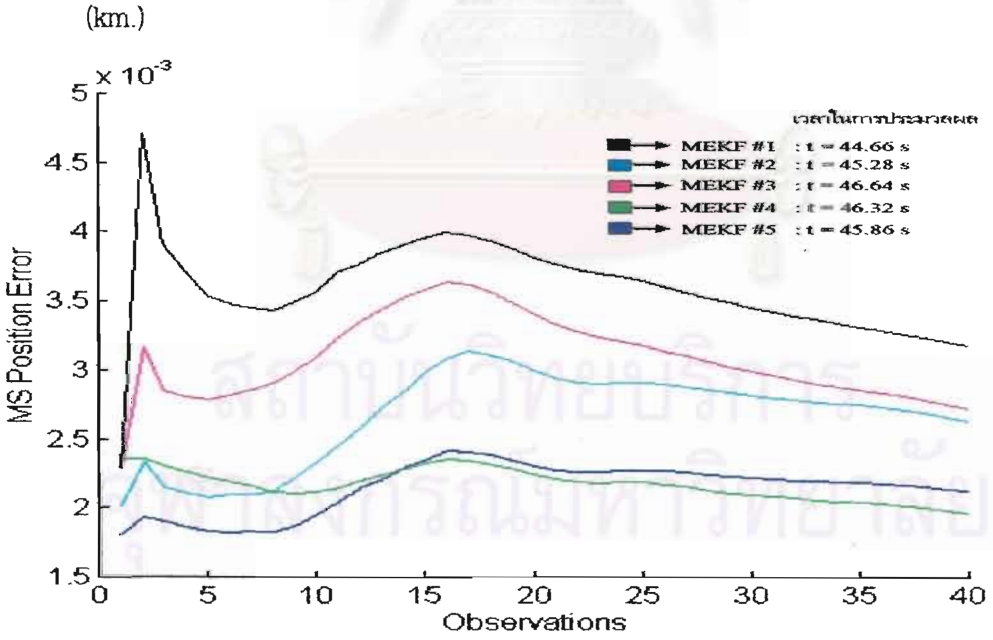
ตารางที่ 3.16 ค่าความคลาดเคลื่อนในรูป Mean square error ของรูปที่ 3.16

Obsv	EKF ($10^{[illegible]}$)	MEKF # 1 ($10^{[illegible]}$)	MEKF # 2 ($10^{[illegible]}$)	MEKF # 3 ($10^{[illegible]}$)	MEKF # 4 ($10^{[illegible]}$)	MEKF # 5 ($10^{[illegible]}$)
1	4.6006	3.2717	3.2022	3.3819	1.5737	1.4800
2	3.5876	2.8174	3.2598	2.3913	1.2468	9.1378
3	3.1820	2.2618	2.5663	2.0686	1.0530	7.1006
4	3.0331	2.0014	2.2105	2.0325	9.2139	5.6410
5	3.0174	1.9293	1.9953	2.1091	8.2221	4.7397
6	3.0433	1.8898	1.8447	2.1622	7.5118	4.1135
7	3.0338	1.8823	1.7509	2.2259	6.9592	3.6726
8	3.0080	1.8191	1.6757	2.2094	6.5304	3.3382
9	3.0000	1.7886	1.6211	2.1962	6.1512	3.0898
10	2.9972	1.7650	1.5122	2.1859	5.8692	2.8809
11	2.9855	1.7599	1.5367	2.2227	5.6333	2.7208
12	2.9739	1.7459	1.4986	2.2575	5.4358	2.5754
13	2.9649	1.7275	1.4653	2.2723	5.2616	2.4556
14	2.9580	1.7145	1.4336	2.2957	5.0992	2.3460
15	2.9485	1.7028	1.4059	2.3038	4.9522	2.2529
16	2.9347	1.6762	1.3853	2.2911	4.8214	2.1700
17	2.9107	1.6369	1.3748	2.2462	4.77121	2.1057
18	2.8795	1.5989	1.3828	2.1866	4.6290	2.0596
19	2.8462	1.5688	1.4034	2.1300	4.5670	2.0371
20	2.8126	1.5467	1.4244	2.0830	4.5176	2.0286
21	2.7830	1.5218	1.4262	2.0452	4.4660	2.0163
22	2.7564	15015	1.4313	2.0182	4.4166	1.9914
23	2.7319	1.4856	1.4299	2.0006	4.3683	1.9720
24	2.7099	1.4731	1.4336	1.9880	4.3221	1.9476
25	2.6888	1.4608	1.4394	1.9728	4.2808	1.9307
26	2.6686	1.4500	1.4487	1.9574	4.2393	1.9155
27	2.6490	1.4391	1.4559	1.9402	4.1977	1.9046
28	2.6297	1.4276	1.4642	1.9205	4.1564	1.8930
29	2.6115	1.4163	1.4679	1.9016	4.1134	1.8830
30	2.5943	1.4051	1.4728	1.8836	4.0708	1.8702
31	2.5782	1.3940	1.4725	1.8677	4.0270	1.8590
32	2.5637	1.3839	1.4721	1.8541	3.9824	1.8437
33	2.5504	1.3736	1.4687	1.8411	3.9372	1.8292
34	2.5380	1.3632	1.4647	1.8280	3.8915	1.8119
35	2.5267	1.3511	1.4605	1.8119	3.8472	1.7956
36	2.5167	1.3392	1.4562	1.7947	3.8043	1.7802
37	2.5068	1.3271	1.4541	1.7762	3.7638	1.7655
38	2.4973	1.3152	1.4504	1.7564	3.7223	1.7518
39	2.4881	1.3023	1.4451	1.7369	3.6796	1.7368
40	2.4791	1.2892	1.4369	1.7173	3.6356	1.7208

กรณีที่ 6 – จำนวน 4 Observers และ Emitter อยู่นอก Observers area



รูปที่ 3.17 MS Position error บนแกน X ที่ได้จากอัลกอริทึม EKF และ MEKF ประเภทต่างๆ

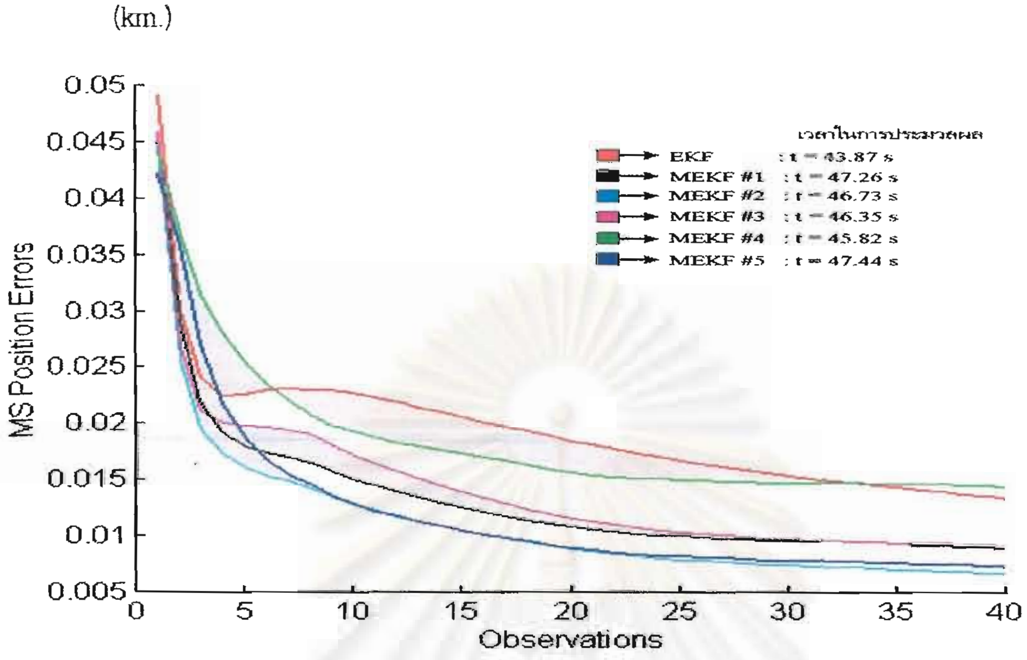


รูปที่ 3.17.1 MS Position error บนแกน X ที่ได้จากอัลกอริทึม MEKF ประเภทต่างๆ

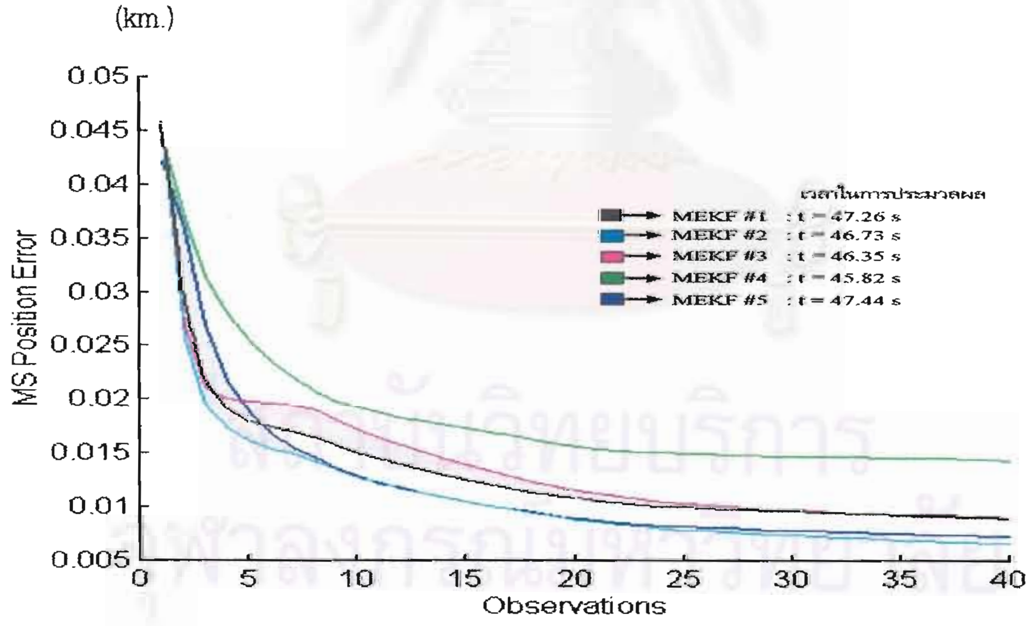
ตารางที่ 3.17 ค่าความคลาดเคลื่อนในรูป Mean square error ของรูปที่ 3.17

Obsv	EKF (10[illegible])	MEKF # 1 (10[illegible])	MEKF # 2 (10[illegible])	MEKF # 3 (10[illegible])	MEKF # 4 (10[illegible])	MEKF # 5 (10[illegible])
1	3.9997	2.2875	2.0134	2.2939	2.3504	1.8103
2	4.8638	4.7168	2.3336	3.1624	2.3488	1.9358
3	5.4669	3.9097	2.1477	2.8420	2.2983	1.8955
4	6.0218	3.7136	2.1015	28021	2.2571	1.8609
5	6.4623	3.5394	2.0733	2.7873	2.2196	1.8306
6	6.7684	3.4829	2.0915	2.8145	2.1844	1.8164
7	6.9316	3.451	2.0922	2.8590	2.1522	1.8249
8	7.0164	3.4335	2.1261	2.9055	2.1208	1.8329
9	7.1120	3.4950	2.2193	2.9880	2.1054	1.8699
10	7.1922	3.5633	2.3282	3.0930	2.1155	1.9470
11	7.2464	3.7075	2.4519	3.2305	2.1471	2.0394
12	7.2905	3.7703	2.5815	3.3471	2.1981	2.1420
13	7.3269	3.8519	2.7302	3.4373	2.2428	2.2085
14	7.3578	3.8994	2.8400	3.5226	2.2834	2.2949
15	7.3793	3.9551	2.9763	3.5773	2.3186	2.3416
16	7.3913	3.9933	3.0848	3.6375	2.3516	2.4116
17	7.3737	3.9770	3.1314	3.6192	2.3436	2.4074
18	7.3329	3.9298	3.1036	3.5583	2.3142	2.3844
19	7.2794	3.8779	3.0630	3.4819	2.2779	2.3441
20	7.2179	3.8127	2.9911	3.3954	2.2404	2.3021

Obsv	EKF (10[illegible])	MEKF # 1 (10[illegible])	MEKF # 2 (10[illegible])	MEKF # 3 (10[illegible])	MEKF # 4 (10[illegible])	MEKF # 5 [illegible]
21	7.1615	3.7673	2.9389	3.3262	2.2080	2.2749
22	7.1085	3.7266	2.9113	3.2713	2.1870	2.2598
23	7.0581	3.7001	2.8993	3.2324	2.1789	2.2614
24	7.0108	3.6737	2.9104	3.2057	2.1869	2.2737
25	6.9619	3.6406	2.9085	3.1728	2.1830	2.2745
26	6.9125	3.6046	2.8954	3.1361	2.1707	2.2683
27	6.8627	3.5639	2.8765	3.0963	2.1522	2.2558
28	6.8126	3.5253	2.8524	3.0569	2.1299	2.2409
29	6.7640	3.4889	2.8303	3.0204	2.1105	2.2286
30	6.7166	3.4548	2.8118	2.9864	2.0941	2.2175
31	6.6706	3.4241	2.7937	2.9562	2.0805	2.2096
32	6.6264	3.3925	2.7787	2.9275	2.0697	2.2017
33	6.5836	3.3632	2.7645	2.9004	2.0584	2.1936
34	6.5418	3.3338	2.7497	2.8737	2.4066	2.1854
35	6.5013	3.3074	2.7381	2.8487	2.0351	2.1779
36	6.4622	3.2838	2.7256	2.8251	2.0233	2.1718
37	6.4230	3.2569	2.7060	2.7992	2.0075	2.1614
38	6.3839	3.2305	2.6821	2.7712	1.9889	2.1467
39	6.3452	3.2028	2.6534	2.7435	1.9687	2.1309
40	6.3068	3.1758	2.6233	2.7151	1.9478	2.1127



รูปที่ 3.18 MSE ระหว่าง EKF และ MEKF ประเภทต่างๆ บนแกน Y



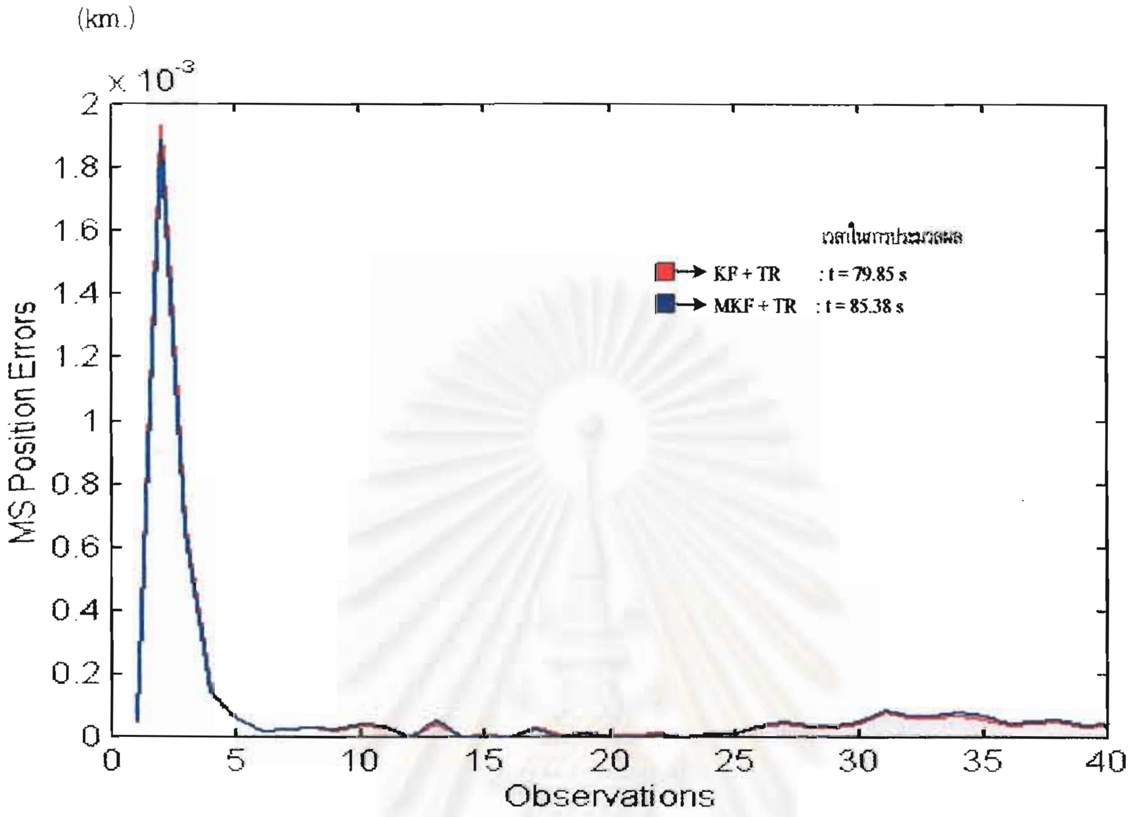
รูปที่ 3.18.1 MSE ระหว่าง MEKF ประเภทต่างๆ บนแกน Y

ตารางที่ 3.18 ค่าความคลาดเคลื่อนในรูป Mean square error ของรูปที่ 3.18

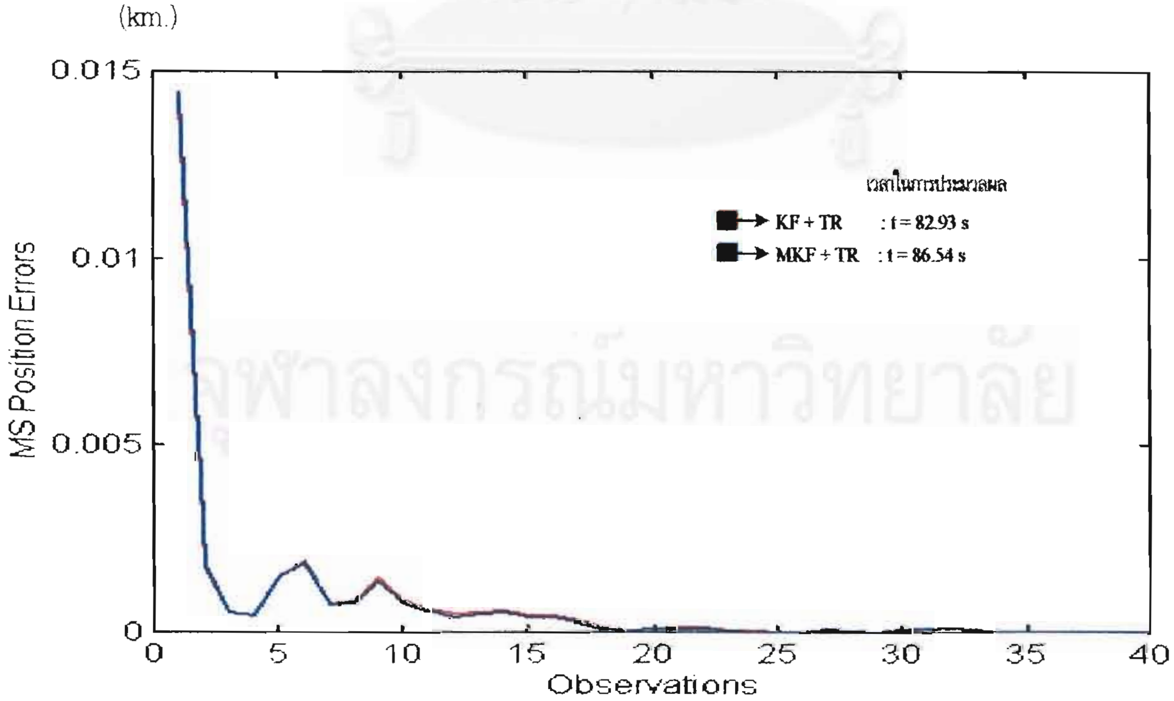
Obsv	EKF (10^{-4})	MEKF # 1 (10^{-5})	MEKF # 2 (10^{-5})	MEKF # 3 (10^{-5})	MEKF # 4 (10^{-5})	MEKF # 5 (10^{-5})
1	4.9030	4.5470	4.4561	4.5773	4.4021	4.2141
2	2.9916	2.9022	2.5541	2.6720	3.6859	3.5684
3	2.4007	2.1874	1.9360	2.1085	3.1420	2.6931
4	2.2452	1.9293	1.7268	2.0103	2.7928	2.1820
5	2.2559	1.7919	1.6042	1.9808	2.5459	1.8763
6	2.2940	1.7365	1.5236	1.9583	2.3540	1.6781
7	2.3043	1.7028	1.4863	1.9380	2.1977	1.5387
8	2.3002	1.6465	1.4096	1.8945	2.0726	1.4470
9	2.2912	1.5785	1.3434	1.8125	1.9839	1.3577
10	2.2663	1.5045	1.2694	1.7201	1.9201	1.2846
11	2.2268	1.4465	1.2219	1.6388	1.8679	1.2190
12	2.1842	1.3908	1.1658	1.5692	1.8252	1.1681
13	2.1409	1.3425	1.1238	1.5013	1.7884	1.1205
14	2.0980	1.2967	1.0822	1.4403	1.7537	1.0807
15	2.0546	1.2534	1.0437	1.3825	1.7213	1.0424
16	2.0109	1.2120	1.0042	1.3272	1.6921	1.0075
17	1.9672	1.1741	0.9702	1.2759	1.6570	0.9743
18	1.9229	1.1385	0.9351	1.2276	1.6211	0.9437
19	1.8801	1.1065	0.9039	1.1847	1.5868	0.9141
20	1.8392	1.0786	0.8760	1.1468	1.5533	0.8882
21	1.8006	1.0548	0.8510	1.1143	1.5290	0.8659
22	1.7639	1.0354	0.8299	1.0878	1.5126	0.8482
23	1.7287	1.0195	0.8115	1.0654	1.5021	0.8341
24	1.6953	1.0071	0.7968	1.0468	1.4975	0.8225
25	1.6634	0.9969	0.7837	1.0313	1.4921	0.8132
26	1.6333	0.9881	0.7722	1.0171	1.4863	0.8046
27	1.6047	0.9799	0.7612	1.0042	1.4800	0.7971
28	1.5773	0.9721	0.7511	0.9919	1.4737	0.7897
29	1.5513	0.9653	0.7417	0.9814	1.4690	0.7836
30	1.5264	0.9596	0.7328	0.9714	1.4656	0.7779
31	1.5026	0.9532	0.7238	0.9615	1.4626	0.7725
32	1.4799	0.9465	0.7475	0.9510	1.4600	0.7664
33	1.4583	0.9394	0.7058	0.9404	1.4570	0.7603
34	1.4374	0.9327	0.6974	0.9304	1.4538	0.7540
35	1.4175	0.9272	0.6903	0.9217	1.4510	0.7480
36	1.3984	0.9232	0.6838	0.9150	1.4485	0.7436
37	1.3801	0.9193	0.6776	0.9086	1.4444	0.7389
38	1.3626	0.9144	0.6706	0.9013	1.4388	0.7341
39	1.3458	0.9081	0.6629	0.8929	1.4317	0.7285
40	1.3298	0.9003	0.6545	0.8833	1.4234	0.7218

แสดงผลการทดลองใน Block ที่ 3 (MKF + TR) ทั้ง 6 กรณี

<u>กรณีที่ 1 – จำนวน 2 Observers และ Emitter อยู่ใน Observers area</u>



รูปที่ 3.19 การเปรียบเทียบ MS Position error ระหว่าง KF+TR และ MKF+TR บนแกน X



รูปที่ 3.20 การเปรียบเทียบ MS Position error ระหว่าง KF+TR และ MKF+TR บนแกน Y

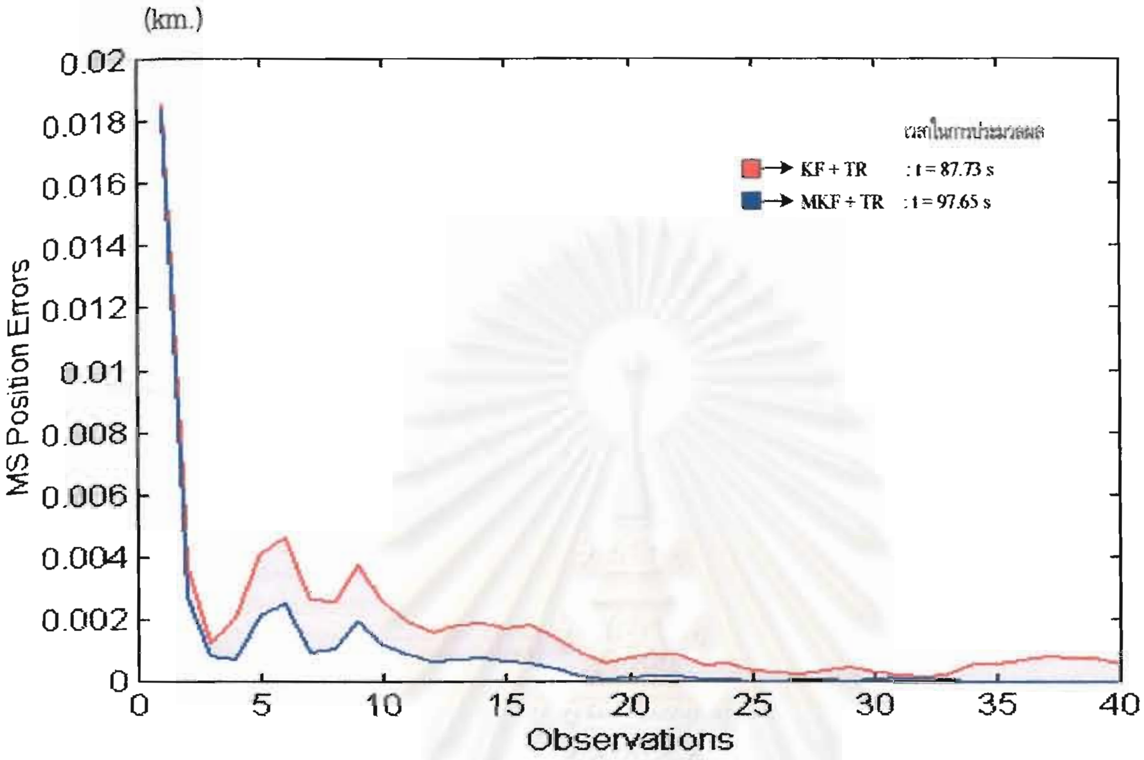
ตารางที่ 3.19 ค่าความคลาดเคลื่อนในรูป Mean square error ของรูปที่ 3.19

Obsv	KF+TR $(10)^{-3}$	MKF+TR $(10)^{-3}$	Obsv	KF+TR $(10)^{-3}$	MKF+TR $(10)^{-3}$	Obsv	KF+TR $(10)^{-3}$	MKF+TR $(10)^{-3}$	Obsv	KF+TR $(10)^{-3}$	MKF+TR $(10)^{-3}$
1	0.0492	0.0520	11	0.0273	0.0330	21	0.0051	0.0025	31	0.7495	0.0844
2	1.9273	1.8844	12	0.0000	0.0000	22	0.0090	0.0054	32	0.0563	0.0645
3	0.6673	0.6481	13	0.0428	0.0504	23	0.0009	0.0000	33	0.0572	0.0651
4	0.1443	0.1391	14	0.0014	0.0027	24	0.0076	0.0117	34	0.0672	0.0769
5	0.0582	0.0567	15	0.0046	0.0028	25	0.0092	0.0134	35	0.0548	0.0457
6	0.0171	0.0160	16	0.0015	0.0004	26	0.0304	0.0372	36	0.0337	0.0400
7	0.0229	0.0239	17	0.0276	0.0221	27	0.0408	0.0473	37	0.0423	0.0492
8	0.0292	0.0321	18	0.0039	0.0019	28	0.0291	00344	38	0.0463	0.0532
9	0.0159	0.0208	19	0.0144	0.0100	29	0.0313	0.0375	39	0.0295	0.0346
10	0.0323	0.0388	20	0.0041	0.0020	30	0.0383	0.0450	40	0.0359	0.0419

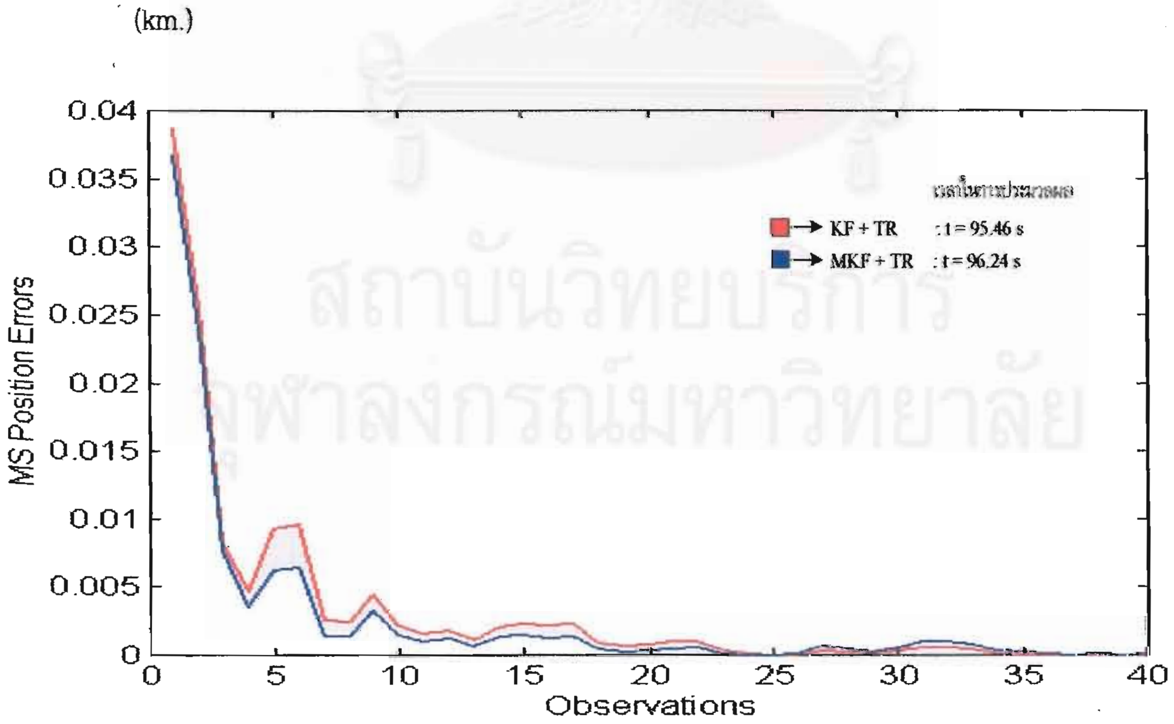
ตารางที่ 3.20 ค่าความคลาดเคลื่อนในรูป Mean square error ของรูปที่ 3.20

Obsv	KF+TR $(10)^{-3}$	MKF+TR $(10)^{-3}$	Obsv	KF+TR $(10)^{-3}$	MKF+TR $(10)^{-3}$	Obsv	KF+TR $(10)^{-3}$	MKF+TR $(10)^{-3}$	Obsv	KF+TR $(10)^{-3}$	MKF+TR $(10)^{-3}$
1	14.224	14.140	11	0.6128	0.0557	21	0.1356	0.1091	31	0.0706	0.0911
2	1.7353	1.7725	12	0.4708	0.4137	22	0.1271	0.1003	32	0.0889	0.0758
3	0.5093	0.5122	13	0.5187	0.4630	23	0.0379	0.0223	33	0.0625	0.0459
4	0.4533	0.4511	14	0.5900	0.5359	24	0.0317	0.0197	34	0.0013	0.0008
5	1.5148	1.4821	15	0.4586	0.4167	25	0.0006	0.0000	35	0.0000	0.0000
6	1.8761	1.8532	16	0.4283	0.3817	26	0.0010	0.0039	36	0.0039	0.0004
7	0.7528	0.7379	17	0.2896	0.2534	27	0.0630	0.0809	37	0.0020	0.0047
8	0.8343	0.7019	18	0.1217	0.0961	28	0.0335	0.0458	38	0.0082	0.0025
9	1.4297	1.3488	19	0.3860	0.0252	29	0.0112	0.0189	39	0.0057	0.0042
10	0.8573	0.7863	20	0.9221	0.0702	30	0.0532	0.0699	40	0.0003	0.0001

<u>กรณีที่ 2 - จำนวน 2 Observers และ Emitter อยู่นอก Observers area</u>



รูปที่ 3.21 การเปรียบเทียบ MS Position error ระหว่าง KF+TR และ MKF+TR บนแกน X



รูปที่ 3.22 การเปรียบเทียบ MS Position error ระหว่าง KF+TR และ MKF+TR บนแกน Y

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

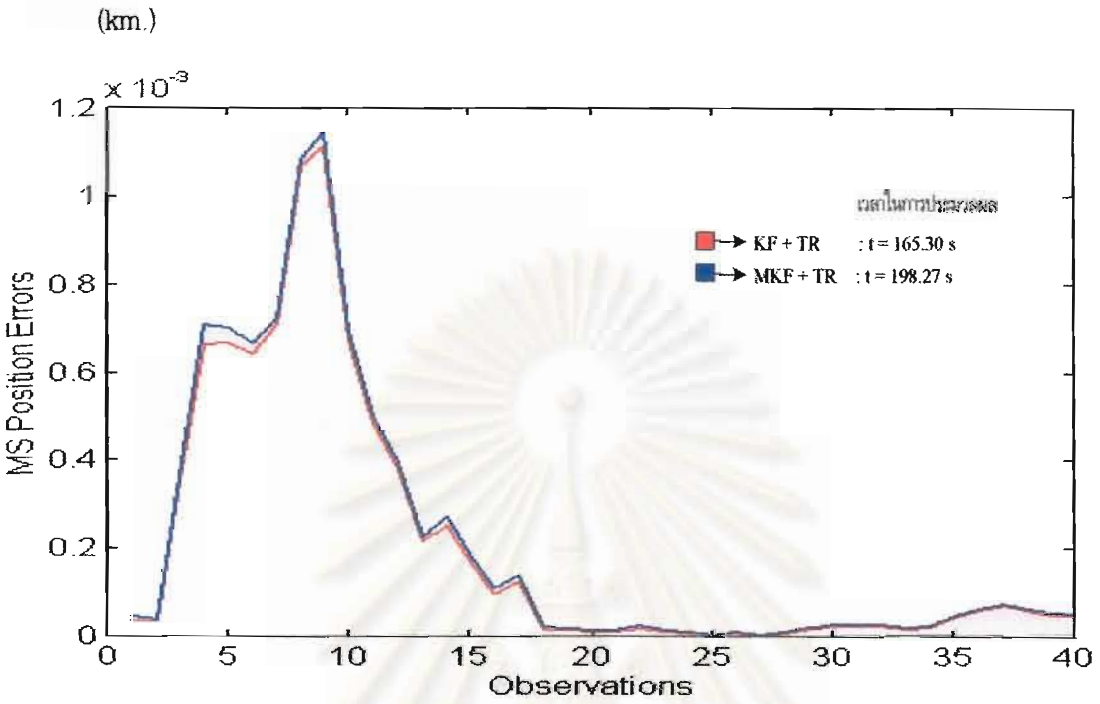
ตารางที่ 3.21 ค่าความคลาดเคลื่อนในรูป Mean square error ของรูปที่ 3.21

Obsv	KF+TR $(10)^{-3}$	MKF+TR $(10)^{-3}$	Obsv	KF+TR $(10)^{-3}$	MKF+TR $(10)^{-3}$	Obsv	KF+TR $(10)^{-3}$	MKF+TR $(10)^{-3}$	Obsv	KF+TR $(10)^{-3}$	MKF+TR $(10)^{-3}$
1	18.504	18.363	11	1.9392	0.8598	21	0.8589	0.2010	31	0.2346	0.0947
2	3.6920	2.7214	12	1.5894	0.6665	22	0.9024	0.1944	32	0.1785	0.1170
3	1.2545	0.8380	13	1.8180	0.6993	23	0.5367	0.0759	33	0.2601	0.0837
4	2.0321	0.6830	14	1.8741	0.7855	24	0.5652	0.0634	34	0.5629	0.0000
5	4.1002	2.0825	15	1.7153	0.6230	25	0.3403	0.0057	35	0.6070	0.0000
6	4.6437	2.5227	16	1.8395	0.5828	26	0.2738	0.0000	36	0.7294	0.0009
7	2.6208	0.9423	17	1.4368	0.4086	27	0.2405	0.0752	37	0.8243	0.0223
8	2.5693	1.0350	18	0.9401	0.1766	28	0.3629	0.0377	38	0.7461	0.0157
9	3.7765	1.1916	19	0.6111	0.0637	29	0.4890	0.0106	39	0.7422	0.0209
10	2.5756	1.1923	20	0.7623	0.1405	30	0.2772	0.0632	40	0.5681	0.0032

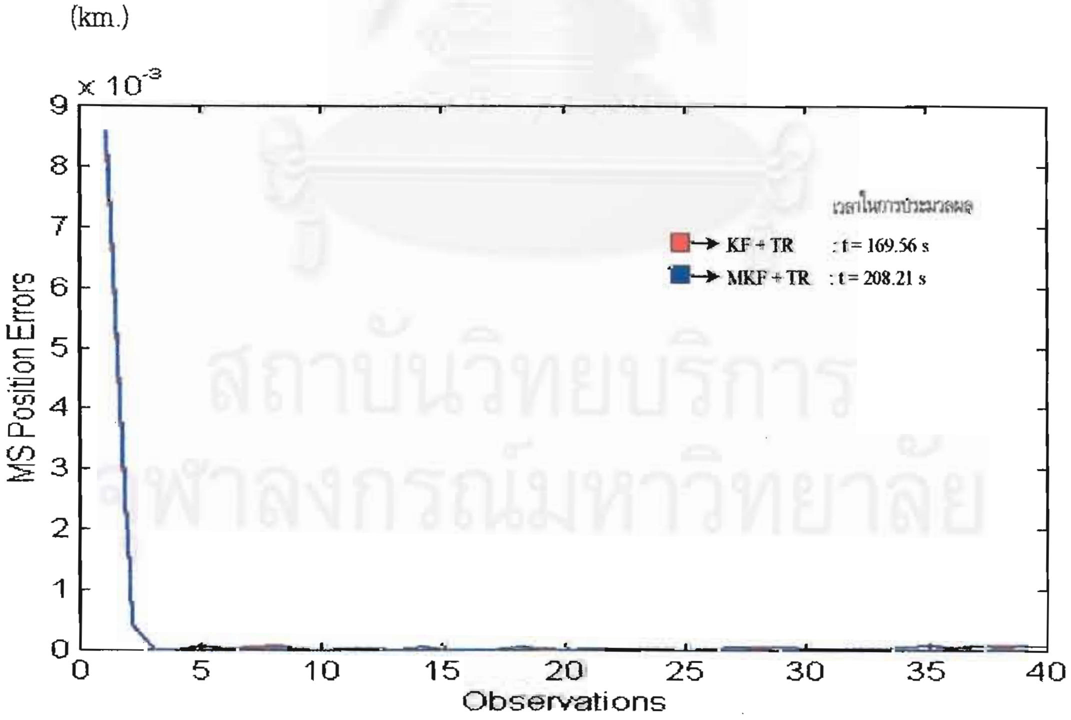
ตารางที่ 3.22 ค่าความคลาดเคลื่อนในรูป Mean square error ของรูปที่ 3.22

Obsv	KF+TR $(10)^{-3}$	MKF+TR $(10)^{-3}$	Obsv	KF+TR $(10)^{-3}$	MKF+TR $(10)^{-3}$	Obsv	KF+TR $(10)^{-3}$	MKF+TR $(10)^{-3}$	Obsv	KF+TR $(10)^{-3}$	MKF+TR $(10)^{-3}$
1	38.767	36.749	11	1.5893	1.0272	21	1.0110	0.5226	31	0.5794	1.0342
2	25.443	23.791	12	1.8400	1.2511	22	1.0797	0.5541	32	0.5738	1.0080
3	8.2237	7.5991	13	1.1572	0.6486	23	0.3322	0.0833	33	0.4475	0.5510
4	4.7037	3.5085	14	2.1476	1.4444	24	0.1627	0.0122	34	0.0712	0.0945
5	9.2195	6.2079	15	2.3763	1.5572	25	0.0076	0.0241	35	0.0443	0.0567
6	9.6372	6.4932	16	2.2455	1.3110	26	0.0293	0.1553	36	0.0009	0.0015
7	2.5285	1.3649	17	2.3100	1.4060	27	0.3828	0.6977	37	0.0071	0.0070
8	2.4751	1.4576	18	0.9333	0.4436	28	0.1896	0.4297	38	0.0006	0.0006
9	4.4226	3.2418	19	0.6545	0.2639	29	0.0792	0.2838	39	0.0155	0.0145
10	2.2528	1.5069	20	0.7755	0.3586	30	0.2940	0.6377	40	0.0024	0.0024

กรณีที่ 3 – จำนวน 3 Observers และ Emitter อยู่ใน Observers area



รูปที่ 3.23 การเปรียบเทียบ MS Position error ระหว่าง KF+TR และ MKF+TR บนแกน X



รูปที่ 3.24 การเปรียบเทียบ MS Position error ระหว่าง KF+TR และ MKF+TR บนแกน Y

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

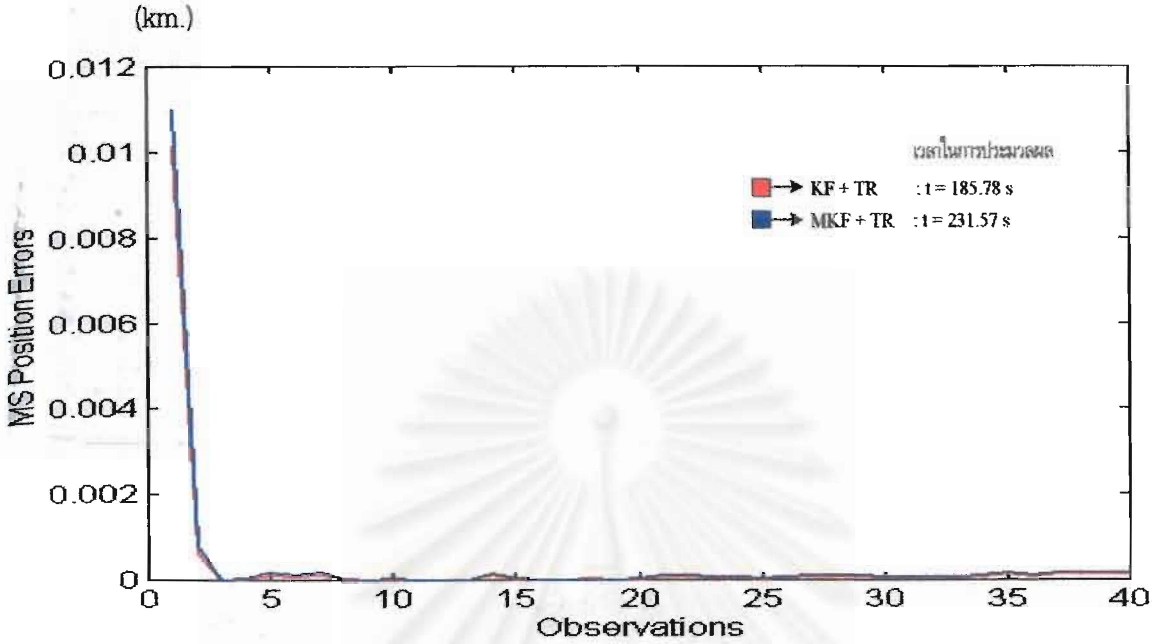
ตารางที่ 3.23 ค่าความคลาดเคลื่อนในรูป Mean square error ของรูปที่ 3.23

Obsv	KF+TR $(10)^{-3}$	MKF+TR $(10)^{-3}$	Obsv	KF+TR $(10)^{-3}$	MKF+TR $(10)^{-3}$	Obsv	KF+TR $(10)^{-3}$	MKF+TR $(10)^{-3}$	Obsv	KF+TR $(10)^{-3}$	MKF+TR $(10)^{-3}$
1	0.0345	0.0440	11	0.4812	0.49898	21	0.0105	0.0136	31	0.0221	0.0252
2	0.0341	0.0377	12	0.3802	0.3937	22	0.0193	0.0235	32	0.0211	0.0242
3	0.3533	0.3612	13	0.2149	0.2261	23	0.0109	0.0136	33	0.0153	0.0179
4	0.6612	0.7089	14	0.2503	0.2714	24	0.0082	0.0105	34	0.0173	0.0200
5	0.6664	0.6997	15	0.1697	0.1868	25	0.0032	0.0042	35	0.0422	0.0415
6	0.6395	0.6663	16	0.0958	0.1010	26	0.0080	0.0102	36	0.0555	0.0548
7	0.7067	0.7206	17	0.1237	0.1380	27	0.0110	0.0115	37	0.0669	0.0715
8	1.0661	1.0842	18	0.0148	0.0196	28	0.0084	0.0107	38	0.0561	0.0541
9	1.1128	1.1421	19	0.0134	0.0176	29	0.0183	0.0087	39	0.0474	0.0511
10	0.6724	0.6931	20	0.0106	0.0140	30	0.0228	0.0261	40	0.0464	0.0500

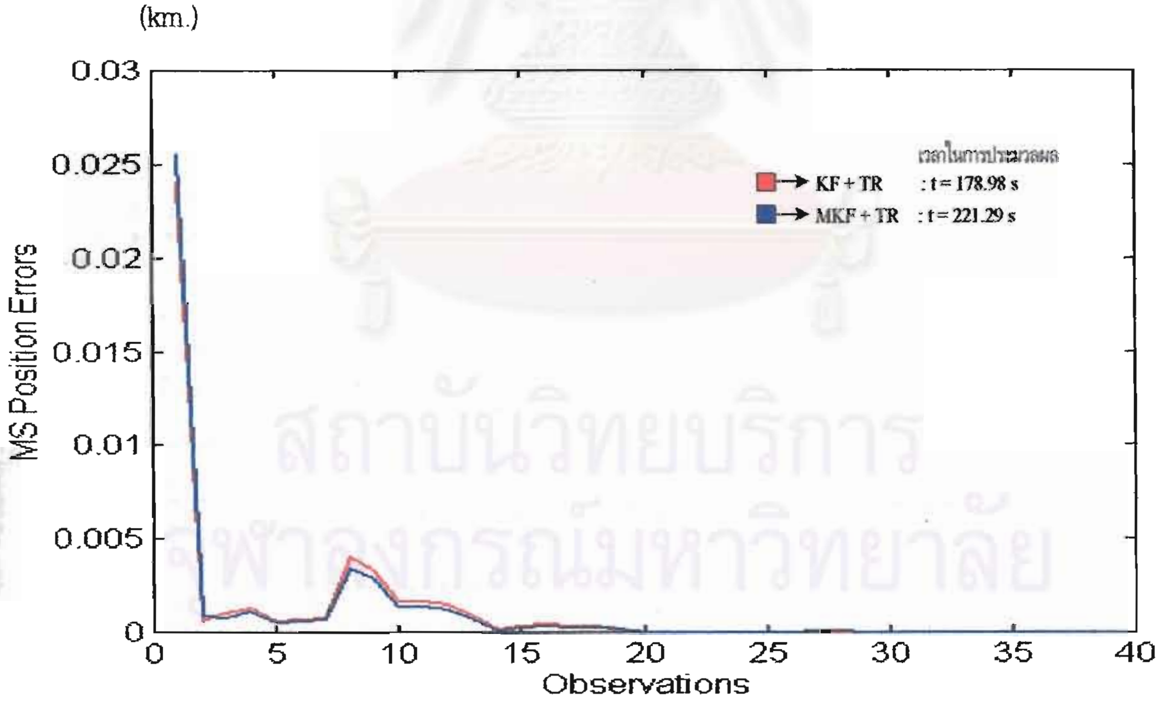
ตารางที่ 3.24 ค่าความคลาดเคลื่อนในรูป Mean square error ของรูปที่ 3.24

Obsv	KF+TR $(10)^{-3}$	MKF+TR $(10)^{-3}$	Obsv	KF+TR $(10)^{-3}$	MKF+TR $(10)^{-3}$	Obsv	KF+TR $(10)^{-3}$	MKF+TR $(10)^{-3}$	Obsv	KF+TR $(10)^{-3}$	MKF+TR $(10)^{-3}$
1	8.4080	8.5794	11	0.0167	0.0145	21	0.0289	0.0264	31	0.0226	0.0207
2	0.3864	0.4133	12	0.0246	0.0231	22	0.0275	0.0249	32	0.0294	0.0270
3	0.0039	0.0040	13	0.0110	0.0110	23	0.0278	0.0229	33	0.0280	0.0250
4	0.0027	0.0009	14	0.0416	0.0443	24	0.0211	0.0187	34	0.0477	0.0433
5	0.0662	0.0567	15	0.0316	0.0036	25	0.0046	0.0036	35	0.0757	0.0704
6	0.0365	0.0314	16	0.0413	0.0038	26	0.0158	0.0140	36	0.0396	0.0363
7	0.0422	0.0381	17	0.0878	0.0008	27	0.0459	0.0432	37	0.0727	0.0681
8	0.0700	0.0617	18	0.0412	0.0415	28	0.0419	0.0392	38	0.0695	0.0645
9	0.0220	0.0198	19	0.0123	0.0135	29	0.0429	0.0403	39	0.0806	0.0760
10	0.0106	0.0005	20	0.0398	0.0031	30	0.0158	0.0139	40	0.0765	0.0721

<u>กรณีที่ 4 - จำนวน 3 Observers และ Emitter อยู่นอก Observers area</u>



รูปที่ 3.25 การเปรียบเทียบ MS Position error ระหว่าง KF+TR และ MKF+TR บนแกน X



รูปที่ 3.26 การเปรียบเทียบ MS Position error ระหว่าง KF+TR และ MKF+TR บนแกน Y

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

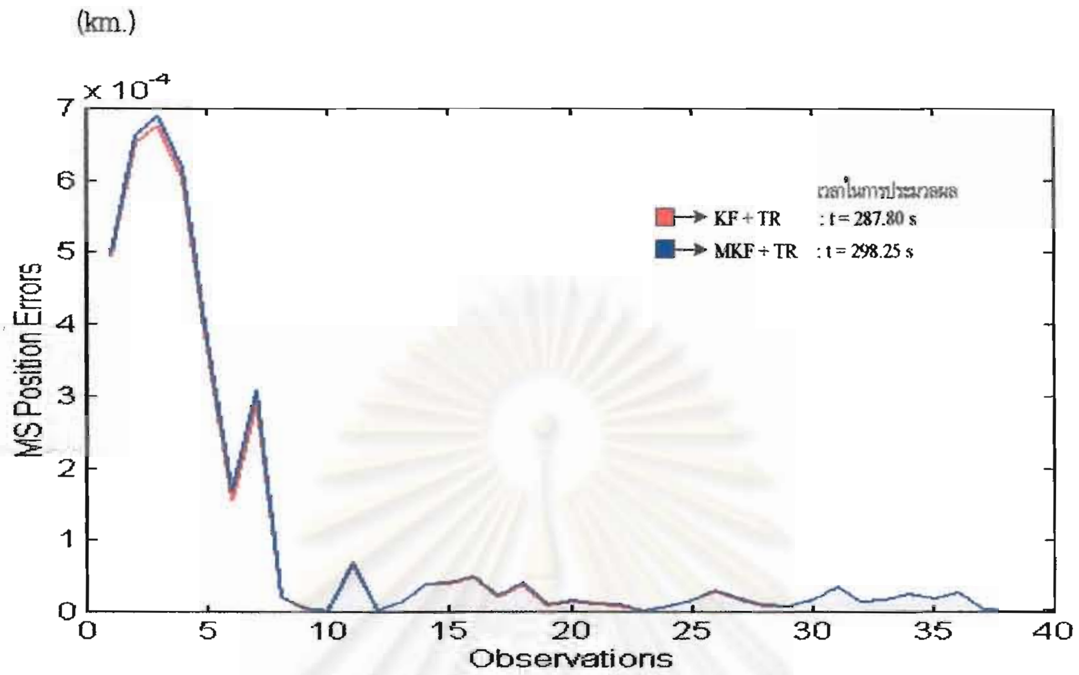
ตารางที่ 3.25 ค่าความคลาดเคลื่อนในรูป Mean square error ของรูปที่ 3.25

Obsv	KF+TR $(10)^{-3}$	MKF+TR $(10)^{-3}$	Obsv	KF+TR $(10)^{-3}$	MKF+TR $(10)^{-3}$	Obsv	KF+TR $(10)^{-3}$	MKF+TR $(10)^{-3}$	Obsv	KF+TR $(10)^{-3}$	MKF+TR $(10)^{-3}$
1	10.131	10.994	11	0.0041	0.0008	21	0.0568	0.0941	31	0.0374	0.0607
2	0.5883	0.7731	12	0.0108	0.0007	22	0.0588	0.0942	32	0.0451	0.0719
3	0.0097	0.0007	13	0.0043	0.0000	23	0.0488	0.0800	33	0.0452	0.0709
4	0.0072	0.0263	14	0.0920	0.1514	24	0.0359	0.0614	34	0.0744	0.1058
5	0.1080	0.1701	15	0.0169	0.0407	25	0.0096	0.0238	35	0.1316	0.1719
6	0.0775	0.1214	16	0.0002	0.0030	26	0.0297	0.0531	36	0.0752	0.1092
7	0.1197	0.1672	17	0.0014	0.0103	27	0.0718	0.1116	37	0.1268	0.1679
8	0.0401	0.0082	18	0.0385	0.0173	28	0.0723	0.1108	38	0.1254	0.1637
9	0.0041	0.0013	19	0.0043	0.0003	29	0.0766	0.1143	39	0.1432	0.1838
10	0.0046	0.0274	20	0.0168	0.0367	30	0.0309	0.0552	40	0.1363	0.1857

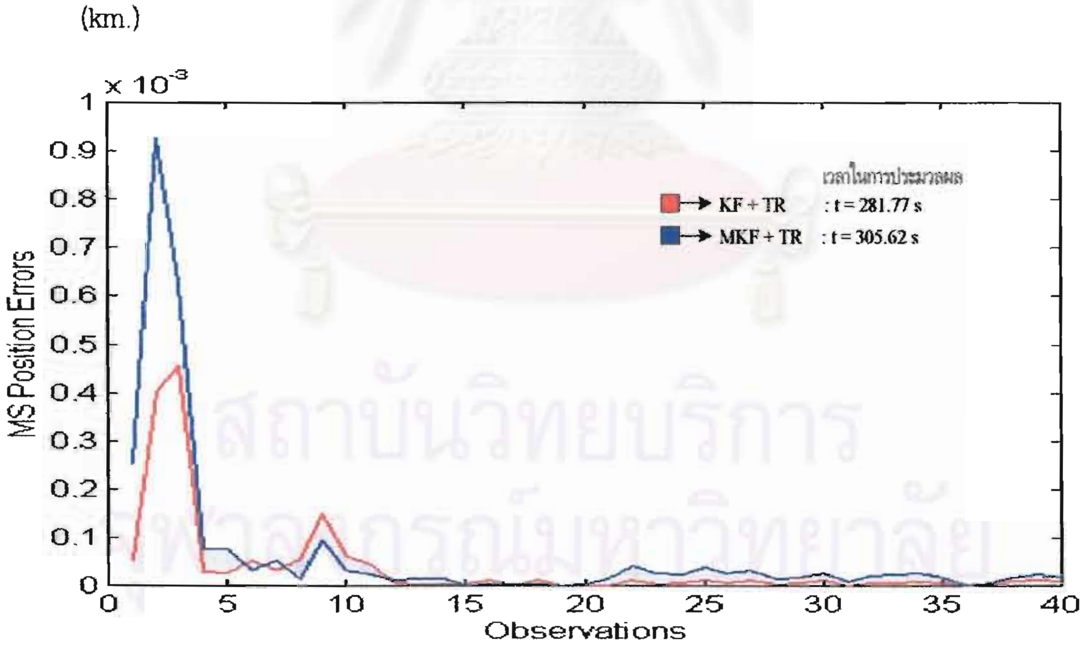
ตารางที่ 3.26 ค่าความคลาดเคลื่อนในรูป Mean square error ของรูปที่ 3.26

Obsv	KF+TR $(10)^{-3}$	MKF+TR $(10)^{-3}$	Obsv	KF+TR $(10)^{-3}$	MKF+TR $(10)^{-3}$	Obsv	KF+TR $(10)^{-3}$	MKF+TR $(10)^{-3}$	Obsv	KF+TR $(10)^{-3}$	MKF+TR $(10)^{-3}$
1	24.037	25.519	11	1.6799	1.3688	21	0.0045	0.0199	31	0.0000	0.0000
2	0.6334	0.8924	12	1.5138	1.2684	22	0.0002	0.0066	32	0.0015	0.0093
3	1.0699	0.8284	13	0.8618	0.6940	23	0.0031	0.0146	33	0.0033	0.0129
4	1.3426	1.1826	14	0.1813	0.1088	24	0.0028	0.0135	34	0.0200	0.0373
5	0.6225	0.5056	15	0.3150	0.2309	25	0.0012	0.0007	35	0.0209	0.0221
6	0.7450	0.6337	16	0.4048	0.3114	26	0.0006	0.0072	36	0.0013	0.0003
7	0.7733	0.6729	17	0.3943	0.3074	27	0.0662	0.1074	37	0.0039	0.0034
8	4.0063	3.3898	18	0.3859	0.3048	28	0.0252	0.0515	38	0.0066	0.0071
9	3.3458	2.8766	19	0.1973	0.1557	29	0.0115	0.0300	39	0.0208	0.0186
10	1.6947	1,3640	20	0.0156	0.0360	30	0.0035	0.0000	40	0.0183	0.0180

กรณีที่ 5 – จำนวน 4 Observers และ Emitter อยู่ใน Observers area



รูปที่ 3.27 การเปรียบเทียบ MS Position error ระหว่าง KF+TR และ MKF+TR บนแกน X



รูปที่ 3.28 การเปรียบเทียบ MS Position error ระหว่าง KF+TR และ MKF+TR บนแกน Y

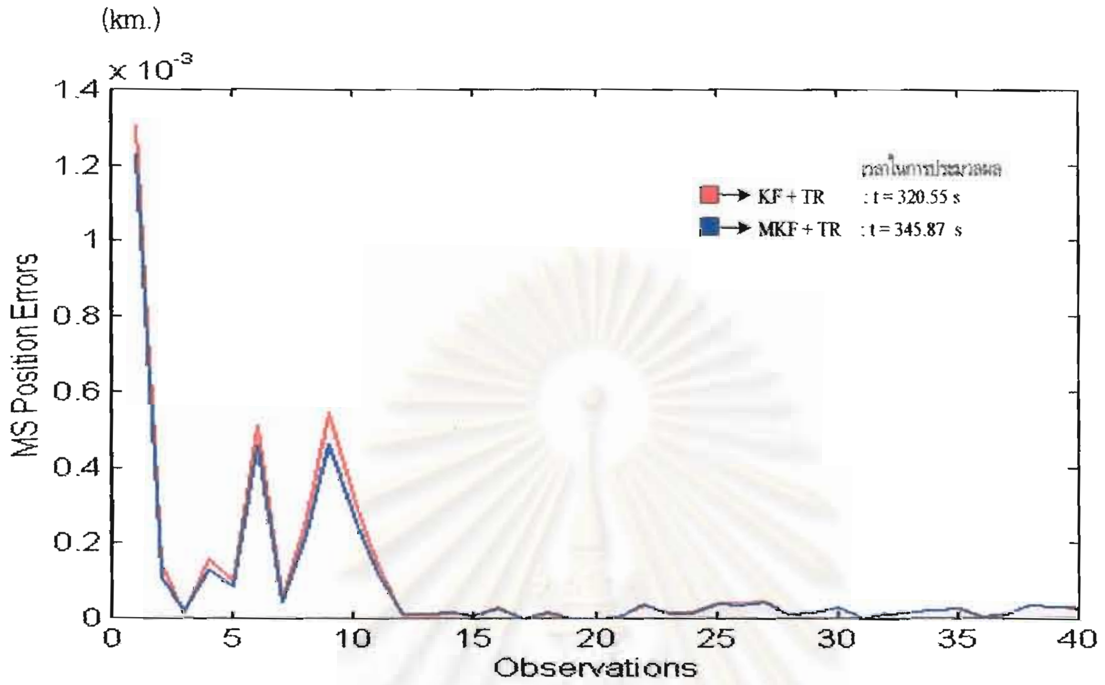
ตารางที่ 3.27 ค่าความคลาดเคลื่อนในรูป Mean square error ของรูปที่ 3.27

Obsv	KF+TR $(10)^{-3}$	MKF+TR $(10)^{-3}$	Obsv	KF+TR $(10)^{-3}$	MKF+TR $(10)^{-3}$	Obsv	KF+TR $(10)^{-3}$	MKF+TR $(10)^{-3}$	Obsv	KF+TR $(10)^{-3}$	MKF+TR $(10)^{-3}$
1	0.4980	0.4947	11	0.0672	0.0696	21	0.0119	0.0102	31	0.0359	0.0355
2	0.6622	0.6530	12	0.0021	0.0021	22	0.0104	0.0088	32	0.0149	0.0147
3	0.6891	0.6747	13	0.0140	0.0138	23	0.0020	0.0014	33	0.0162	0.0160
4	0.6171	0.6027	14	0.0393	0.0384	24	0.0088	0.0075	34	0.0249	0.0249
5	0.3786	0.3683	15	0.0410	0.0387	25	0.0174	0.0156	35	0.0190	0.0183
6	0.1691	0.1561	16	0.0493	0.0466	26	0.0301	0.0278	36	0.0260	0.0260
7	0.3089	0.2911	17	0.0219	0.0199	27	0.0178	0.0159	37	0.0040	0.0043
8	0.0214	0.0212	18	0.0403	0.0375	28	0.0095	0.0081	38	0.0018	0.0010
9	0.0057	0.0050	19	0.0094	0.0078	29	0.0090	0.0078	39	0.0019	0.0018
10	0.0014	0.0012	20	0.0159	0.0138	30	0.0173	0.0159	40	0.0045	0.0045

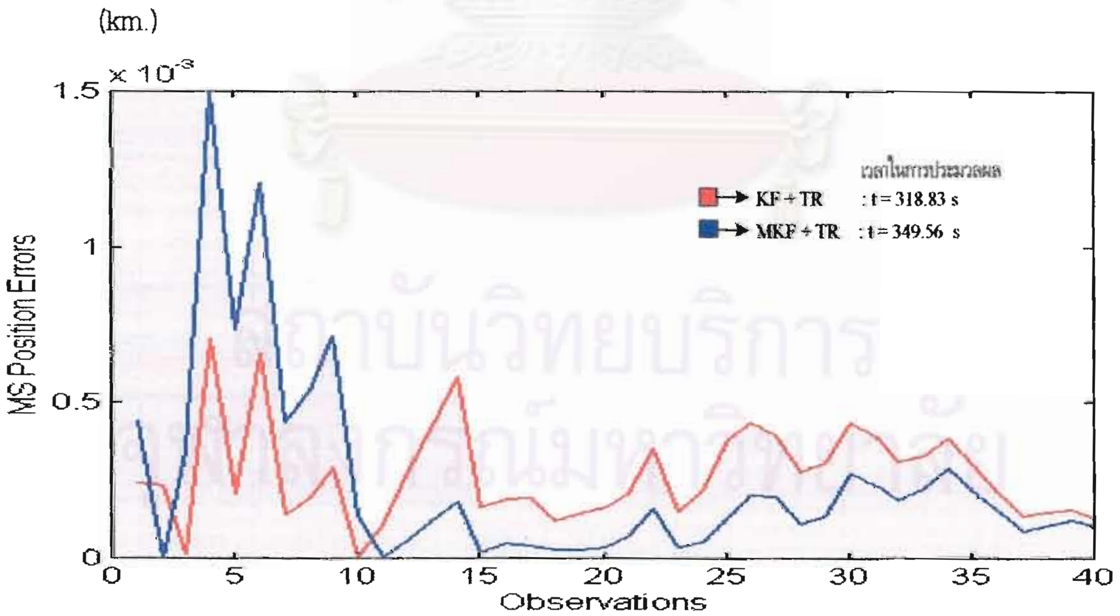
ตารางที่ 3.28 ค่าความคลาดเคลื่อนในรูป Mean square error ของรูปที่ 3.28

Obsv	KF+TR $(10)^{-3}$	MKF+TR $(10)^{-3}$	Obsv	KF+TR $(10)^{-3}$	MKF+TR $(10)^{-3}$	Obsv	KF+TR $(10)^{-3}$	MKF+TR $(10)^{-3}$	Obsv	KF+TR $(10)^{-3}$	MKF+TR $(10)^{-3}$
1	0.2523	0.0533	11	0.0226	0.0451	21	0.0155	0.0012	31	0.0093	0.0006
2	0.9263	0.4021	12	0.0118	0.0047	22	0.0406	0.0112	32	0.0193	0.0055
3	0.6234	0.4550	13	0.0146	0.0044	23	0.0255	0.0043	33	0.0239	0.0072
4	0.0770	0.0284	14	0.0134	0.0038	24	0.0249	0.0046	34	0.0267	0.0084
5	0.0775	0.0277	15	0.0000	0.0043	25	0.0392	0.0124	35	0.0167	0.0047
6	0.0316	0.0539	16	0.0040	0.0127	26	0.0253	0.0067	36	0.0012	0.0002
7	0.0530	0.0318	17	0.0016	0.0007	27	0.0317	0.0117	37	0.0071	0.0020
8	0.0159	0.0546	18	0.0029	0.0126	28	0.0139	0.0035	38	0.0167	0.0084
9	0.0927	0.1501	19	0.0013	0.0011	29	0.0162	0.0045	39	0.0240	0.0121
10	0.0326	0.0603	20	0.0021	0.0007	30	0.0273	0.0108	40	0.0168	0.0086

กรณีที่ 6 – จำนวน 4 Observers และ Emitter อยู่นอก Observers area



รูปที่ 3.29 การเปรียบเทียบ MS Position error ระหว่าง KF+TR และ MKF+TR บนแกน X



รูปที่ 3.30 การเปรียบเทียบ MS Position error ระหว่าง KF+TR และ MKF+TR บนแกน Y

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ตารางที่ 3.29 ค่าความคลาดเคลื่อนในรูป Mean square error ของรูปที่ 3.29

Obsv	KF+TR $(10)^{-3}$	MKF+TR $(10)^{-3}$
1	1.3071	1.2277
2	0.1408	0.1048
3	0.0186	0.0184
4	0.1553	0.1269
5	0.0982	0.0803
6	0.5092	0.4543
7	0.0512	0.0407
8	0.2650	0.2246
9	0.5469	0.4594
10	0.3254	0.2715
11	0.1447	0.1188
12	0.0122	0.0119
13	0.0112	0.0105
14	0.0131	0.0178
15	0.0092	0.0073
16	0.0298	0.0243
17	0.0000	0.0000
18	0.0148	0.0149
19	0.0000	0.0000
20	0.0000	0.0000
21	0.0036	0.0042
22	0.0343	0.0361
23	0.0146	0.0121
24	0.0155	0.0138
25	0.0429	0.0380
26	0.0393	0.0342
27	0.0459	0.0418
28	0.0137	0.0118
29	0.0175	0.0152
30	0.0274	0.0267
31	0.0009	0.0007
32	0.0117	0.0113
33	0.0182	0.0171
34	0.0217	0.0216
35	0.0284	0.0261
36	0.0028	0.0025
37	0.0133	0.0122
38	0.0335	0.0344
39	0.0292	0.0207
40	0.0258	0.0240

ตารางที่ 3.30 ค่าความคลาดเคลื่อนในรูป Mean square error ของรูปที่ 3.30

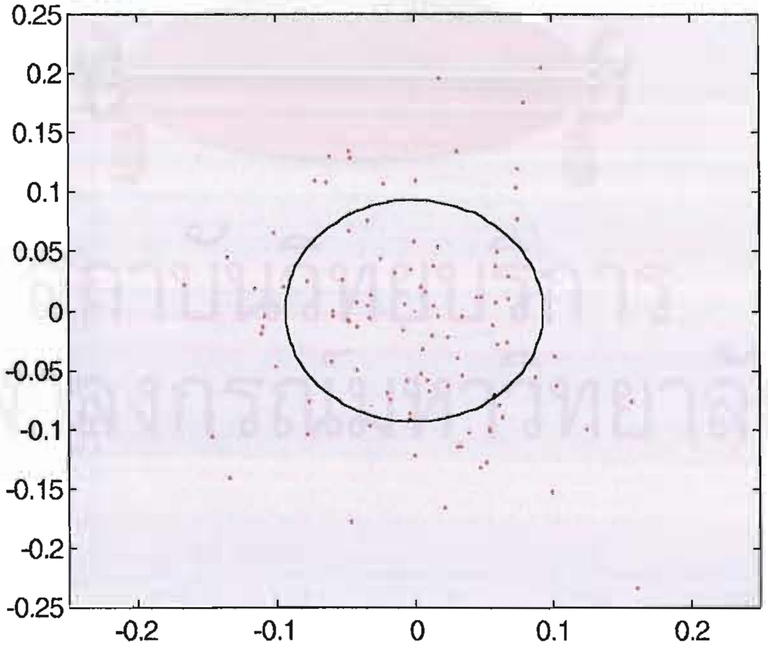
Obsv	KF+TR $(10)^{-3}$	MKF+TR $(10)^{-3}$
1	0.2413	0.4387
2	0.2330	0.0001
3	0.0129	0.3427
4	0.7027	1.4948
5	0.2057	0.7334
6	0.6553	1.2068
7	0.1418	0.4372
8	0.1933	0.5400
9	0.2888	0.7131
10	0.0051	0.1358
11	0.1087	0.0045
12	0.2765	0.0572
13	0.4284	0.1221
14	0.5794	0.1789
15	0.1606	0.0165
16	0.1902	0.0463
17	0.1939	0.0378
18	0.1182	0.0242
19	0.1445	0.0263
20	0.1635	0.0362
21	0.2082	0.0724
22	0.3520	0.1574
23	0.1493	0.0370
24	0.2188	0.0523
25	0.3810	0.1336
26	0.4373	0.2032
27	0.3944	0.1978
28	0.2770	0.1119
29	0.3062	0.1357
30	0.4347	0.2747
31	0.3898	0.2341
32	0.3075	0.1846
33	0.3258	0.2202
34	0.3863	0.2867
35	0.2929	0.2107
36	0.2057	0.1478
37	0.1316	0.0826
38	0.1457	0.0996
39	0.1554	0.1167
40	0.1243	0.0994

การแสดงผลการทดสอบในรูป Circular Error Probability

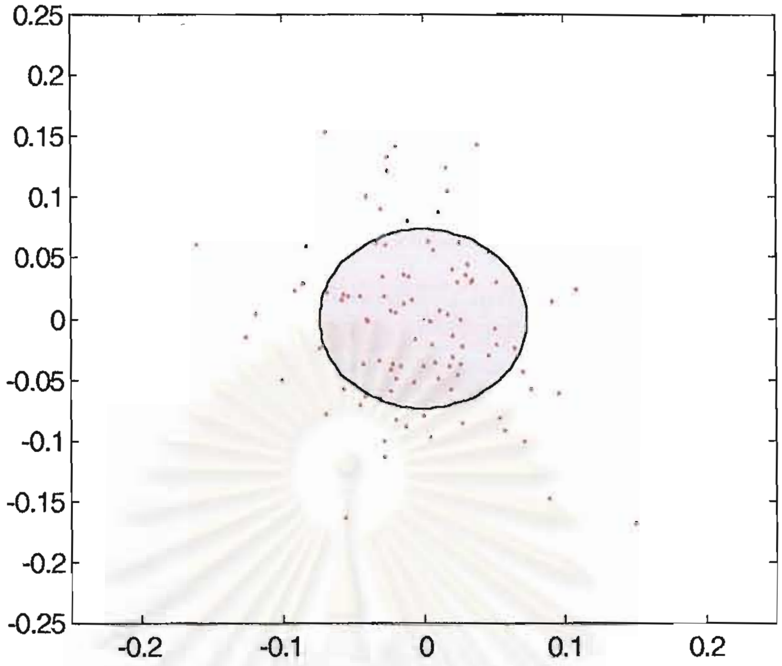
การแสดงผลการทดสอบในรูป Circular Error Probability (CEP) 50 % โดยการนำค่าระยะทางจากจุดของตำแหน่งจริง ถึงจุดของตำแหน่งที่หาได้ มาจัดลำดับทั้งหมด 100 ค่า เนื่องมาจากการทำ 100 Monte Carlo runs โดยค่า CEP 50 % นั้นคือค่ามัธยฐาน (Median) ของค่าที่ 50-51 นั้นหมายถึงค่า CEP 50 % เป็นตัวกำหนดรัศมีของวงกลม โดยจะมีค่าที่น้อยกว่า CEP 50 % อยู่ในวงกลม

เนื่องจาก การแสดงผลการทดสอบในรูป Circular Error Probability (CEP) 50 % นั้นมีค่าที่ละเอียดมากต่างกันในจุดทศนิยมตำแหน่งที่ 3 หรือ 4 ดังนั้นการกำหนดจุดให้เป็นไปตาม scale นั้นสามารถทำได้แต่จะเห็นข้อแตกต่างไม่เด่นชัด จึงใช้วิธีการจำลองตำแหน่งหรือ การขยายหรือลดขนาดของ scale ให้ใกล้เคียงความจริงมากที่สุดและสามารถเห็นข้อแตกต่างได้อย่างชัดเจน โดยแสดงในลักษณะการจำลองภาพเพื่อให้เห็นภาพการกระจายของตำแหน่งที่หาได้ ใน 100 Monte Carlo runs โดยทุกๆ case จะมีลักษณะการกระจาย แบบเดียวกัน ซึ่งในวิทยานิพนธ์ฉบับนี้ จะนำเสนอการแสดงผลการทดสอบในรูปของ CEP 50% ใน***กรณีที่ 3 – จำนวน 3 Observers และ Emitter อยู่ใน Observers area*** โดยจะแยกให้เห็นเป็น 4 ขั้นตอนนั่นคือ จะแสดงเริ่มต้นที่ 10 Observations และจะเพิ่มขึ้นทีละ 10 Observations จนถึง 40 Observations ซึ่งสามารถแสดงได้ดังนี้

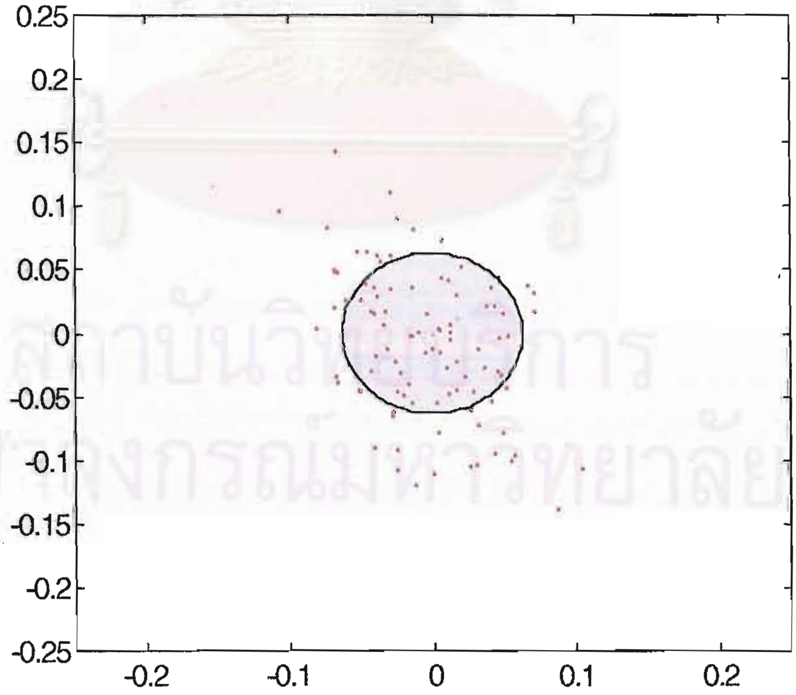
- ***กรณีของ EKF***



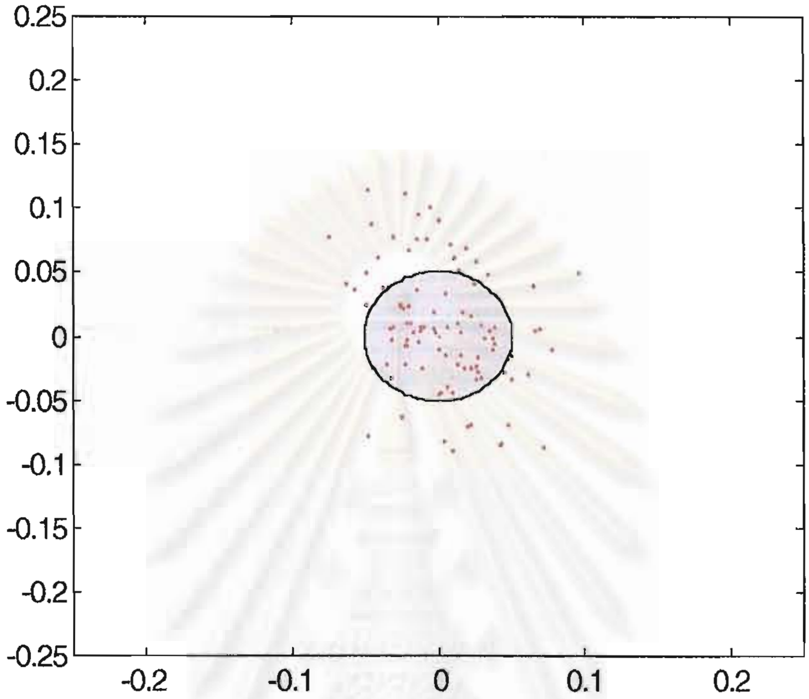
รูปที่ 3.31 วงกลมแสดง Circular Error Probability 50% ของ 100 Monte Carlo runs 10 Observations (EKF)



รูปที่ 3.32 วงกลมแสดง Circular Error Probability 50% ของ 100 Monte Carlo runs 20 Observations (EKF)



รูปที่ 3.33 วงกลมแสดง Circular Error Probability 50% ของ 100 Monte Carlo runs 30 Observations (EKF)



รูปที่ 3.34 วงกลมแสดง Circular Error Probability 50% ของ 100 Monte Carlo runs
40 Observations (EKF)

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

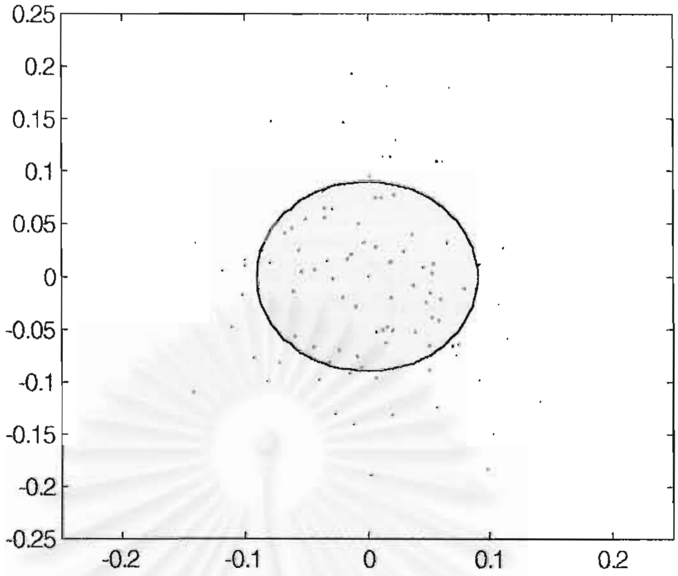
ตารางที่ 3.31 ค่า Circular Error Probability 50%, 40 Observations (EKF) ของรูปที่ 3.34

No.	X	Y	No.	X	Y	No.	X	Y	No.	X	Y
1	15.037	14.997	26	15.026	15.057	51	15.014	14.976	76	14.977	14.991
2	15.000	14.953	27	15.095	15.048	52	15.035	15.003	77	15.029	15.008
3	14.965	14.976	28	15.077	14.989	53	14.983	15.002	78	15.065	15.038
4	15.001	14.955	29	15.048	14.981	54	15.022	15.015	79	15.069	15.005
5	14.955	14.973	30	15.004	14.916	55	15.043	14.915	80	14.978	14.992
6	14.951	15.049	31	15.034	15.047	56	14.989	15.003	81	14.943	15.036
7	15.019	15.068	32	15.017	15.017	57	14.954	15.087	82	14.951	15.114
8	15.005	14.984	33	14.962	15.037	58	14.987	15.094	83	14.985	15.075
9	15.026	14.965	34	14.979	14.996	59	15.015	14.984	84	15.050	14.965
10	15.039	15.006	35	14.985	15.035	60	14.969	15.006	85	15.066	15.003
11	15.005	14.959	36	14.976	15.020	61	14.979	15.009	86	15.027	14.983
12	15.042	14.914	37	15.026	14.975	62	14.925	15.077	87	15.002	15.090
13	14.967	15.005	38	15.071	14.912	63	14.974	15.022	88	15.031	14.998
14	14.980	15.008	39	14.968	14.996	64	14.979	15.067	89	14.986	14.994
15	14.992	15.076	40	14.995	15.099	65	15.026	14.971	90	15.009	14.955
16	15.008	15.070	41	15.010	15.061	66	14.969	15.076	91	15.030	14.967
17	14.973	15.024	42	14.987	15.006	67	14.977	15.111	92	15.015	15.002
18	15.013	15.009	43	15.014	15.050	68	14.966	15.037	93	15.039	14.992
19	15.022	14.930	44	15.020	14.929	69	15.061	14.969	94	14.996	15.017
20	15.009	14.910	45	14.952	14.921	70	15.048	14.930	95	14.989	15.007
21	14.979	15.022	46	14.950	15.023	71	14.985	14.950	96	14.998	14.974
22	14.959	15.060	47	15.000	14.968	72	14.977	14.976	97	14.998	15.002
23	15.005	15.033	48	15.022	14.974	73	14.967	14.966	98	15.017	14.794
24	14.997	15.004	49	15.045	14.971	74	15.049	14.984	99	14.974	14.935
25	14.937	15.040	50	15.006	15.006	75	15.025	15.040	100	15.038	14.989

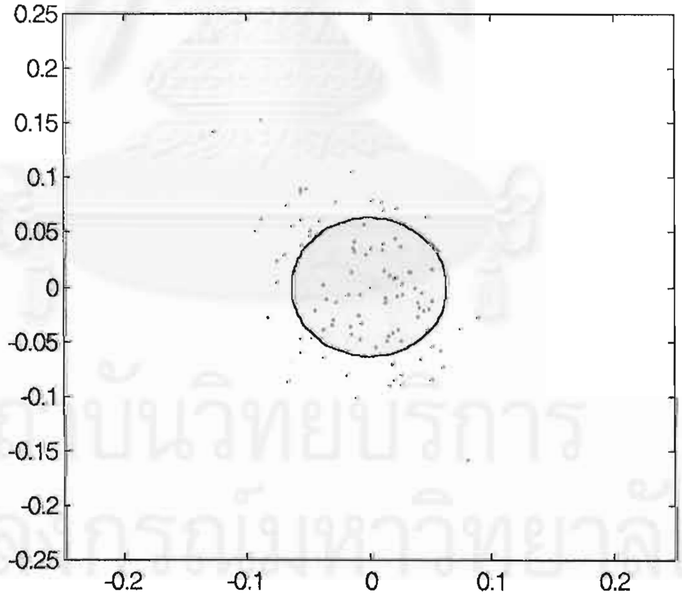
Circular Error Probability (CEP) 50 % ได้ค่า median คือ (15.0037 15.0047)

ได้ระยะ median คือ 0.0060

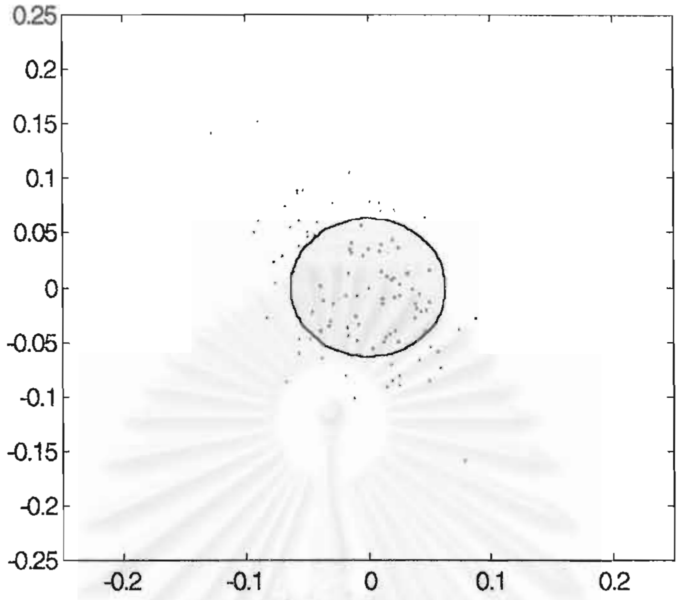
- ***กรณีของ MEKF***



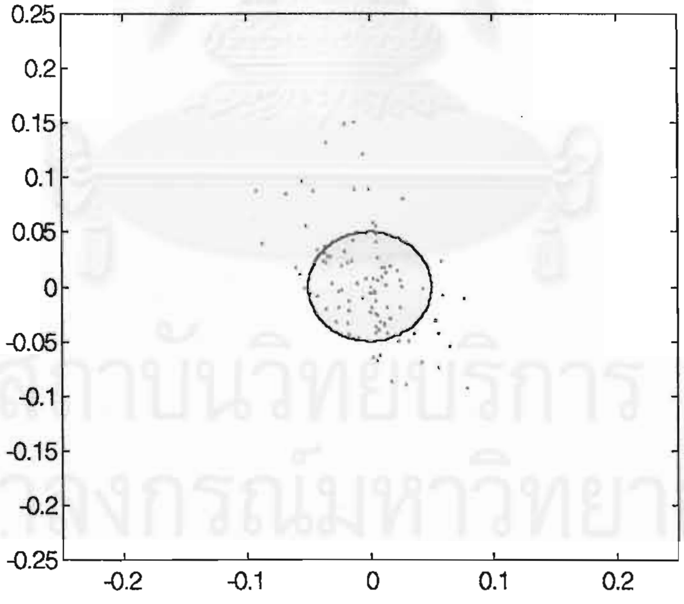
รูปที่ 3.35 วงกลมแสดง Circular Error Probability 50% ของ 100 Monte Carlo runs 10 Observations (MEKF)



รูปที่ 3.36 วงกลมแสดง Circular Error Probability 50% ของ 100 Monte Carlo runs 20 observations (MEKF)



รูปที่ 3.37 วงกลมแสดง Circular Error Probability 50% ของ 100 Monte Carlo runs 30 observations (MEKF)



รูปที่ 3.38 วงกลมแสดง Circular Error Probability 50% ของ 100 Monte Carlo runs 40 observations (MEKF)

ตารางที่ 3.32 ค่า Circular Error Probability, 40 Observations (MEKF) ของรูปที่ 3.38

No.	X	Y	No.	X	Y	No.	X	Y	No.	X	Y
1	15.037	14.970	26	15.024	15.036	51	14.981	14.935	76	14.970	15.001
2	14.998	15.001	27	15.041	15.055	52	14.949	14.990	77	15.025	15.016
3	14.993	14.976	28	15.027	15.027	53	14.961	14.993	78	15.045	14.996
4	15.045	14.976	29	15.031	14.995	54	15.012	15.020	79	15.038	15.024
5	14.965	14.993	30	14.979	14.944	55	15.028	14.970	80	14.971	15.012
6	14.985	15.013	31	15.030	15.062	56	15.007	15.082	81	14.976	15.064
7	15.006	15.050	32	14.978	15.055	57	14.968	15.117	82	14.913	14.993
8	14.969	14.972	33	14.970	15.068	58	14.999	15.106	83	14.995	15.050
9	15.027	14.999	34	14.998	14.972	59	15.044	14.986	84	15.077	14.987
10	15.023	14.972	35	14.965	14.995	60	15.027	15.018	85	15.035	14.965
11	15.009	14.973	36	14.966	14.994	61	14.975	15.053	86	14.976	15.010
12	15.043	14.932	37	15.008	14.977	62	14.920	15.103	87	14.993	15.045
13	14.954	15.017	38	14.971	14.945	63	14.970	15.009	88	15.027	14.931
14	14.995	15.058	39	14.942	14.998	64	14.985	15.015	89	14.976	15.005
15	14.996	15.118	40	14.992	14.987	65	15.019	14.897	90	14.993	14.984
16	15.013	15.077	41	15.012	14.985	66	14.940	15.071	91	14.988	14.968
17	14.944	15.023	42	15.013	14.990	67	14.945	15.053	92	14.969	14.932
18	15.015	14.983	43	14.982	14.979	68	14.970	15.030	93	15.055	15.003
19	15.006	14.915	44	15.024	14.982	69	15.069	14.926	94	14.978	15.043
20	15.023	14.934	45	14.974	14.982	70	14.996	14.894	95	14.957	14.953
21	14.991	15.033	46	14.937	15.049	71	15.010	14.946	96	14.972	14.961
22	14.963	15.059	47	14.992	14.933	72	15.008	14.996	97	15.029	14.943
23	15.009	14.972	48	15.038	14.978	73	14.979	14.993	98	14.992	15.009
24	14.989	14.994	49	15.026	14.979	74	15.013	14.959	99	14.986	15.023
25	14.976	15.045	50	14.986	14.980	75	14.978	15.028	100	15.041	14.997

Circular Error Probability (CEP) 50 % ได้ค่า median คือ (14.9966 15.0014)

ได้ระยะ median คือ 0.0037

จากผลการทดสอบดังที่แสดงให้ทราบแล้วในข้างต้นนั้น ในบทนี้จะนำเสนอข้อพิจารณาสรุป โดยแบ่งข้อพิจารณาสรุปออกเป็น 3 ส่วนใหญ่ ๆ ด้วยกันคือ ส่วนแรก เป็นส่วนของ *Moving observers* ในส่วนที่สอง จะเป็นส่วนของ *Multiple fixed observers (EKF & MEKF)* และในส่วนสุดท้าย จะเป็นส่วนของ *Multiple fixed observers* ซึ่งใช้ *Torrieri's algorithm (KF+TR & MKF+TR)* โดยที่รายละเอียดในแต่ละส่วนของข้อพิจารณาสรุปนั้น สามารถนำมาพิจารณาได้ดังนี้

ในการพิจารณาเพื่อเปรียบเทียบ จัดลำดับเรียงตามความแม่นยำนั้น ในกรณีนี้กำหนดให้เลขที่น้อยกว่าให้ความแม่นยำที่มากกว่าเพื่อให้ง่ายต่อการเปรียบเทียบ โดยแสดงในรูปตารางดังนี้

1. Moving Observers

ตารางที่ 3.33 ลำดับค่าความแม่นยำของการเปรียบเทียบ ระหว่าง EKF & MEKF ทั้ง 5 ประเภท

	EKF		MEKF #1		MEKF #2		MEKF #3		MEKF #4		MEKF #5	
	X	*Y*	*X*	*Y*	*X*	*Y*	*X*	*Y*	*X*	*Y*	*X*	*Y*
2 Obvs	6	6	1	1	4	4	5	5	2	2	3	3
9 Obvs	6	3	5	1	2	4	1	2	4	6	3	5

ตารางที่ 3.34 ลำดับค่าความแม่นยำของการเปรียบเทียบ ระหว่าง EKF & MEKF ทั้ง 5 ประเภท

	EKF	MEKF #1	MEKF #2	MEKF #3	MEKF #4	MEKF #5
	XY	*XY*	*XY*	*XY*	*XY*	*XY*
2 Obvs	6	1	4	5	2	3
9 Obvs	6	4	2	1	5	3

2. Multiple fixed observers (EKF vs MEKFS)

ตารางที่ 3.35 ลำดับค่าความแม่นยำของการเปรียบเทียบ ระหว่าง EKF & MEKF ทั้ง 5 ประเภท

	EKF		MEKF #1		MEKF #2		MEKF #3		MEKF #4		MEKF #5	
	X	Y	X	Y	X	Y	X	Y	X	Y	X	Y
กรณีที่ 1	6	5	4	2	1	3	3	1	5	6	2	4
กรณีที่ 2	6	5	4	2	3	3	5	1	1	6	2	4
กรณีที่ 3	5	3	1	4	4	1	2	5	6	6	3	2
กรณีที่ 4	4	5	6	3	2	2	5	4	1	6	3	1
กรณีที่ 5	5	5	2	1	3	2	1	3	6	6	4	4
กรณีที่ 6	6	5	5	3	3	1	4	4	1	6	2	2

3. Multiple fixed observers (KF+TR vs MKF+TR)

ตารางที่ 3.36 ลำดับค่าความแม่นยำของการเปรียบเทียบ ระหว่าง KF+TR & MKF+TR

	KF + TR		MKF + TR	
	X	Y	X	Y
กรณีที่ 1	1	2	2	1
กรณีที่ 2	2	2	1	1
กรณีที่ 3	1	2	2	1
กรณีที่ 4	1	2	2	1
กรณีที่ 5	2	1	1	2
กรณีที่ 6	2	2	1	1

จากผลการทดลองในส่วนของ Circular Error Probability (CEP) 50% สามารถนำมาสรุปเปรียบเทียบในรูปของตารางได้ดังนี้

ตารางที่ 3.37 ลำดับค่าความแม่นยำในกรณีของ Moving observers ในรูปของ CEP 50% กรณี [illegible] Obsvs

	CEP 50% (km)	ลำดับที่
EKF	1.2090	6
MEKF#1	0.5108	1
MEKF#2	1.0720	4
MEKF#3	1.1215	5
MEKF#4	0.6386	2
MEKF#5	0.8637	3

ตารางที่ 3.38 ลำดับค่าความแม่นยำในกรณีของ Moving observers ในรูปของ CEP 50% กรณี 9 Obsvs

	CEP 50% (km)	ลำดับที่
EKF	0.7663	6
MEKF#1	0.4842	5
MEKF#2	0.3293	2
MEKF#3	0.2230	1
MEKF#4	0.4760	4
MEKF#5	0.3882	3

ตารางที่ 3.39 ลำดับค่าความแม่นยำในกรณีของ Multiple fixed observations ในรูปของ CEP 50% ในกรณีที่ 1

	CEP 50% (km)	ลำดับที่
EKF	0.0135	6
MEKF#1	0.0108	5
MEKF#2	0.0080	3
MEKF#3	0.0103	4
MEKF#4	0.0039	1
MEKF#5	0.0074	2

ตารางที่ 3.40 ลำดับค่าความแม่นยำในกรณีของ Multiple fixed observations ในรูปของ CEP 50% ในกรณีที่ 2

	CEP 50% (km)	ลำดับที่
EKF	0.0204	6
MEKF#1	0.0180	5
MEKF#2	0.0116	3
MEKF#3	0.0094	1
MEKF#4	0.0162	4
MEKF#5	0.0107	2

ตารางที่ 3.41 ลำดับค่าความแม่นยำในกรณีของ Multiple fixed observations ในรูปของ CEP 50% ในกรณีที่ 3

	CEP 50% (km)	ลำดับที่
EKF	0.0062	5
MEKF#1	0.0037	4
MEKF#2	0.0016	1
MEKF#3	0.0023	2
MEKF#4	0.0070	6
MEKF#5	0.0025	3

ตารางที่ 3.42 ลำดับค่าความแม่นยำในกรณีของ Multiple fixed observations ในรูปของ CEP 50% ในกรณีที่ 4

	CEP 50% (km)	ลำดับที่
EKF	0.0548	6
MEKF#1	0.0222	4
MEKF#2	0.0156	2
MEKF#3	0.0100	1
MEKF#4	0.0365	5
MEKF#5	0.0159	3

ตารางที่ 3.43 ลำดับค่าความแม่นยำในกรณีของ Multiple fixed observations ในรูปของ CEP 50% ในกรณีที่ 5

	CEP 50% (km)	ลำดับที่
EKF	0.0087	5
MEKF#1	0.0042	1
MEKF#2	0.0058	4
MEKF#3	0.0057	3
MEKF#4	0.0115	6
MEKF#5	0.0044	2

ตารางที่ 3.44 ลำดับค่าความแม่นยำในกรณีของ Multiple fixed observations ในรูปของ CEP 50% ในกรณีที่ 6

	CEP 50% (km)	ลำดับที่
EKF	0.0652	6
MEKF#1	0.0409	5
MEKF#2	0.0100	1
MEKF#3	0.0291	4
MEKF#4	0.0132	2
MEKF#5	0.0171	3

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

วิเคราะห์ผลการทดสอบ

กรณีเครื่องดักรับหาทิศสัญญาณแบบเคลื่อนที่

การเปรียบเทียบเชิงปริมาณใช้อัลกอริทึมแบบ EKF เป็นเกณฑ์ที่ใช้ในการเปรียบเทียบ เพื่อให้ได้ทราบว่า อัลกอริทึมแบบ MEKF ทั้ง 5 ประเภท มีผลการทดสอบที่แม่นยำกว่ามาก น้อยเพียงใด โดยใช้ข้อมูลจากตาราง CEP 50%

กรณีที่ 1 จำนวน 2 Observations

จากตารางที่ 3.37 สามารถสรุปความแม่นยำเป็นเปอร์เซ็นต์ได้ดังนี้

อัลกอริทึมของ MEKF # 1 ให้ความแม่นยำกว่าอัลกอริทึมของ	EKF	57.75 %
อัลกอริทึมของ MEKF # 2 ให้ความแม่นยำกว่าอัลกอริทึมของ	EKF	11.33 %
อัลกอริทึมของ MEKF # 3 ให้ความแม่นยำกว่าอัลกอริทึมของ	EKF	7.24 %
อัลกอริทึมของ MEKF # 4 ให้ความแม่นยำกว่าอัลกอริทึมของ	EKF	47.18 %
อัลกอริทึมของ MEKF # 5 ให้ความแม่นยำกว่าอัลกอริทึมของ	EKF	28.56 %

จากค่าที่ได้สามารถสรุปได้คือ MEKF # 1 ให้ความแม่นยำได้มากที่สุด และ EKF ให้ความแม่นยำน้อยที่สุด ดังนั้น อัลกอริทึมที่เหมาะสมที่จะใช้ในกรณีนี้ ซึ่งทางปฏิบัติจริงใช้ในทางการบินทางยุทธวิธีนั้นได้แก่ อัลกอริทึมของ MEKF # 1

กรณีที่ 2 จำนวน 9 Observations

จากตารางที่ 3.38 สามารถสรุปความแม่นยำเป็นเปอร์เซ็นต์ได้ดังนี้

อัลกอริทึมของ MEKF # 1 ให้ความแม่นยำกว่าอัลกอริทึมของ	EKF	36.81 %
อัลกอริทึมของ MEKF # 2 ให้ความแม่นยำกว่าอัลกอริทึมของ	EKF	57.03 %
อัลกอริทึมของ MEKF # 3 ให้ความแม่นยำกว่าอัลกอริทึมของ	EKF	70.89 %
อัลกอริทึมของ MEKF # 4 ให้ความแม่นยำกว่าอัลกอริทึมของ	EKF	37.88 %
อัลกอริทึมของ MEKF # 5 ให้ความแม่นยำกว่าอัลกอริทึมของ	EKF	49.34 %

จากค่าที่ได้สามารถสรุปได้คือ MEKF # 3 ให้ความแม่นยำได้มากที่สุด และ EKF ให้ความแม่นยำน้อยที่สุด ดังนั้น อัลกอริทึมที่เหมาะสมที่จะใช้ในกรณีนี้ ซึ่งทางปฏิบัติจริงใช้ในทางการบินหาทิศทางยุทธศาสตร์ นั้นได้แก่ อัลกอริทึมของ MEKF # 3

กรณีเครื่องดักรับหาทิศสัญญาณแบบอยู่กับที่

ลักษณะการเปรียบเทียบเชิงปริมาณใช้ ลักษณะเดียวกับกรณี เครื่องดักรับหาทิศสัญญาณแบบเคลื่อนที่ กล่าวคือใช้อัลกอริทึมแบบ EKF เป็นเกณฑ์ที่ใช้ในการเปรียบเทียบ โดยใช้ข้อมูลจากตาราง CEP 50%

กรณีที่ 1 จำนวน 2 DF, Source in

จากตารางที่ 3.39 สามารถสรุปความแม่นยำเป็นเปอร์เซ็นต์ได้ดังนี้

อัลกอริทึมของ MEKF # 1 ให้ความแม่นยำกว่าอัลกอริทึมของ	EKF	20.00 %
อัลกอริทึมของ MEKF # 2 ให้ความแม่นยำกว่าอัลกอริทึมของ	EKF	40.74 %
อัลกอริทึมของ MEKF # 3 ให้ความแม่นยำกว่าอัลกอริทึมของ	EKF	23.70 %
อัลกอริทึมของ MEKF # 4 ให้ความแม่นยำกว่าอัลกอริทึมของ	EKF	71.11 %
อัลกอริทึมของ MEKF # 5 ให้ความแม่นยำกว่าอัลกอริทึมของ	EKF	45.19 %

จากค่าที่ได้สามารถสรุปได้คือ MEKF # 4 ให้ความแม่นยำได้มากที่สุด และ EKF ให้ความแม่นยำน้อยที่สุด ดังนั้น อัลกอริทึมที่เหมาะสมที่จะใช้ในกรณีนี้ ซึ่งทางปฏิบัติจริงใช้ในทางการดักรับหาทิศสัญญาณแบบเร่งด่วนได้แก่ อัลกอริทึมของ MEKF # 4

กรณีที่ 2 จำนวน 2 DF, Source out

จากตารางที่ 3.40 สามารถสรุปความแม่นยำเป็นเปอร์เซ็นต์ได้ดังนี้

อัลกอริทึมของ MEKF # 1 ให้ความแม่นยำกว่าอัลกอริทึมของ	EKF	11.76 %
อัลกอริทึมของ MEKF # 2 ให้ความแม่นยำกว่าอัลกอริทึมของ	EKF	43.14 %
อัลกอริทึมของ MEKF # 3 ให้ความแม่นยำกว่าอัลกอริทึมของ	EKF	53.92 %
อัลกอริทึมของ MEKF # 4 ให้ความแม่นยำกว่าอัลกอริทึมของ	EKF	20.59 %
อัลกอริทึมของ MEKF # 5 ให้ความแม่นยำกว่าอัลกอริทึมของ	EKF	47.55 %

จากค่าที่ได้สามารถสรุปได้คือ MEKF # 3 ให้ความแม่นยำได้มากที่สุด และ EKF ให้ความแม่นยำน้อยที่สุด ดังนั้น อัลกอริทึมที่เหมาะสมที่จะใช้ในกรณีนี้ ซึ่งทางปฏิบัติจริงใช้ในทางการดักรับหาทิศสัญญาณแบบเร่งด่วน โดยไม่ทราบว่าสัญญาณที่เข้ามาอยู่ภายในพื้นที่ดักรับหรือไม่ ได้แก่ อัลกอริทึมของ MEKF # 4

กรณีที่ 3 จำนวน 3 DF, Source in

จากตารางที่ 3.41 สามารถสรุปความแม่นยำเป็นเปอร์เซ็นต์ได้ดังนี้

อัลกอริทึมของ MEKF # 1 ให้ความแม่นยำกว่าอัลกอริทึมของ EKF 40.32 %

อัลกอริทึมของ MEKF # 2 ให้ความแม่นยำกว่าอัลกอริทึมของ EKF 74.19 %

อัลกอริทึมของ MEKF # 3 ให้ความแม่นยำกว่าอัลกอริทึมของ EKF 62.90 %

อัลกอริทึมของ MEKF # 4 ให้ความแม่นยำน้อยกว่าอัลกอริทึมของ EKF 12.90 %

อัลกอริทึมของ MEKF # 5 ให้ความแม่นยำกว่าอัลกอริทึมของ EKF 59.68 %

จากค่าที่ได้สามารถสรุปได้คือ MEKF # 2 ให้ความแม่นยำได้มากที่สุด และ MEKF # 4 ให้ความแม่นยำน้อยที่สุด ดังนั้น อัลกอริทึมที่เหมาะสมที่จะใช้ในกรณีนี้ ซึ่งทางปฏิบัติจริงใช้ในทางการดักรับหาทิศสัญญาณแบบปกติหรือ ทางยุทธวิธีขั้นเตรียมพร้อม ได้แก่ อัลกอริทึมของ MEKF # 2

กรณีที่ 4 จำนวน 3 DF, Source out

จากตารางที่ 3.42 สามารถสรุปความแม่นยำเป็นเปอร์เซ็นต์ได้ดังนี้

อัลกอริทึมของ MEKF # 1 ให้ความแม่นยำกว่าอัลกอริทึมของ EKF 59.49 %

อัลกอริทึมของ MEKF # 2 ให้ความแม่นยำกว่าอัลกอริทึมของ EKF 71.53 %

อัลกอริทึมของ MEKF # 3 ให้ความแม่นยำกว่าอัลกอริทึมของ EKF 81.75 %

อัลกอริทึมของ MEKF # 4 ให้ความแม่นยำกว่าอัลกอริทึมของ EKF 33.39 %

อัลกอริทึมของ MEKF # 5 ให้ความแม่นยำกว่าอัลกอริทึมของ EKF 70.98 %

จากค่าที่ได้สามารถสรุปได้คือ MEKF # 3 ให้ความแม่นยำได้มากที่สุด และ EKF ให้ความแม่นยำน้อยที่สุด ดังนั้น อัลกอริทึมที่เหมาะสมที่จะใช้ในกรณีนี้ ซึ่งทางปฏิบัติจริงใช้ในทางการดักรับหาทิศสัญญาณแบบปกติหรือ ทางยุทธวิธีขั้นเตรียมพร้อม โดยไม่ทราบว่าสัญญาณที่เข้ามาอยู่ภายในพื้นที่ดักรับหรือไม่ ได้แก่ อัลกอริทึมของ MEKF # 3

กรณีที่ 5 จำนวน 4 DF, Source in

จากตารางที่ 3.43 สามารถสรุปความแม่นยำเป็นเปอร์เซ็นต์ได้ดังนี้

อัลกอริทึมของ MEKF # 1 ให้ความแม่นยำกว่าอัลกอริทึมของ EKF 51.72 %

อัลกอริทึมของ MEKF # 2 ให้ความแม่นยำกว่าอัลกอริทึมของ EKF 33.33 %

อัลกอริทึมของ MEKF # 3 ให้ความแม่นยำกว่าอัลกอริทึมของ EKF 34.48 %

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

อัลกอริทึมของ MEKF # 4 ให้ความแม่นยำน้อยกว่าอัลกอริทึมของ EKF 32.18 %

อัลกอริทึมของ MEKF # 5 ให้ความแม่นยำกว่าอัลกอริทึมของ EKF 49.43 %

จากค่าที่ได้สามารถสรุปได้คือ MEKF # 1 ให้ความแม่นยำได้มากที่สุด และ MEKF # 4 ให้ความแม่นยำน้อยที่สุด ดังนั้น อัลกอริทึมที่เหมาะสมที่จะใช้ในกรณีนี้ ซึ่งทางปฏิบัติจริงใช้ในทางการดักรับหาทิศสัญญาณทางยุทธศาสตร์ได้แก่ อัลกอริทึมของ MEKF # 1

กรณีที่ 6 จำนวน 4 DF, Source out

จากตารางที่ 3.44 สามารถสรุปความแม่นยำเป็นเปอร์เซ็นต์ได้ดังนี้

อัลกอริทึมของ MEKF # 1 ให้ความแม่นยำกว่าอัลกอริทึมของ EKF 37.27 %

อัลกอริทึมของ MEKF # 2 ให้ความแม่นยำกว่าอัลกอริทึมของ EKF 84.66 %

อัลกอริทึมของ MEKF # 3 ให้ความแม่นยำกว่าอัลกอริทึมของ EKF 55.36 %

อัลกอริทึมของ MEKF # 4 ให้ความแม่นยำกว่าอัลกอริทึมของ EKF 79.75 %

อัลกอริทึมของ MEKF # 5 ให้ความแม่นยำกว่าอัลกอริทึมของ EKF 73.77 %

จากค่าที่ได้สามารถสรุปได้คือ MEKF # 2 ให้ความแม่นยำได้มากที่สุด และ EKF ให้ความแม่นยำน้อยที่สุด ดังนั้น อัลกอริทึมที่เหมาะสมที่จะใช้ในกรณีนี้ ซึ่งทางปฏิบัติจริงใช้ในทางการดักรับหาทิศสัญญาณทางยุทธศาสตร์โดยไม่ทราบว่าสัญญาณที่เข้ามาอยู่ภายในพื้นที่ดักรับหรือไม่ ได้แก่ อัลกอริทึมของ MEKF # 2

วิจารณ์ผลการทดสอบ

กรณีเครื่องดักรับหาทิศสัญญาณเคลื่อนที่

กรณีที่ 1 : เครื่องดักรับหาทิศสัญญาณจำนวน 2 เครื่อง

จากผลการทดสอบที่ได้ และจากการแสดงผลใน รูปที่ 3.3 ตารางที่ 3.3 และตารางที่ 3.34 จะเห็นได้อย่างชัดเจนว่า อัลกอริทึมของ EKF นั้น ให้ความแม่นยำน้อยกว่า อัลกอริทึมของ MEKF ในทุกๆ ประเภท และในแต่ละประเภทของ อัลกอริทึมแบบ MEKF นั้น MEKF ประเภทที่ 1 ให้ความแม่นยำมากที่สุด ดังนั้นในการประยุกต์ใช้กับ การบินหาทิศทางสัญญาณแบบจำกัดเวลาหรือ การบินทางยุทธวิธีนั้น อัลกอริทึมของ MEKF ประเภทที่ 1 จะสามารถให้ความแม่นยำได้มากที่สุด

กรณีที่ 2 : เครื่องดักรับหาทิศสัญญาณจำนวน 9 เครื่อง

จากผลการทดสอบที่ได้ และจากการแสดงผลใน รูปที่ 3.6 ตารางที่ 3.6 และตารางที่ 3.34 จะเห็นได้อย่างชัดเจนว่า อัลกอริทึมของ EKF นั้น ให้ความแม่นยำน้อยกว่า อัลกอริทึมของ MEKF ในทุกๆ ประเภท และในแต่ละประเภทของ อัลกอริทึมแบบ MEKF นั้น MEKF ประเภทที่ 3 ให้ความแม่นยำมากที่สุด ดังนั้นในการประยุกต์ใช้กับ การบินหาทิศทางยุทธศาสตร์นั้น อัลกอริทึมของ MEKF ประเภทที่ 3 จะสามารถให้ความแม่นยำได้มากที่สุด

กรณีเครื่องดักรับหาทิศสัญญาณอยู่กับที่ (MEKF)

กรณีที่ 1 : เครื่องดักรับหาทิศสัญญาณจำนวน 2 เครื่อง อยู่ภายในพื้นที่ดักรับ

จากผลการทดสอบที่ได้ และจากการแสดงผลใน ตารางที่ 3.39 จะเห็นได้อย่างชัดเจนว่า อัลกอริทึมของ EKF นั้น ให้ความแม่นยำน้อยกว่า อัลกอริทึมของ MEKF ในทุกๆ ประเภท และในแต่ละประเภทของ อัลกอริทึมแบบ MEKF นั้น MEKF ประเภทที่ 4 ให้ความแม่นยำมากที่สุด ดังนั้นในการประยุกต์ใช้กับ การดักรับหาทิศสัญญาณแบบเร่งด่วนหรือ ทางยุทธวิธีนั้น อัลกอริทึมของ MEKF ประเภทที่ 4 จะสามารถให้ความแม่นยำได้มากที่สุด ทั้งนี้ผู้ปฏิบัติต้องสามารถระบุทิศทางของสัญญาณโดยประมาณได้ว่าสัญญาณที่ต้องการดักรับนั้นอยู่ภายในพื้นที่ดักรับ

ในกรณีเครื่องดักรับหาทิศสัญญาณที่อยู่ภายในพื้นที่ดักรับนี้ มีข้อดีคือ สามารถให้ความแม่นยำ และความรวดเร็วได้มากกว่า กรณีเครื่องดักรับหาทิศสัญญาณที่อยู่ภายนอกพื้นที่ดักรับ ดังในรูปที่ 3.7 และรูปที่ 3.8 ซึ่งแสดงผลการทดสอบ ในกรณีของเครื่องดักรับหาทิศสัญญาณจำนวน 2 เครื่อง อยู่ภายในพื้นที่ดักรับ ในแกน X และ Y ตามลำดับนั้น ใช้เวลาในการประมวลผลเฉลี่ย 20.58 วินาที ส่วน รูปที่ 3.9 และ รูปที่ 3.10 ซึ่งแสดงผลการทดสอบของ เครื่องดักรับหาทิศสัญญาณจำนวน 2 เครื่องที่อยู่ภายนอกพื้นที่ดักรับนั้น ใช้เวลาในการประมวลผลเฉลี่ย 24.69 วินาที ซึ่งจากผลของการประมวลผลเวลาโดยเฉลี่ยในทั้ง 2 กรณีนั้น จะเห็นได้ว่า กรณีเครื่องดักรับหาทิศสัญญาณที่อยู่ภายในพื้นที่ดักรับนั้น ใช้เวลาประมวลผลน้อยกว่า กรณีเครื่องดักรับหาทิศสัญญาณที่อยู่ภายนอกพื้นที่ดักรับ 4.11 วินาที ซึ่งจากการที่ใช้เวลาน้อยกว่าในการประมวลผลนี้ จะเป็นประโยชน์ในทางปฏิบัติ ในเรื่องของความได้เปรียบในการเริ่มดำเนินกลยุทธ

กรณีที่ 2 : เครื่องดักรับหาทิศสัญญาณจำนวน 2 เครื่อง อยู่ภายนอกพื้นที่ดักรับ

จากผลการทดสอบที่ได้ และจากการแสดงผลใน ตารางที่ 3.40 จะเห็นได้อย่างชัดเจนว่า อัลกอริทึมของ EKF นั้น ให้ความแม่นยำน้อยกว่า อัลกอริทึมของ MEKF ในทุกๆ ประเภท และในแต่ละประเภทของ อัลกอริทึมแบบ MEKF นั้น MEKF ประเภทที่ 3 ให้ความแม่นยำมากที่สุด ดังนั้นในการประยุกต์ใช้กับ การดักรับหาทิศสัญญาณแบบเร่งด่วนหรือ ทางยุทธวิธีนั้น อัลกอริทึมของ MEKF ประเภทที่ 3 จะสามารถให้ความแม่นยำได้มากที่สุดซึ่งในกรณีนี้ผู้ปฏิบัติไม่จำเป็นต้องระบุทิศทางของสัญญาณว่า สัญญาณที่ต้องการดักรับนั้นอยู่ภายในพื้นที่ดักรับหรือไม่

ในกรณีเครื่องดักรับหาทิศสัญญาณที่อยู่ภายนอกพื้นที่ดักรับนี้ มีข้อดีคือ ในทางการปฏิบัติจริงนั้นส่วนใหญ่เราไม่สามารถระบุได้ว่า สัญญาณที่ต้องการหานั้นอยู่ภายในพื้นที่ดักรับหรือไม่ ส่วนข้อเสียคือใช้เวลาในการประมวลผลเฉลี่ยมากกว่า กรณีที่ 1 (มากกว่า 4.11 วินาที) ซึ่งการเริ่มดำเนินกลยุทธที่ช้ากว่า จะเป็นข้อเสียเปรียบของฝ่ายเรา

กรณีที่ 3 : เครื่องดักรับหาทิศสัญญาณจำนวน 3 เครื่อง อยู่ภายในพื้นที่ดักรับ

จากผลการทดลองที่ได้ และจากการแสดงผลใน ตารางที่ 3.41 จะเห็นได้ว่า อัลกอริทึมของ EKF นั้น ให้ความแม่นยำได้น้อยกว่า อัลกอริทึมของ MEKF ทั้ง 4 ประเภท และในแต่ละประเภทของอัลกอริทึมแบบ MEKF นั้น MEKF ประเภทที่ 2 ให้ความแม่นยำมากที่สุด ดังนั้นในการประยุกต์ใช้กับ การดักรับหาทิศสัญญาณแบบปกติหรือ ทางยุทธวิธีขั้นเตรียมพร้อมนั้น อัลกอริทึมของ MEKF ประเภทที่ 2 สามารถให้ความแม่นยำได้มากที่สุด ทั้งนี้ผู้ปฏิบัติต้องสามารถระบุทิศทางของสัญญาณโดยประมาณได้ว่า สัญญาณที่ต้องการดักรับนั้นอยู่ภายในพื้นที่ดักรับ

ในกรณีนี้ ข้อดีก็คือ สามารถให้ความแม่นยำได้มากกว่าในกรณีที่ 1 และ กรณีที่ 2 แต่ข้อเสียก็คือ ใช้เวลาในการประมวลผลนานกว่า ทั้ง 2 กรณี (กรณีที่ 1 ใช้เวลาประมวลผลเฉลี่ย 20.58 วินาที และ กรณีที่ 2 ใช้เวลาประมวลผล 24.69 วินาที) ซึ่งจากรูปที่ 3.11และ รูปที่ 3.12 ที่แสดงผลการทดสอบในกรณีที่ 3 นี้ ใช้เวลาประมวลผลเฉลี่ย 36.62 วินาที ซึ่งจะเห็นได้ว่า ใช้เวลาในการประมวลผลเฉลี่ยมากกว่า ในกรณีที่ 1 และกรณีที่ 2 อยู่ 16.04 วินาที และ 11.93 วินาทีตามลำดับ ซึ่งในทางยุทธวิธีนั้นถือว่าเป็นเวลาที่ค่อนข้างมาก ดังนั้นในกรณีที่ 3 นี้ จึงเหมาะแก่การใช้ในขั้นปกติหรือ ขั้นเตรียมพร้อมมากกว่าใช้เชิงยุทธวิธี

กรณีที่ 4 : เครื่องดักรับหาทิศสัญญาณจำนวน 3 เครื่อง อยู่ภายนอกพื้นที่ดักรับ

จากผลการทดสอบที่ได้ และจากการแสดงผลใน ตารางที่ 3.42 จะเห็นได้อย่างชัดเจนว่า อัลกอริทึมของ EKF นั้น ให้ความแม่นยำน้อยกว่า อัลกอริทึมของ MEKF ในทุกๆ กรณี และในแต่ละประเภทของ อัลกอริทึมแบบ MEKF นั้น MEKF ประเภทที่ 3 ให้ความแม่นยำมากที่สุด ดังนั้นในการประยุกต์ใช้กับ การดักรับหาทิศสัญญาณแบบปกติหรือ ทางยุทธวิธีขั้นเตรียมพร้อมนั้น อัลกอริทึมของ MEKF ประเภทที่ 3 สามารถให้ความแม่นยำได้มากที่สุด ซึ่งในกรณีนี้ผู้ปฏิบัติไม่จำเป็นต้องระบุทิศทางของสัญญาณว่า สัญญาณที่ต้องการดักรับนั้นอยู่ภายในพื้นที่ดักรับหรือไม่

ในกรณีนี้ ข้อดีก็คือ สามารถให้ความแม่นยำได้มากกว่าในกรณีที่ 1 (รูปที่3.7และ รูปที่3.8) และ กรณีที่ 2 (รูปที่ 3.9และ รูปที่ 3.10) โดยที่ไม่จำเป็นต้องระบุว่าทิศทางของสัญญาณจะต้องอยู่ภายในพื้นที่ดักรับ แต่ข้อเสียก็คือใช้เวลาในการประมวลผลนานกว่า ทั้ง 3 กรณี ซึ่งจากรูปที่ 3.13และ รูปที่ 3.14 ที่แสดงผลการทดสอบในกรณีนี้นั้น ใช้เวลาประมวลผลเฉลี่ย 37.84 วินาที ซึ่งจะเห็นได้ว่า ในกรณีที่ 4 นี้ ใช้เวลาในการประมวลผลเฉลี่ยมากกว่า ในกรณีที่ 1 อยู่ 17.06 วินาที มากกว่ากรณีที่ 2 อยู่ 12.95 วินาที และ มากกว่ากรณีที่ 3 อยู่ 1.02 วินาที ซึ่งในทางยุทธวิธีนั้นถือว่าเป็นเวลาที่ค่อนข้างมาก ถ้านำมาใช้อาจเป็นเหตุให้เกิดความสูญเสียต่อฝ่ายเราได้ ดังนั้นในกรณีที่ 4 นี้ จึงเหมาะแก่การใช้ในขั้นปกติหรือ ขั้นเตรียมพร้อมมากกว่าใช้เชิงยุทธวิธี

กรณีที่ 5 : เครื่องดักรับหาทิศสัญญาณจำนวน 4 เครื่อง อยู่ภายในพื้นที่ดักรับ

จากผลการทดสอบที่ได้ และจากการแสดงผลใน ตารางที่ 3.43 จะเห็นได้ว่า อัลกอริทึมของ EKF นั้น ให้ความแม่นยำน้อยกว่า อัลกอริทึมของ MEKF ทั้ง 4 ประเภท และในแต่ละประเภทของ อัลกอริทึมแบบ MEKF นั้น MEKF ประเภทที่ 1 ให้ความแม่นยำมากที่สุด ดังนั้นในการประยุกต์ใช้กับ การดักรับหาทิศสัญญาณทางยุทธศาสตร์นั้น อัลกอริทึมของ MEKF ประเภทที่ 1 สามารถให้ความแม่นยำได้มากที่สุด ทั้งนี้ผู้ปฏิบัติต้องสามารถระบุทิศทางของสัญญาณโดยประมาณได้ว่า สัญญาณที่ต้องการดักรับนั้นอยู่ภายในพื้นที่ดักรับ

ในกรณีนี้ ข้อดีก็คือ สามารถให้ความแม่นยำได้มากกว่าในกรณีที่ 1 (รูปที่3.7และ รูปที่3.8) กรณีที่ 2 (รูปที่ 3.9และ รูปที่ 3.10), กรณีที่ 3 (รูปที่3.11และ รูปที่ 3.12) และกรณีที่ 4 (รูปที่ 3.13และ รูปที่ 3.14) แต่ข้อเสียก็คือใช้เวลาในการประมวลผลนานกว่า ทั้ง 4 กรณี ซึ่งจากรูปที่ 3.15และ รูปที่ 3.16 ที่แสดงผลการทดสอบในกรณีนี้ ใช้เวลาประมวลผลเฉลี่ย 47.08 วินาที ซึ่งจะเห็นได้ว่า ในกรณีที่ 5 นี้ ใช้เวลาในการประมวลผลเฉลี่ยมากกว่า ในกรณีที่ 1 อยู่ 26.5 วินาที, มากกว่ากรณีที่ 2 อยู่ 22.39, มากกว่ากรณีที่ 3 อยู่

10.46 วินาทีและมากกว่ากรณีที่ 4 อยู่ 9.44 วินาที ซึ่งเป็นเวลาที่ค่อนข้างมาก ดังนั้นในกรณีที่ 5 นี้ จึงเหมาะแก่การใช้ในทางยุทธศาสตร์หรือในระดับประเทศเพื่อเสริมสร้างความมั่นคง มากกว่าการใช้ทางยุทธวิธีหรือการใช้ในขั้นปกติ

กรณีที่ 6 : เครื่องดักรับหาทิศสัญญาณจำนวน 4 เครื่อง อยู่ภายนอกพื้นที่ดักรับ

จากผลการทดสอบที่ได้ และจากการแสดงผลใน ตารางที่ 3.44 จะเห็นได้อย่างชัดเจนว่า อัลกอริทึมของ EKF นั้น ให้ความแม่นยำน้อยกว่า อัลกอริทึมของ MEKF ในทุกๆ กรณี และในแต่ละประเภทของ อัลกอริทึมแบบ MEKF นั้น MEKF ประเภทที่ 2 ให้ความแม่นยำมากที่สุด ดังนั้นในการประยุกต์ใช้กับ การดักรับหาทิศสัญญาณทางยุทธศาสตร์นั้น อัลกอริทึมของ MEKF ประเภทที่ 2 สามารถให้ความแม่นยำได้มากที่สุด ซึ่งในกรณีนี้ผู้ปฏิบัติไม่จำเป็นต้องระบุทิศทางของสัญญาณว่า สัญญาณที่ต้องการดักรับนั้นอยู่ภายในพื้นที่ดักรับหรือไม่

ในกรณีนี้ ข้อดีก็คือ สามารถให้ความแม่นยำได้มากกว่าในกรณีที่ 1 (รูปที่3.7และ รูปที่3.8 กรณีที่ 2 (รูปที่ 3.9และ รูปที่ 3.10), กรณีที่ 3 (รูปที่3.11และ รูปที่ 3.12) และกรณีที่ 4 (รูปที่ 3.13และ รูปที่ 3.14) แต่ข้อเสียก็คือใช้เวลาในการประมวลผลนานกว่า ทั้ง 5 กรณี ซึ่งจากรูปที่ 3.17และ รูปที่ 3.18 ที่แสดงผลการทดสอบในกรณีนี้ ใช้เวลาประมวลผลเฉลี่ย 49.85 วินาที ซึ่งจะเห็นได้ว่า ในกรณีที่ 6 นี้ ใช้เวลาในการประมวลผลเฉลี่ยมากกว่า ในกรณีที่ 1 อยู่ 29.27 วินาที, มากกว่ากรณีที่ 2 อยู่ 25.16, มากกว่ากรณีที่ 3 อยู่ 12.21 วินาที, มากกว่ากรณีที่ 4 อยู่ 13.23 วินาที และมากกว่ากรณีที่ 5 อยู่ 2.77 วินาที ซึ่งเป็นเวลาที่ค่อนข้างมาก ดังนั้นในกรณีที่ 6 นี้ จึงเหมาะแก่การใช้ในทางยุทธศาสตร์หรือในระดับประเทศเพื่อเสริมสร้างความมั่นคง มากกว่าการใช้ทางยุทธวิธีหรือ การใช้ในขั้นปกติ

กรณีเครื่องดักรับหาทิศสัญญาณอยู่กับที่ (MKF+TR)

จากผลการทดลองที่ได้และ จากตารางที่ 3.36 นั้น สังเกตได้ว่า ในทั้ง 6 กรณีนั้น อัลกอริทึมของ KF+TR นั้น ให้ความแม่นยำได้ใกล้เคียงกับอัลกอริทึมของ MKF+TR มากยกเว้นในกรณีที่ 2 และ กรณีที่ 6 ที่อัลกอริทึมของ MKF+TR ให้ความแม่นยำมากกว่าอย่างชัดเจน ทั้งนี้เนื่องจาก อัลกอริทึมของ MKF+TR นั้น ได้นำการประมวลผลแบบขนานมาประยุกต์ใช้ กล่าวคือได้นำ KF 2 ตัว มาประมวลผลแบบขนานและทำการแลกข้อมูลป้อนเข้าซึ่งกันและกัน ในลักษณะเดียวกันกับ MEKF ส่วนในกรณี KF+TR นั้น ได้ใช้ KF เพียง 1 ตัวเท่านั้นในการประมวลผล ข้อดีของอัลกอริทึมประเภทนี้ก็คือสามารถให้ค่าความแม่นยำได้ในระดับหนึ่ง แต่ข้อเสียก็คือ ใช้เวลาในการประมวลผลนานมาก

จากรูปที่ 3.27 จะเห็นได้ว่าใช้เวลาในการประมวลผลประมาณ 298 วินาที หรือประมาณ 4 นาที 58 วินาที ซึ่งในทางปฏิบัตินั้น ผู้ปฏิบัติจำเป็นต้องใช้ความรวดเร็วและความแม่นยำอย่างสูงในการปฏิบัติเพื่อหลีกเลี่ยงการถูกกระทำในลักษณะเดียวกันจากฝ่ายตรงข้าม ดังนั้นในอัลกอริทึมประเภทนี้จึงเหมาะกับการใช้ในที่ตั้งปกติ หรือใช้สำหรับการทดลองเพื่อเปรียบเทียบ หากจะใช้ทางยุทธวิธี ควรใช้เครื่องมือที่มีขีดความสามารถในการประมวลผลเร็วมากๆ จึงจะสามารถใช้ได้ทันเวลาที่ต้องการ

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

หอสมุดกลาง สถาบันวิทยบริการ จุฬาลงกรณ์มหาวิทยาลัย

บทที่ 4
ข้อสรุปและข้อเสนอแนะ

ข้อสรุป

1. อัลกอริทึมของ Modified extended Kalman filter (MEKF) เกือบทุกกรณีให้ความแม่นยำมากกว่า อัลกอริทึมของ Extanded Kalman filter (EKF) ทั้งในกรณีของ Moving observers และ Multiple fixed observers ดังที่ได้แสดงให้เห็นจากตารางในข้างต้น ซึ่งในภาพรวม ซึ่งหมายถึงการเปรียบเทียบความแม่นยำในทั้ง 2 แกนหรือ ในระนาบ XY แล้ว เราสามารถดูได้จากตารางที่แสดงในรูปของ CEP 50% ซึ่งจากตารางเราสามารถสรุปได้ดังนี้

อัลกอริทึมของ ***MEKF#3*** ให้ค่าความแม่นยำเป็น **อันดับที่ 1**
อัลกอริทึมของ ***MEKF#2*** ให้ค่าความแม่นยำเป็น **อันดับที่ 2**
อัลกอริทึมของ ***MEKF#5 และ MEKF#1*** ให้ค่าความแม่นยำใกล้เคียงกันเป็นอันดับที่ **3,4**
อัลกอริทึมของ ***MEKF#4*** ให้ค่าความแม่นยำเป็น **อันดับที่ 5**
อัลกอริทึมของ ***EKF*** ให้ค่าความแม่นยำเป็น **อันดับสุดท้าย**

2. ในส่วนของ *Torrieri's algorithm* นั้น จากตารางที่ 4.4 จะเห็นได้ว่า อัลกอริทึมของ **MKF+TR** นั้นสามารถให้ค่าความแม่นยำได้มากกว่า อัลกอริทึมของ **KF+TR** โดยเฉพาะอย่างยิ่งจะเห็นได้ชัดในแกน Y

3. ในส่วนของเวลาที่ใช้ในการประมวลผลนั้น ในแต่ละอัลกอริทึมใช้เวลาไม่เท่ากัน ซึ่งปัจจัยเรื่องของเวลานั้นมีผลต่อการนำไปประยุกต์ใช้ในการปฏิบัติจริง ซึ่งขึ้นอยู่กับภารกิจที่ได้รับว่าภารกิจนั้นๆ ต้องการความรวดเร็วมากน้อยเพียงใด ซึ่งเวลาในการประมวลผลในแต่ละวิธีสามารถสรุปได้ดังนี้

กรณีเครื่องดักรับหาทิศสัญญาณแบบเคลื่อนที่

- จำนวน 2 Observations ใช้เวลาประมวลผลโดยเฉลี่ย 11.83 วินาที
- จำนวน 9 Observations ใช้เวลาประมวลผลโดยเฉลี่ย 30.89 วินาที

สถาบันวิทยบริการ จุฬาลงกรณ์มหาวิทยาลัย

กรณีเครื่องดักรับหาทิศสัญญาณแบบอยู่กับที่ (EKF & MEKF)

- จำนวน 2 DF (Source in) ใช้เวลาประมวลผลโดยเฉลี่ย 20.58 วินาที
- จำนวน 2 DF (Source out) ใช้เวลาประมวลผลโดยเฉลี่ย 24.60 วินาที
- จำนวน 3 DF (Source in) ใช้เวลาประมวลผลโดยเฉลี่ย 37.64 วินาที
- จำนวน 3 DF (Source out) ใช้เวลาประมวลผลโดยเฉลี่ย 36.62 วินาที
- จำนวน 4 DF (Source in) ใช้เวลาประมวลผลโดยเฉลี่ย 47.08 วินาที
- จำนวน 4 DF (Source out) ใช้เวลาประมวลผลโดยเฉลี่ย 49.85 วินาที

กรณีเครื่องดักรับหาทิศสัญญาณแบบอยู่กับที่ (KF+TR & MKF+TR)

- จำนวน 2 DF (Source in) ใช้เวลาประมวลผลโดยเฉลี่ย 83.68 วินาที
- จำนวน 2 DF (Source out) ใช้เวลาประมวลผลโดยเฉลี่ย 94.27 วินาที
- จำนวน 3 DF (Source in) ใช้เวลาประมวลผลโดยเฉลี่ย 185.34 วินาที
- จำนวน 3 DF (Source in) ใช้เวลาประมวลผลโดยเฉลี่ย 204.41 วินาที
- จำนวน 4 DF (Source in) ใช้เวลาประมวลผลโดยเฉลี่ย 293.35 วินาที
- จำนวน 4 DF (Source in) ใช้เวลาประมวลผลโดยเฉลี่ย 333.70 วินาที

จากผลการสรุปเวลาในการประมวลผลข้างต้นนั้นสามารถสรุปได้ว่า อัลกอริทึมของ KF+TR และ MKF+TR นั้นใช้เวลาในการประมวลผลมากกว่าอัลกอริทึมของ EKF และ MEKF ประเภทต่างๆ และในแต่ละอัลกอริทึมนั้น จำนวนเครื่องดักรับหาทิศสัญญาณก็มีผลต่อการประมวลผลเช่นกัน กล่าวคือ จำนวนเครื่องดักรับหาทิศสัญญาณยิ่งมากก็จะยิ่งใช้เวลาในการประมวลผลมากตามไปด้วย

3. การประยุกต์ใช้ในทางปฏิบัติ ในปัจจุบันหน่วยงานที่รับผิดชอบในภารกิจ การประมาณหาที่ตั้งแหล่งกำเนิดสัญญาณนี้ได้แก่ กองร้อยปฏิบัติการสงครามอิเล็กทรอนิกส์ ของกองทัพบก และกองบัญชาการทหารสูงสุด ซึ่งในการปฏิบัติงานนั้น ในขั้นตอนของ Filter processing ใช้อัลกอริทึมของ Stanfield [2], [4] ซึ่งในด้านของฮาร์ดแวร์นั้นไม่สามารถเปลี่ยนอัลกอริทึมภายในได้ แต่ในทางปฏิบัตินั้นสามารถประยุกต์ใช้ได้โดยนำข้อมูลที่ได้จาก ขั้นตอน DOA processing มาผ่านขั้นตอนของ Filter processing โดยใช้อัลกอริทึมของ MEKF ประมวลผลทางคอมพิวเตอร์ ซึ่งประโยชน์ที่จะได้รับก็คือ ผลของการประมาณหาที่ตั้งแหล่งกำเนิดสัญญาณจะให้ความแม่นยำมากกว่าอัลกอริทึมแบบเดิม

ข้อเสนอแนะ

ในการประมาณหาที่ตั้งแหล่งกำเนิดสัญญาณนั้น สามารถพัฒนาหรือริเริ่มการพัฒนาได้อีกมากมายหลายวิธี อันได้แก่ การนำ MEKF ทั้ง 5 ประเภทมาประยุกต์ใช้กับ Torrieri's algorithm ซึ่งอาจจะให้ความแม่นยำมากกว่า MEKF ธรรมดาก็เป็นได้แต่ข้อเสียก็คือ เวลาที่ใช้ในการประมวลผลใช้เวลาค่อนข้างนานมาก จึงยังไม่เหมาะกับสถานการณ์ทางด้านการทหารที่ต้องใช้เวลาให้น้อยที่สุด พร้อมกับความแม่นยำที่มากที่สุดไปพร้อมๆกัน

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

รายการอ้างอิง

[1] Karl Spingarn " Passive position location estimation using the extended Kalman filter." *IEEE Trans. Aerospace and elect. systems.* Vol. AES-3, pp.555-567. 1987.

[2] Maj. Kaoropcroo S. " Emitter location estimation from bearing angle measurements by the extended Kalman filter method." Chulalongkorn.1996.

[3] Torrieri. D.J. " Statistical theory of passive location system." *IEEE, Trans. Aerospace and Elec. Sys.* pp.183-198. 1984.

[4] Stanfield, R.G. "Statistical theory of DF fixing" *Journal of IEE (LONDON)* 94(15)(1947): 762770.

[5] Charles K. Chui, Guanrong Chen, and Herman C. Chui. " Modified extended Kalman filtering and a real-time parallel algorithm for system parameter identification." *IEEE Trans. Automat. Contr.* Vol. 35, pp. 100-104. 1990.

[6] Pual, C.Chestnut. "Emitter Location Accuracy Using TDOA and Differential Doppler." *IEEE Trans. Aerospace.* March 1982.

[7] Bertrand, T.Fang. "Simple Solution for Hyperbolic and Related Position Fixes. *IEEE Trans. Aerospace.* September 1990.

[8] Poirot, J. and McWilliams, G. "Navigation by Back Triangulation." *IEEE, Trans. Aerospace.* March 1976.

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ภาคผนวก

โปรแกรมการประมวลผล EKF กรณีที่ 1

```
%Extended Kalman Filter in Passive Position Location
%Associated files: ipu.m, fekf.m
        clear                 % clear all variable values
xe = 15;          % Exact emitter location : (xe, ye)
ye = 15;          % Exact emitter location : (xe, ye)
x1 = 0;           % 1st DF (observer) location
y1 = 0;           %  (x1,y1)
x2 = 20;          % 2nd DF location
y2 = 0;           %   (x2,y2)
        XYk = [x1 x2  ; y1 y2 ];          % Observer location at state k
b1 = atan((ye-y1)/(xe-x1));        % Exact bearing from DF1 to the emitter
b2 = atan((ye-y2)/(xe-x2));        % Exact bearing from DF2 to the emitter
        MSD1 = 0.0175;                % Measurement noise Standard Deviation
        MSD2 = 0.0175;                % Measurement noise Standard Deviation
        MSD = [MSD1 MSD2];
DF1 = [x1 y1 b1 MSD1]';          % DF1 data in vector form
DF2 = [x2 y2 b2 MSD2]';          % DF2 data in vector form
R1 = MSD1^2;                     % Measurement noise variances
R2 = MSD2^2;                     % Measurement noise variances
        q22 = 0.025^2;
                Q = [0 0;0 q22];          % Process noise variance, can't set Q = 0,
                                            so set q22 small as possible
Phi = eye(2);                    % Transition matrix, 2x2 identity matrix
        Steady State = 1;                 % Define steady state point
                Observations = 40;                % Number of observations
                        Monte = 100;                      % Set Number of Iteration
```

```
t1 = clock;                         % Start Timer
for s = 1:Monte                     % Set loop for 100 Monte Carlo Run
    randn('seed',s);                    % Set seeds for each run
        Runs = s
[X00,CovXY] = ipu(DF1,DF2);     % Initial Position Uncertainty
P00 = diag(diag(CovXY));        % Initial Covariance Matrix
N1 = MSD1*randn(Observations,1);  % Measurement noise ~ N1(0,R1)
N2 = MSD2*randn(Observations,2);  % Measurement noise ~ N2(0,R2)
B1 = b1+N1(:,1);                % Add noises to DF1
B2 = b2+N2(:,1);                % Add noises to DF2
    Bk = [B1 B2];                   % Measurement at state k
% Start EKF algorithm
%       X(k|k) == Xkk
%       X(k-1|k-1) == Xk1k1
%       X(k|k-1) == Xkk1
Xk1k1 = X00;   % X(1|1) = X(0|0)
Pk1k1 = P00;   % P(k-1|k-1) = P(0|0)
count = 0;                  % Count for total number of observation.
for obsv = 1:Observations
    for df = 1:2
    count = count+1;                    % Increase number of observations
    yk = XYk(2,df); xk = XYk(1,df);     % observer location
    DFk = [xk yk Bk(obsv,df)]';         % DF data at state k
    [Xkk,Pkk] = fekft(DFk,Xk1k1,Pk1k1,Phi,MSD(df));  % call EKF function
% Update statistics for the next state; k = k+1
    Xk1k1 = Xkk ;
    Xkk1 = Phi*[Xk1k1];
    Pk1k1 = Pkk;
```

```
            yekk1 = Xkk1(2);
            xekk1 = Xkk1(1);
           % Compute error statistics
          ErrorX(count) = xe - Xkk(1);
          ErrorY(count) = ye - Xkk(2);
           % ErrorXY(count) = sqrt(ErrorX(count)^2 + ErrorY(count)^2) ;
          MErX(s,count) = (mean(ErrorX))^2;
          MErY(s,count) = (mean(ErrorY))^2;
          MErXY(s,count) = sqrt(MErX(s,count)+ MErY(s,count));
                   xCepkk(count) = Xkk(1);
                   yCepkk(count) = Xkk(2);
                         MxCepkk(count) = mean(xCepkk);
                         MyCepkk(count) = mean(yCepkk);
                   end % DF loop
          end   % Observation loop
  %  Steady State Average Position Error of each run
          XYcep(s,1)=mean(MxCepkk);
          XYcep(s,2)=mean(MyCepkk);
   end % Monte Loop
 % Compute Simulation time
          SimTime = etime(clock,t1)/60
AvgMErX = sum(MErX)/Monte;
for i = 1:1*Observations
  AvgAvg(i) = sum(AvgMErX(1:i))/i;
end
% Plot error graphs:
figure
OBSV = linspace(1,1*Observations,1*Observations);
plot(OBSV,AvgAvg,'r')
xlabel('Observations')
ylabel('MS Position Errors')
cep50(XYcep,[xe ye])
```

โปรแกรมการประมวลผล MEKF#1 กรณีที่ 1

```
% Modified Extended Kalman Filter in Passive Position Location
% Associated files: ipu.m, fekf.m
clear                    % clear all variable values
xe = 15;          % Exact emitter location :
ye = 15;          %  (xe,ye)
x1 = 0;           % 1st DF (observer) location
y1 = 0;           %  (x1,y1)
x2 = 20;          % 2nd DF location
y2 = 0;           %  (x2,y2)
XYk = [x1 x2 ; y1 y2];     % Observer location at state k
b1 = atan((ye-y1)/(xe-x1)); % Exact bearing from DF1 to the emitter
b2 = atan((ye-y2)/(xe-x2)); % Exact bearing from DF2 to the emitter
MSD1 = 0.0175;        % Measurement noise Standard Deviation
MSD2 = 0.0175;        % Measurement noise Standard Deviation
MSD = [MSD1 MSD2];
DF1 = [x1 y1 b1 MSD1]'; % DF1 data in vector form
DF2 = [x2 y2 b2 MSD2]'; % DF2 data in vector form
R1 = MSD1^2;              % Measurement noise variances
R2 = MSD2^2;
q22 = 0.025^2;
Q = [0 0;0 q22];      % Process noise variance, can't set Q = 0,
                        so set q22 small as possible
Phi = eye(2);         %Transition matrix, 2x2 identity matrix
SteadyState = 1;        % Define steady state point
Observations =40;       % Number of observations
Monte = 100;              % Set Number of Iteration
```

```
t1 = clock;                 % Start Timer
for s = 1:Monte             % Set loop for 100 Monte Carlo Run
    randn('seed',s);            % Set seeds for each run
    Runs = s
[X00,CovXY] = ipu(DF1,DF2);  % Initial Position Uncertainty
P00 = diag(diag(CovXY));     % Initial Covariance Matrix
N1 = MSD1*randn(Observations,1);   % Measurement noise ~ N1(0,R1)
N2 = MSD2*randn(Observations,1);   % Measurement noise ~ N2(0,R2)
B1 = b1+N1(:,1);       % Add noises to DF1
B2 = b2+N2(:,1);       % Add noises to DF2
    Bk = [B1 B2 ];         % Measurement at state k
% Start EKF algorithm
%           X(k|k) == Xkk
%         X(k-1|k-1) == Xk1k1
%          X(k|k-1) == Xkk1
Xk1k1 = X00;   % X(1|1) = X(0|0)
Pk1k1 = P00;   % P(k-1|k-1) = P(0|0)
XK1K1=X00;
PK1K1=P00;
Xk1=X00;
count = 0;   % Count for total number of observation.
for obsv = 1:Observations
for df = 1:2
   count = count+1;            % Increase number of obserations
   yk = XYk(2,df); xk = XYk(1,df);   % observer location
   DFk = [xk yk Bk(obsv,df)]';       % DF data at state k
[Xkk,Pkk] = fekft(DFk,Xk1k1,Pk1k1,Phi,MSD(df));         % call EKF function
[XKK,PKK] = fkf(DFk,XK1K1,PK1K1,Phi,MSD(df));           % call KF function
   % Update statistics for the next state; k = k+1
   XK1K1 = Xkk ;
   Xk1k1 = XKK;
```

```
            PK1PK1 = Pkk;
            Pk1k1 = PKK;
    % yekk1 = Xkk1(2);
    % xekk1 = Xkk1(1);
    % Compute error statistics
            ErrorX(count) = xe - Xkk(1);
            ErrorY(count) = ye - Xkk(2);
            ErrorXY(count) = sqrt(ErrorX(count)^2 + ErrorY(count)^2) ;
     MErX(s,count) = (mean(ErrorX))^2;
     MErY(s,count) = (mean(ErrorY))^2;
     MErXY(s,count) =sqrt(MErX(s,count)+ MErY(s,count));
            xCepkk(count) = Xkk(1);
            yCepkk(count) = Xkk(2);
            MxCepkk(count) = mean(xCepkk);
            MyCepkk(count) = mean(yCepkk);
     end % DF loop
            end   % Observation loop
 %  Steady State Average Position Error of each run
  TotalAvg(s) = mean(NEy); %  Grand Mean of Average Position Errors
     XYcep(s,1)=mean(MxCepkk);
     XYcep(s,2)=mean(MyCepkk);
end % Monte Loop
% Compute Simulation time
     SimTime = etime(clock,t1)/60
AvgMErX = sum(MErX)/Monte;
for i = 1:1*Observations
AvgAvg(i) = sum(AvgMErX(1:i))/i;
End
%Polt MS error graphs:
     OBSV = linspace(1,1*Observations,1*Observations);
     plot(OBSV,AvgAvg,'k')
     cep50(XYcep,[xe ye])
```

โปรแกรมการประมวลผล MEKF#2 กรณีที่ 1

```
% Modified Extended Kalman Filter in Passive Position Location
% Associated files: ipu.m, fekf.m
        clear                   % clear all variable values
xe = 15;          % Exact emitter location :
ye = 15;          %  (xe,ye)
x1 = 0;           % 1st DF (observer) location
y1 = 0;           %  (x1,y1)
x2 = 20;          % 2nd DF location
y2 = 0;           %  (x2,y2)
        XYk = [x1 x2  ; y1 y2];     % Observer location at state k
b1 = atan((ye-y1)/(xe-x1)); % Exact bearing from DF1 to the emitter
b2 = atan((ye-y2)/(xe-x2)); % Exact bearing from DF2 to the emitter
        MSD1 = 0.0175;        % Measurement noise Standard Deviation
        MSD2 = 0.0175;        % Measurement noise Standard Deviation
        MSD = [MSD1 MSD2];
DF1 = [x1 y1 b1 MSD1]'; % DF1 data in vector form
DF2 = [x2 y2 b2 MSD2]'; % DF2 data in vector form
R1 = MSD1^2;            % Measurement noise variances
R2 = MSD2^2;
        q22 = 0.025^2;
              Q = [0 0;0 q22];      % Process noise variance, can't set Q = 0,
                                      so set q22 small as possible
Phi = eye(2);         %Transition matrix, 2x2 identity matrix
        SteadyState = 1;        % Define steady state point
              Observations =40;       % Number of observations
                     Monte = 100;            % Set Number of Iteration
```

```
t1 = clock;                % Start Timer
for s = 1:Monte            % Set loop for 100 Monte Carlo Run
    randn('seed',s);           % Set seeds for each run
    Runs = s
[X00,CovXY] = ipu(DF1,DF2);  % Initial Position Uncertainty
P00 = diag(diag(CovXY));     % Initial Covariance Matrix
N1 = MSD1*randn(Observations,1);  % Measurement noise ~ N1(0,R1)
N2 = MSD2*randn(Observations,1);  % Measurement noise ~ N2(0,R2)
B1 = b1+N1(:,1);     % Add noises to DF1
B2 = b2+N2(:,1);     % Add noises to DF2
    Bk = [B1 B2 ];       % Measurement at state k
% Start EKF algorithm
%           X(k|k) == Xkk
%         X(k-1|k-1) == Xk1k1
%          X(k|k-1) == Xkk1
Xk1k1 = X00;   % X(1|1) = X(0|0)
Pk1k1 = P00;   % P(k-1|k-1) = P(0|0)
XK1K1=X00;
PK1K1=P00;
Xk1=X00;
count = 0;    % Count for total number of observation.
for obsv = 1:Observations
for df = 1:2
count = count+1;            % Increase number of obserations
yk = XYk(2,df); xk = XYk(1,df);    % observer location
DFk = [xk yk Bk(obsv,df)]';        % DF data at state k
[Xkk,Pkk] = fekft(DFk,Xk1k1,Pk1k1,Phi,MSD(df));          % call EKF function
[XKK,PKK] = fekfs(DFk,XK1K1,PK1K1,Phi,MSD(df));                % call KF function
% Update statistics for the next state; k = k+1
XK1K1 = Xkk ;
Xk1k1 = XKK;
```

```
            PK1PK1 = Pkk;
            Pk1k1 = PKK;
        % yekk1 = Xkk1(2);
        % xekk1 = Xkk1(1);
        % Compute error statistics
            ErrorX(count) = xe - Xkk(1);
            ErrorY(count) = ye - Xkk(2);
            ErrorXY(count) = sqrt(ErrorX(count)^2 + ErrorY(count)^2),;
        MErX(s,count) = (mean(ErrorX))^2;
        MErY(s,count) = (mean(ErrorY))^2;
        MErXY(s,count) =sqrt(MErX(s,count)+ MErY(s,count));
            xCepkk(count) = Xkk(1);
            yCepkk(count) = Xkk(2);
            MxCepkk(count) = mean(xCepkk);
            MyCepkk(count) = mean(yCepkk);
        end % DF loop
            end   % Observation loop
  %  Steady State Average Position Error of each run
   TotalAvg(s) = mean(NEy); %  Grand Mean of Average Position Errors
        XYcep(s,1)=mean(MxCepkk);
        XYcep(s,2)=mean(MyCepkk);
end % Monte Loop
% Compute Simulation time
        SimTime = etime(clock,t1)/60
AvgMErX = sum(MErX)/Monte;
for i = 1:1*Observations
AvgAvg(i) = sum(AvgMErX(1:i))/i;
End
%Polt MS error graphs:
        OBSV = linspace(1,1*Observations,1*Observations);
        plot(OBSV,AvgAvg,'c')
        cep50(XYcep,[xe ye])
```

โปรแกรมการประมวลผล MEKF#3 กรณีที่ 1

```
% Modified Extended Kalman Filter in Passive Position Location
% Associated files: ipu.m, fekf.m
        clear                    % clear all variable values
xe = 15;          % Exact emitter location :
ye = 15;          %  (xe,ye)
x1 = 0;           % 1st DF (observer) location
y1 = 0;           %  (x1,y1)
x2 = 20;          % 2nd DF location
y2 = 0;           %  (x2,y2)
        XYk = [x1 x2  ; y1 y2];    % Observer location at state k
b1 = atan((ye-y1)/(xe-x1)); % Exact bearing from DF1 to the emitter
b2 = atan((ye-y2)/(xe-x2)); % Exact bearing from DF2 to the emitter
        MSD1 = 0.0175;        % Measurement noise Standard Deviation
        MSD2 = 0.0175;        % Measurement noise Standard Deviation
        MSD = [MSD1 MSD2];
DF1 = [x1 y1 b1 MSD1]'; % DF1 data in vector form
DF2 = [x2 y2 b2 MSD2]'; % DF2 data in vector form
R1 = MSD1^2;            % Measurement noise variances
R2 = MSD2^2;
        q22 = 0.025^2;
                Q = [0 0;0 q22];    % Process noise variance, can't set Q = 0,
                                      so set q22 small as possible
Phi = eye(2);        %Transition matrix, 2x2 identity matrix
        SteadyState = 1;        % Define steady state point
                Observations =40;    % Number of observations
                        Monte = 100;            % Set Number of Iteration
```

```
t1 = clock;              % Start Timer
for s = 1:Monte          % Set loop for 100 Monte Carlo Run
    randn('seed',s);          % Set seeds for each run
    Runs = s
[X00,CovXY] = ipu(DF1,DF2);  % Initial Position Uncertainty
P00 = diag(diag(CovXY));     % Initial Covariance Matrix
N1 = MSD1*randn(Observations,1);   % Measurement noise ~ N1(0,R1)
N2 = MSD2*randn(Observations,1);   % Measurement noise ~ N2(0,R2)
B1 = b1+N1(:,1);       % Add noises to DF1
B2 = b2+N2(:,1);       % Add noises to DF2
    Bk = [B1 B2 ];        % Measurement at state k
% Start EKF algorithm
%          X(k|k) == Xkk
%        X(k-1|k-1) == Xk1k1
%         X(k|k-1) == Xkk1
Xk1k1 = X00;   % X(1|1) = X(0|0)
Pk1k1 = P00;   % P(k-1|k-1) = P(0|0)
XK1K1=X00;
PK1K1=P00;
Xk1=X00;
count = 0;     % Count for total number of observation.
for obsv = 1:Observations
for df = 1:2
    count = count+1;          % Increase number of obserations
    yk = XYk(2,df); xk = XYk(1,df);     % observer location
    DFk = [xk yk Bk(obsv,df)]';        % DF data at state k
[Xkk,Pkk] = fekft(DFk,Xk1k1,Pk1k1,Phi,MSD(df));          % call EKF function
[XKK,PKK] = fekft(DFk,XK1K1,PK1K1,Phi,MSD(df));               % call KF function
% Update statistics for the next state; k = k+1
    XK1K1 = Xkk ;
    Xk1k1 = XKK;
```

```
                PK1PK1 = Pkk;
                Pk1k1 = PKK;
        % yekk1 = Xkk1(2);
        % xekk1 = Xkk1(1);
        % Compute error statistics
                ErrorX(count) = xe - Xkk(1);
                ErrorY(count) = ye - Xkk(2);
                ErrorXY(count) = sqrt(ErrorX(count)^2 + ErrorY(count)^2) ;
         MErX(s,count) = (mean(ErrorX))^2;
         MErY(s,count) = (mean(ErrorY))^2;
         MErXY(s,count) =sqrt(MErX(s,count)+ MErY(s,count));
                xCepkk(count) = Xkk(1);
                yCepkk(count) = Xkk(2);
                MxCepkk(count) = mean(xCepkk);
                MyCepkk(count) = mean(yCepkk);
        end % DF loop
                end   % Observation loop
  %  Steady State Average Position Error of each run
   TotalAvg(s) = mean(NEy); %  Grand Mean of Average Position Errors
        XYcep(s,1)=mean(MxCepkk);
        XYcep(s,2)=mean(MyCepkk);
end % Monte Loop
% Compute Simulation time
        SimTime = etime(clock,t1)/60
AvgMErX = sum(MErX)/Monte;
for i = 1:1*Observations
AvgAvg(i) = sum(AvgMErX(1:i))/i;
End
%Polt MS error graphs:
        OBSV = linspace(1,1*Observations,1*Observations);
        plot(OBSV,AvgAvg,'m')
        cep50(XYcep,[xe ye])
```

โปรแกรมการประมวลผล MEKF#4 กรณีที่ 1

```
% Modified Extended Kalman Filter in Passive Position Location
% Associated files: ipr.m, feki.m
        clear                   % clear all variable values
xe = 15;            % Exact emitter location :
ye = 15;            %  (xe,ye)
x1 = 0;             % 1st DF (observer) location
y1 = 0;             %  (x1,y1)
x2 = 20;            % 2nd DF location
y2 = 0;             %  (x2,y2)
        XYk = [x1 x2  ; y1 y2];     % Observer location at state k
b1 = atan((ye-y1)/(xe-x1)); % Exact bearing from DF1 to the emitter
b2 = atan((ye-y2)/(xe-x2)); % Exact bearing from DF2 to the emitter
        MSD1 = 0.0175;          % Measurement noise Standard Deviation
        MSD2 = 0.0175;          % Measurement noise Standard Deviation
        MSD = [MSD1 MSD2];
DF1 = [x1 y1 b1 MSD1]'; % DF1 data in vector form
DF2 = [x2 y2 b2 MSD2]'; % DF2 data in vector form
R1 = MSD1^2;            % Measurement noise variances
R2 = MSD2^2;
        q22 = 0.025^2;
                Q = [0 0;0 q22];      % Process noise variance, can't set Q = 0,
                                        so set q22 small as possible
Phi = eye(2);           %Transition matrix, 2x2 identity matrix
        SteadyState = 1;        % Define steady state point
                Observations =40;       % Number of observations
                        Monte = 100;            % Set Number of Iteration
```

```
t1 = clock;                    % Start Timer
for s = 1:Monte                % Set loop for 100 Monte Carlo Run
    randn('seed',s);           % Set seeds for each run
    Runs = s
[X00,CovXY] = ipu(DF1,DF2);  % Initial Position Uncertainty
P00 = diag(diag(CovXY));     % Initial Covariance Matrix
N1 = MSD1*randn(Observations,1);  % Measurement noise ~ N1(0,R1)
N2 = MSD2*randn(Observations,1);  % Measurement noise ~ N2(0,R2)
B1 = b1+N1(:,1);      % Add noises to DF1
B2 = b2+N2(:,1);      % Add noises to DF2
    Bk = [B1 B2 ];        % Measurement at state k
% Start EKF algorithm
%            X(k|k) == Xkk
%         X(k-1|k-1) == Xk1k1
%           X(k|k-1) == Xkk1
Xk1k1 = X00;   % X(1!1) = X(0|0)
Pk1k1 = P00;     % P(k-1|k-1) = P(0|0)
XK1K1=X00;
PK1K1=P00;
Xk1=X00;
count = 0;    % Count for total number of observation.
for obsv = 1:Observations
for df = 1:2
   count = count+1;            % Increase number of obserations
   yk = XYk(2,df); xk = XYk(1,df);     % observer location
   DFk = [xk yk Bk(obsv,df)]';          % DF data at state k
[Xkk,Pkk] = fekfs(DFk,Xk1k1,Pk1k1,Phi,MSD(df));          % call EKF function
[XKK,PKK] = fekfs(DFk,XK1K1,PK1K1,Phi,MSD(df));                % call KF function
 % Update statistics for the next state; k = k+1
    XK1K1 = Xkk ;
    Xk1k1 = XKK;
```

```
PK1PK1 = Pkk;
Pk1k1 = PKK;
% yekk1 = Xkk1(2);
% xekk1 = Xkk1(1);
% Compute error statistics
ErrorX(count) = xe - Xkk(1);
ErrorY(count) = ye - Xkk(2);
ErrorXY(count) = sqrt(ErrorX(count)^2 + ErrorY(count)^2) .
MErX(s,count) = (mean(ErrorX))^2;
MErY(s,count) = (mean(ErrorY))^2;
MErXY(s,count) =sqrt(MErX(s,count)+ MErY(s,count));
xCepkk(count) = Xkk(1);
yCepkk(count) = Xkk(2);
MxCepkk(count) = mean(xCepkk);
MyCepkk(count) = mean(yCepkk);
end % DF loop
end   % Observation loop
%  Steady State Average Position Error of each run
TotalAvg(s) = mean(NEy); %  Grand Mean of Average Position Errors
XYcep(s,1)=mean(MxCepkk);
XYcep(s,2)=mean(MyCepkk);
end % Monte Loop
% Compute Simulation time
SimTime = etime(clock,t1)/60
AvgMErX = sum(MErX)/Monte;
for i = 1:1*Observations
AvgAvg(i) = sum(AvgMErX(1:i))/i;
End
%Polt MS error graphs:
OBSV = linspace(1,1*Observations,1*Observations);
plot(OBSV,AvgAvg,'g')
cep50(XYcep,[xe ye])
```

โปรแกรมการประมวลผล MEKF#5 กรณีที่ 1

```
% Modified Extended Kalman Filter in Passive Position Location
% Associated files: ipu.m, fekf.m
        clear                   % clear all variable values
xe = 15;        % Exact emitter location :
ye = 15;        %  (xe,ye)
x1 = 0;         % 1st DF (observer) location
y1 = 0;         %  (x1,y1)
x2 = 20;        % 2nd DF location
y2 = 0;         %  (x2,y2)
        XYk = [x1 x2  ; y1 y2];     % Observer location at state k
b1 = atan((ye-y1)/(xe-x1)); % Exact bearing from DF1 to the emitter
b2 = atan((ye-y2)/(xe-x2)); % Exact bearing from DF2 to the emitter
        MSD1 = 0.0175;        % Measurement noise Standard Deviation
        MSD2 = 0.0175;        % Measurement noise Standard Deviation
        MSD = [MSD1 MSD2];
DF1 = [x1 y1 b1 MSD1]'; % DF1 data in vector form
DF2 = [x2 y2 b2 MSD2]'; % DF2 data in vector form
R1 = MSD1^2;            % Measurement noise variances
R2 = MSD2^2;
        q22 = 0.025^2;
                Q = [0 0;0 q22];      % Process noise variance, can't set Q = 0,
                                        so set q22 small as possible
Phi = eye(2);        %Transition matrix, 2x2 identity matrix
        SteadyState = 1;        % Define steady state point
                Observations =40;       % Number of observations
                        Monte = 100;            % Set Number of Iteration
```

```
t1 = clock;                    % Start Timer
for s = 1:Monte                % Set loop for 100 Monte Carlo Run
    randn('seed',s);           % Set seeds for each run
    Runs = s
[X00,CovXY] = ipu(DF1,DF2);  % Initial Position Uncertainty
P00 = diag(diag(CovXY));      % Initial Covariance Matrix
N1 = MSD1*randn(Observations,1);  % Measurement noise ~ N1(0,R1)
N2 = MSD2*randn(Observations,1);  % Measurement noise ~ N2(0,R2)
B1 = b1+N1(:,1);        % Add noises to DF1
B2 = b2+N2(:,1);        % Add noises to DF2
    Bk = [B1 B2 ];         % Measurement at state k
% Start EKF algorithm
%           X(k|k) == Xkk
%        X(k-1|k-1) == Xk1k1
%          X(k|k-1) == Xkk1
Xk1k1 = X00;   % X(1|1) = X(0|0)
Pk1k1 = P00;    % P(k-1|k-1) = P(0|0)
XK1K1=X00;
PK1K1=P00;
Xk1=X00;
count = 0;    % Count for total number of observation.
for obsv = 1:Observations
for df = 1:2
count = count+1;              % Increase number of obserations
yk = XYk(2,df); xk = XYk(1,df);    % observer location
DFk = [xk yk Bk(obsv,df)]';        % DF data at state k
[Xkk,Pkk] = fekfs(DFk,Xk1k1,Pk1k1,Phi,MSD(df));            % call EKF function
[XKK,PKK] = fekft(DFk,XK1K1,PK1K1,Phi,MSD(df)),                   % call KF function
% Update statistics for the next state; k = k+1
XK1K1 = Xkk ;
Xk1k1 = XKK;
```

```
                PK1PK1 = Pkk;
                Pk1k1 = PKK;
        % yekk1 = Xkk1(2);
        % xekk1 = Xkk1(1);
        % Compute error statistics
                ErrorX(count) = xe - Xkk(1);
                ErrorY(count) = ye - Xkk(2);
                ErrorXY(count) = sqrt(ErrorX(count)^2 + ErrorY(count)^2) ;
         MErX(s,count) = (mean(ErrorX))^2;
         MErY(s,count) = (mean(ErrorY))^2;
         MErXY(s,count) =sqrt(MErX(s,count)+ MErY(s,count));
                xCepkk(count) = Xkk(1);
                yCepkk(count) = Xkk(2);
                MxCepkk(count) = mean(xCepkk);
                MyCepkk(count) = mean(yCepkk);
         end % DF loop
                end   % Observation loop
  %  Steady State Average Position Error of each run
   TotalAvg(s) = mean(NEy); %  Grand Mean of Average Position Errors
        XYcep(s,1)=mean(MxCepkk);
        XYcep(s,2)=mean(MyCepkk);
end % Monte Loop
% Compute Simulation time
        SimTime = etime(clock,t1)/60
AvgMErX = sum(MErX)/Monte;
for i = 1:1*Observations
AvgAvg(i) = sum(AvgMErX(1:i))/i;
End
%Polt MS error graphs:
        OBSV = linspace(1,1*Observations,1*Observations);
        plot(OBSV,AvgAvg,'b')
        cep50(XYcep,[xe ye])
```

Associated files : fekft.m

```
function [Xkk,Pkk,Pkk1,Skk1,vk] = fekft(DFk,Xk1k1,Pk1k1,Phi,MSD)
%FEKF   Extended Kalman Filter Function (Bearing angle function)
%Initialization
        R = MSD^2;          % Measurement noise variance
        q22 = 0.025^2;
Q = [0 0;0 q22];      % Process noise variance, can't set Q = 0,
                        ;so set q22 small as possible
Bk = DFk(3);          % Measurement at state k ; contaminated by noise
        % Start EKF algorithm
        %       X(k|k) == Xkk
        %       X(k-1|k-1) == Xk1k1
        %       X(k|k-1) == Xkk1
Xkk1 = Phi*Xk1k1; % X(1|0) = Phi*X(0|0);
yekk1 = Xkk1(2); xekk1 = Xkk1(1);
yk = DFk(2); xk = DFk(1);  % observer location
u = (yekk1 - yk)/(xekk1-xk);
        hXkk1 = atan(u);   % h(X(k|k-1))
        H11 = -u*((1+u^2) * (xekk1 - xk))^(-1);
        H12 = ((1+u^2) * (xekk1 - xk))^(-1);
                HXkk1 = [H11 H12];  % H(X(k|k-1))
Pkk1 = Phi*Pk1k1*Phi' + Q;    % P(k|k-1) , error covariance extrapolation
Skk1 = HXkk1*Pkk1*HXkk1'+R; % noise corresponding covariance extrapolation
vk = Bk - hXkk1;               % Estimated noise
K = Pkk1*HXkk1'*Skk1^(-1);       % Kalman gain
Xkk = Xkk1+K*(Bk - hXkk1);              % State estimate equation
Pkk = (eye(2) - K*HXkk1) * Pkk1;          % Error covariance update
```

Associated files : fekfs.m

```
function [Xkk,Pkk,Pkk1,Skk1,vk] -- fekft(DFk,Xk1k1,Pk1k1,Phi,MSD)
%FEKFS   Extended Kalman Filter Function (Slope function)
%Initialization
        R = MSD^2;           % Measurement noise variance
        q22 = 0.025^2;
Q = [0 0;0 q22];      % Process noise variance, can't set Q = 0,
                         so set q22 small as possible
Bk = DFk(3);          % Measurement at state k ; contaminated by noise
        % Start EKF algorithm
        %       X(k|k) == Xkk
        %       X(k-1|k-1) == Xk1k1
        %       X(k|k-1) == Xkk1
Xkk1 = Phi*Xk1k1; % X(1|0) = Phi*X(0|0);
yekk1 = Xkk1(2); xekk1 = Xkk1(1);
yk = DFk(2); xk = DFk(1);  % observer location
u = (yekk1 - yk)/(xekk1-xk);
        hXkk1 = atan(u);   % h(X(k|k-1))
        H11= -u/(xekk1-xk)/xekk1;
        H12=1/(xekk1-xk)/yekk1;
                HXkk1 = [H11 H12];  % H(X(k|k-1))
Pkk1 = Phi*Pk1k1*Phi' + Q;    % P(k|k-1) , error covariance extrapolation
Skk1 = HXkk1*Pkk1*HXkk1'+R; % noise corresponding covariance extrapolation
vk = Bk - hXkk1;               % Estimated noise
K = Pkk1*HXkk1'*Skk1^(-1);       % Kalman gain
Xkk = Xkk1+K*(Bk - hXkk1);             % State estimate equation
Pkk = (eye(2) - K*HXkk1) * Pkk1;         % Error covariance update
```

Associated files : fkf.m

```
function [Xkk,Pkk,Pkk1,Skk1,vk] = fekft(DFk,Xk1k1,Pk1k1,Phi,MSD)
%FKF   Kalman Filter Function
%Initialization
        R = MSD^2;          % Measurement noise variance
        q22 = 0.025^2;
Q = [0 0;0 q22];     % Process noise variance, can't set Q = 0,
                        so set q22 small as possible
Bk = DFk(3);          % Measurement at state k ; contaminated by noise
        % Start EKF algorithm
        %       X(klk) == Xkk
        %       X(k-1lk-1) == Xk1k1
        %       X(klk-1) == Xkk1
Xkk1 = Phi*Xk1k1; % X(1l0) = Phi*X(0l0);
yekk1 = Xkk1(2); xekk1 = Xkk1(1);
yk = DFk(2); xk = DFk(1);  % observer location
u = (yekk1 - yk)/(xekk1-xk);
        hXkk1 = atan(u);   % h(X(klk-1))
        H11 = 0.5*atan(u)/xekk1;
        H12 = 0.5*atan(u)/yekk1;
                HXkk1 = [H11 H12];  % H(X(klk-1))
Pkk1 = Phi*Pk1k1*Phi' + Q;   % P(klk-1) , error covariance extrapolation
Skk1 = HXkk1*Pkk1*HXkk1'+R; % noise corresponding covariance extrapolation
vk = Bk - hXkk1;              % Estimated noise
K = Pkk1*HXkk1'*Skk1^(-1);         % Kalman gain
Xkk = Xkk1+K*(Bk - hXkk1);              % State estimate equation
Pkk = (eye(2) - K*HXkk1) * Pkk1;         % Error covariance update
```

Associated files : ipu.m

```
function [X00,CovXY,NS] = ipu(DF1,DF2)
        xe=15;
        ye=15;
x1 = 0;         % store DF1 X coordinate
x2 = 20;        % store DF2 X coordinate
y1 = 0;         % store DF1 Y coordinate
y2 = 0;         % store DF2 Y coordinate

        b1 = atan((ye-y1)/(ye-x1));     % store DF1 exact bearing to the emitter
        b2 = atan((ye-y2)/(ye-x2));     % store DF2 exact bearing to the emitter
msd1 =0.0175;  % store DFs' measurement standard deviations
msd2 =0.0175;  % store DFs' measurement standard deviations
        NS1 = msd1*randn;
        NS2 = msd2*randn;
                NS = [NS1;NS2];

beta1 = b1 + NS1; % Add noise to bearing
beta2 = b2 + NS2; % Add noise to bearing
theta = abs(atan2((y2-y1),(x2-x1))); % DF line slope
a = sqrt((x2-x1)^2 + (y2-y1)^2); % Distance between two DF's
alpha1 = beta1 +  theta; % Axis rotation
alpha2 = beta2 + theta;
d = (sin(alpha1 - alpha2))^(-2);

        F11 = -a*d*sin(alpha2)*cos(alpha2);
        F12 = a*d*sin(alpha1)*cos(alpha1);
        F21 = -a*d*(sin(alpha2))^2;
        F22 = a*d*(sin(alpha1))^2;
                F = [F11 F12; F21 F22];
```

โปรแกรมการประมวลผล MKF+TR

```
% TORRIERI's Algorithm: Passive Location Finding System
% Associated Files: refpoint.m, ipur.m
% Variable Initialization
clear
   % Real Emitter's Location
xe = 15;          % Emitter's X coordinate
ye = 15;          % Emitter's Y coordinate
  % DFs' Locations
x1 = 0 ; y1 = 0;  % DF1's
x2 = 20 ; y2 = 0;  % DF2's
   % Real Bearing Angles
rba1 = atan2((ye-y1),(xe-x1));
rba2 = atan2((ye-y2),(xe-x2));
   % Statistical Parameters' of DFs
Observations = 40;
Monte =100;
for s = 1: Monte;
std1 = 0.0175;        % DF1's Standard deviation (In radian)
std2 = 0.0175;        % DF2's Standard deviation "-------------"

randn('seed',s);    % Set Seed of N~(0,1) Generator as 7
Runs=s
count =0;
for obsv = 1:Observations
 for df = 1:2
  count = count+1;
```

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

```
n1 = std1*randn;     % Noise for DF1
n2 = std2*randn;     % Noise for DF2

bn1 = rba1 + n1;     % Add noise to DF1
bn2 = rba2 + n2;     % Add noise to DF2

DF1 = [ x1;y1;bn1];
DF2 = [ x2;y2;bn2];

% Solve for a Reference point
XYref = y;
% Start Torrieri's Algoritm Kernel
xo = XYref(1); yo = XYref(2);
br1 = bn1 - atan2(yo-y1,xo-x1);
br2 = bn2 - atan2(yo-y2,xo-x2);

Do1 = sqrt((xo-x1)^2 + (yo-y1)^2);
Do2 = sqrt((xo-x2)^2 + (yo-y2)^2);

sinbo1 = (yo-y1)/Do1;       cosbo1 = (xo-x1)/Do1;
sinbo2 = (yo-y2)/Do2;       cosbo2 = (xo-x2)/Do2;

mu = cosbo1^2/(Do1*std1)^2+cosbo2^2/(Do2*std2)^2;
lm  = sinbo1^2/(Do1*std2)^2+sinbo2^2/(Do2*std2)^2;
nu  = sinbo1*cosbo1/(Do1*std1)^2 + sinbo2*cosbo2/(Do2*std2)^2 ;
% Compute for estimates, Xe, and Ye

sx1 = br1 * (nu*cosbo1-mu*sinbo1)/(Do1*std1^2);
sx2 = br2 * (nu*cosbo2-mu*sinbo2)/(Do2*std2^2);
```

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

```
sy1 = br1 * (lm*cosbo1-nu*sinbo1)/(Do1*std1^2);
sy2 = br2 * (lm*cosbo2-nu*sinbo2)/(Do2*std2^2);

Xe = xo + 1/(mu*lm-nu^2) * (sx1+sx2);
Ye = yo + 1/(mu*lm-nu^2) * (sy1+sy2);

ERR(obsv) = (xe-Xe);
MErX(s,count) =(mean(ERR))^2;
AvgMErX = sum(MErX(s,count))/Monte;
avgERR(obsv) =sum(AvgMErX)/Observations;
end  % End Observation Loop
end
end
OBSV = linspace(1,1*Observations,1*Observations);
plot(OBSV,avgERR,'r')
xlabel('Observations')
ylabel('MS Position Errors')
```

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

หอสมุดกลาง สถาบันวิทยบริการ จุฬาลงกรณ์มหาวิทยาลัย

ประวัติผู้เขียน

ร้อยเอก เอกพล จูฑะพันธุ์ เกิดเมื่อวันพฤหัสบดีที่ 31 พฤษภาคม พ.ศ. 2516 ที่จังหวัดกรุงเทพมหานคร สำเร็จการศึกษาปริญญาวิทยาศาสตรบัณฑิต (สาขา วิศวกรรมไฟฟ้าสื่อสาร) จากโรงเรียนนายร้อยพระจุลจอมเกล้า รุ่นที่ 42 เมื่อปี พ.ศ. 2538 และเข้ารับการศึกษาต่อ ในหลักสูตรวิศวกรรมศาสตรมหาบัณฑิต สาขา วิศวกรรมไฟฟ้า ภาควิชา วิศวกรรมไฟฟ้า จุฬาลงกรณ์มหาวิทยาลัย เมื่อปี พ.ศ. 2539 ปัจจุบันปฏิบัติราชการเป็น รักษาราชการ ผู้บังคับกองร้อยวิทยุและศูนย์ข่าว กองพันทหารสื่อสารที่ ๑ รักษาพระองค์ กรุงเทพมหานคร

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย