

การร่วมกันระบุตำแหน่งโดยใช้ข้อมูลเชิงมุม



นายสรวิชัย รัตนพานิชกุล

# สถาบันวิทยบริการ จุฬาลงกรณ์มหาวิทยาลัย

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต

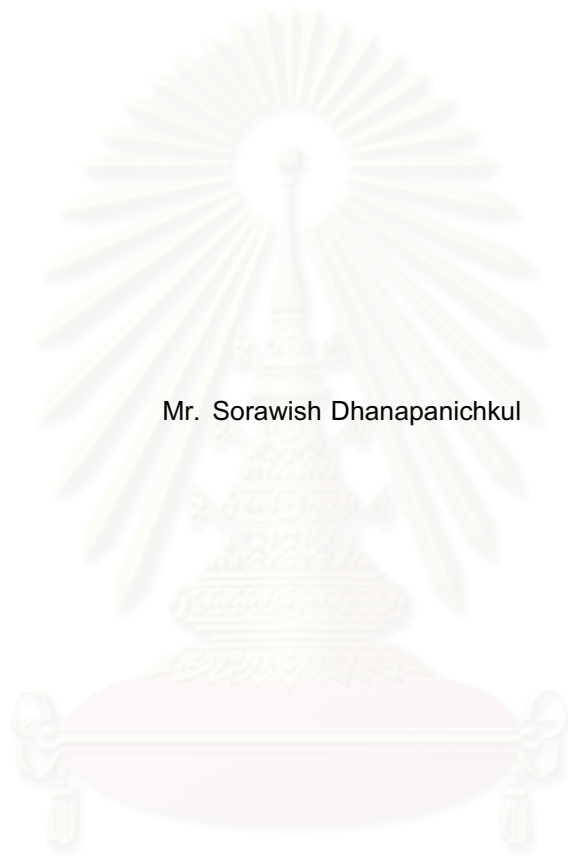
สาขาวิชาวิศวกรรมคอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย

ปีการศึกษา 2550

ลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย

COOPERATIVE LOCALIZATION USING ANGULAR MEASURES



Mr. Sorawish Dhanapanichkul

สถาบันวิทยบริการ  
จุฬาลงกรณ์มหาวิทยาลัย

A Thesis Submitted in Partial Fulfillment of the Requirements  
for the Degree of Master of Engineering Program in Computer Engineering

Department of Computer Engineering

Faculty of Engineering

Chulalongkorn University

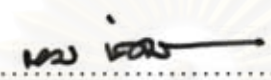
Academic Year 2007

Copyright of Chulalongkorn University

หัวข้อวิทยานิพนธ์ การร่วมกันระบุตำแหน่งโดยใช้ข้อมูลเชิงมุม  
โดย นายสรวิชัย ธนพานิชกุล  
สาขาวิชา วิศวกรรมคอมพิวเตอร์  
อาจารย์ที่ปรึกษา อาจารย์ ดร.อรรถวิทย์ สุดแสง

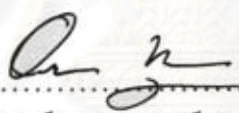
---

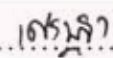
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้บัณฑิตวิทยาลัยฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรบัณฑิต

  
..... คณบดีคณะวิศวกรรมศาสตร์  
(รองศาสตราจารย์ ดร.บุญสม เลิศธีรวัฒน์)

คณะกรรมการสอบวิทยานิพนธ์

  
..... ประธานกรรมการ  
(ผู้ช่วยศาสตราจารย์ ดร.เฉลิมเอก อินทนากรวิวัฒน์)

  
..... อาจารย์ที่ปรึกษา  
(อาจารย์ ดร.อรรถวิทย์ สุดแสง)

  
..... กรรมการ  
(อาจารย์ ดร.เศรษฐา ปานงาม)

  
..... กรรมการ  
(อาจารย์ ดร. วรเศรษฐ สุวรรณิก)

สภามหาวิทยาลัย  
จุฬาลงกรณ์มหาวิทยาลัย



สรวิชัย ชนพานิชกุล : การร่วมกันระบุตำแหน่งโดยใช้ข้อมูลเชิงมุม (COOPERATIVE LOCALIZATION USING ANGULAR MEASURES). อาจารย์ที่ปรึกษา : อ. ดร. อรรถวิทย์ สุดแสง, 73 หน้า.

วิทยานิพนธ์ฉบับนี้ นำเสนอ ขั้นตอนวิธี เพื่อใช้ในการแก้ปัญหาการร่วมกันระบุตำแหน่งของกลุ่มหุ่นยนต์โดยที่หุ่นยนต์แต่ละตัวจะมีตัวรับรู้สองประเภทคือ ประเภทแรกใช้หาทิศทางที่มีหุ่นยนต์ตัวอื่นๆ อยู่แต่ไม่รู้ว่าในทิศทางนั้นเป็นหุ่นยนต์ตัวใด และอย่างที่สองคือ เข็มทิศใช้เพื่อหาทิศทางอ้างอิง โดยในวิทยานิพนธ์นี้ได้ นำเสนอ ขั้นตอนวิธีไว้สอง ขั้นตอนวิธี ขั้นตอนวิธีแรกอาศัยคุณสมบัติทางเรขาคณิตของกลุ่มหุ่นยนต์เพื่อใช้ในการหาค่าตอบ ส่วนขั้นตอนวิธีที่สองใช้การเปลี่ยนปัญหาให้กลายเป็นปัญหาการหาค่าเหมาะที่สุดเชิงคอนเวกซ์แล้วทำการแก้ปัญหาแทนสำหรับสภาพแวดล้อมที่ไม่มีความคลาดเคลื่อนในการวัด ขั้นตอนวิธีแรกได้แสดงให้เห็นว่าความน่าจะเป็นที่ทำงานได้ถูกต้องจะสูงในช่วงที่จำนวนหุ่นยนต์น้อยๆ และจะลดลงเมื่อจำนวนหุ่นยนต์เพิ่มขึ้น ส่วนขั้นตอนวิธีที่สองนั้นสามารถทำงานได้อย่างถูกต้องเสมอและในสภาพแวดล้อมที่มีความคลาดเคลื่อนในการวัด มีเพียงขั้นตอนวิธีที่สองเท่านั้นที่ได้ทำการทดลองและผลการทดลองแสดงให้เห็นว่าเมื่อจำนวนหุ่นยนต์เพิ่มมากขึ้นความคลาดเคลื่อนจากค่าตอบจะลดลงเรื่อยๆ และยิ่งไปกว่านั้นจำนวนรอบที่ใช้ในการทำงานก็ลดลงตามไปด้วย



## สถาบันวิทยบริการ จุฬาลงกรณ์มหาวิทยาลัย

ภาควิชา ..... วิศวกรรมคอมพิวเตอร์ .....  
สาขาวิชา ..... วิศวกรรมคอมพิวเตอร์ .....  
ปีการศึกษา ..... 2550 .....

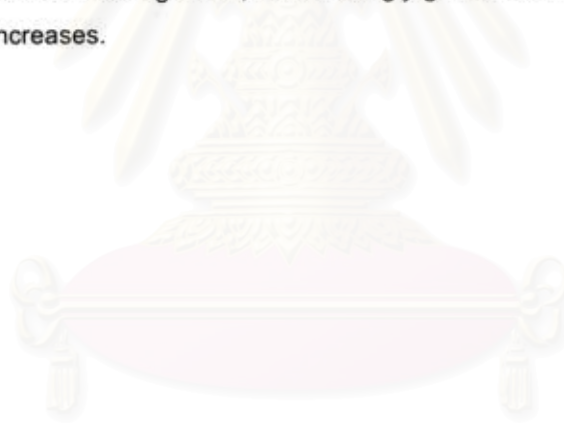
ลายมือชื่อนิสิต .....  .....  
ลายมือชื่ออาจารย์ที่ปรึกษา ..  ..

##4870506621 : MAJOR COMPUTER ENGINEERING

KEYWORDS : MOBILE ROBOT/COOPERATIVE LOCALIZATION/ ANGULAR MEASURES/  
OMNIDIRECTIONAL CAMERA

SORAWISH DHANAPANICHKUL : COOPERATIVE LOCALIZATION USING ANGULAR  
MEASURES. THESIS ADVISOR : ATTAWITH SUDSANG, Ph.D., 73 pp.

This thesis proposes two algorithms for solving the cooperative localization problem of a group of mobile robots. The scope of this thesis requires that each robot is equipped with two types of sensors. The first type can report the directions in which the other robots are located (but cannot distinguish which robot is at a specified direction). The second type of sensors is a normal compass which is needed for finding a reference direction. The first algorithm proposed in the thesis relies on geometry of the robot's group to derive a deterministic method for calculating the positions of all robots. In the second algorithm, the localization problem is transformed into convex optimization which is solved by using a gradient descent method. The first algorithm is able to deal only with small measurement noise and becomes less robust when the number of robots increases. The second algorithm performs significantly better under noisy measurement. Unlike the first algorithm, it interestingly generates more accurate results as the number of robots increases.



สถาบันวิทยบริการ  
จุฬาลงกรณ์มหาวิทยาลัย

Department ..... Computer Engineering ..... Student's signature .....  
Field of study ..... Computer Engineering ..... Advisor's signature .....  
Academic year ..... 2007 .....

## กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงไปได้ด้วยดี เนื่องจากการสนับสนุนเป็นอย่างดีของอาจารย์ที่ปรึกษา อ.ดร.อรรถวิทย์ สุดแสง อาจารย์เป็นที่ปรึกษาทั้งในส่วนของแนวทางในการทำงานวิจัย ชี้แนะ รวมถึงข้อแนะนำต่างๆของงานวิจัยและเรื่องอื่นนอกเหนือจากงานวิจัยข้าพเจ้าจึงขอขอบคุณอาจารย์เป็นอย่างสูง มา ณ ที่นี้

ถ้าไม่ได้เหล่า พี่ๆ เพื่อนๆ และ น้องๆ แห่งห้องปฏิบัติการวิจัยระบบอัจฉริยะ ISL2 ก็คงไม่สามารถทำงานวิจัยให้ผ่านลุล่วงมาได้เป็นอย่างดีรวมถึงประสบการณ์ต่างๆที่ได้รับแบ่งปันจากทุกๆคนในแลบวิจัยนี้ โดยเฉพาะอย่างยิ่ง พี่แต่ที่เป็นที่ปรึกษาได้ในหลายๆด้าน พี่พิมพ์ที่คอยแนะนำให้แนวทางดีๆในการแก้ปัญหาของงานวิจัย และ ป๊อดซึ่งเป็นผู้ริเริ่มงานวิจัยชิ้นนี้

นอกเหนือจากที่กล่าว มา ข้างต้น แล้ว ยังมีเหล่าคณาจารย์ที่คอย สอบถาม สารทุกข์สุขดิบ เรื่อย มา สุดท้ายนี้ ขอกราบขอบพระคุณ บิดา มารดา พี่ ที่ให้กำลังใจและสนับสนุนเรื่อยมา และขอขอบคุณทุกท่านที่คอยเป็นห่วงด้วยความจริงใจ



สถาบันวิทยบริการ  
จุฬาลงกรณ์มหาวิทยาลัย



# สารบัญ

	หน้า
บทคัดย่อภาษาไทย . . . . .	ง
บทคัดย่อภาษาอังกฤษ . . . . .	จ
กิตติกรรมประกาศ . . . . .	ฉ
สารบัญ . . . . .	ช
สารบัญรูปภาพ . . . . .	ฌ
บทที่	
1 บทนำ . . . . .	1
1.1 ปัญหา . . . . .	1
1.2 การนำเสนอและลำดับเนื้อหาในวิทยานิพนธ์ . . . . .	3
2 ทฤษฎีที่เกี่ยวข้อง . . . . .	4
2.1 Triangulation . . . . .	4
2.2 Trilateration . . . . .	4
2.3 Multilateration . . . . .	6
2.4 การหาค่าเหมาะที่สุดเชิงคอนเวกซ์ . . . . .	8
2.4.1 ปัญหาการหาค่าเหมาะที่สุดเชิงคณิตศาสตร์ . . . . .	9
2.4.2 เซตคอนเวกซ์ . . . . .	9
2.4.3 ฟังก์ชันคอนเวกซ์ : คุณสมบัติพื้นฐาน . . . . .	13
2.4.4 ปัญหาการหาค่าเหมาะที่สุดเชิงคอนเวกซ์ . . . . .	14
3 งานวิจัยที่เกี่ยวข้อง . . . . .	17
3.1 งานด้านกลุ่มของหุ่นยนต์ . . . . .	17
3.2 งานด้านเครือข่ายการรับรู้ . . . . .	22
3.3 งานด้านอื่นๆ . . . . .	25
4 ปัญหาการร่วมกันระบุตำแหน่งและขั้นตอนวิธีการแก้ปัญหา . . . . .	26
4.1 ภาพรวมของปัญหา . . . . .	26
4.2 ขั้นตอนการแก้ปัญหา . . . . .	28
4.2.1 แนวทางการแก้ปัญหาเชิงเรขาคณิต . . . . .	29
4.2.2 แนวทางการแก้ปัญหด้วยวิธีการหาค่าเหมาะที่สุดเชิงคอนเวกซ์ . . . . .	36
5 การหาค่าตำแหน่งที่แท้จริงจากรูปแบบตำแหน่งของกลุ่มหุ่นยนต์ . . . . .	44
5.1 นิยามของปัญหา . . . . .	44
5.2 ขั้นตอนการหาเวกเตอร์การเคลื่อนที่และอัตราขยาย . . . . .	45
6 การทดลองและผลการทำงาน . . . . .	48

บทที่	หน้า
6.1 สภาพแวดล้อมจำลอง . . . . .	48
6.1.1 การทดลองสำหรับขั้นตอนวิธีการช่วยกันระบุตำแหน่งแบบที่ 1 . . . . .	48
6.1.2 การทดลองสำหรับขั้นตอนวิธีการช่วยกันระบุตำแหน่งแบบที่ 2 . . . . .	51
6.2 สภาพแวดล้อมจริง . . . . .	53
<b>7 สรุป . . . . .</b>	<b>56</b>
7.1 สรุปผลการทดลอง . . . . .	56
7.2 สรุปงานวิจัย . . . . .	56
7.3 แนวทางการวิจัยในขั้นต่อไป . . . . .	57
รายการอ้างอิง . . . . .	58
ประวัติผู้เขียนวิทยานิพนธ์ . . . . .	62



สถาบันวิทยบริการ  
จุฬาลงกรณ์มหาวิทยาลัย



## สารบัญรูปภาพ

รูปที่	หน้า
1.1 ตัวอย่างภาพที่ได้จากกล้องมองเห็นรอบด้าน . . . . .	2
1.2 แสดงตำแหน่งของหุ่นยนต์ . . . . .	3
1.3 แสดงข้อมูลที่หุ่นยนต์แต่ละตัวรับรู้ . . . . .	3
2.1 แสดงตำแหน่งของหุ่นยนต์และเส้นการรับรู้ . . . . .	4
2.2 แสดงการจัดเรียงวงกลมเพื่อแก้ปัญหา trilateration . . . . .	5
2.3 แสดงเส้นที่ลากผ่านจุด $x_1$ และ $x_2$ โดยอธิบายอยู่ในเชิงพารามิเตอร์โดย $\theta x_1 + (1 - \theta)x_2$ โดยที่ค่า $\theta$ นั้นจะเปลี่ยนได้ทั่วๆ ไป ส่วนของเส้นตรงระหว่างจุด $x_1$ และ $x_2$ จะสอดคล้องกับค่า $\theta$ ที่อยู่ในช่วง 0 ถึง 1 (สี่ที่เข้มกว่า) . . . . .	10
2.4 แสดงตัวอย่างอย่างง่ายของเซตคอนเวกซ์และเซตเว้า ก. รูปหกเหลี่ยมที่รวมถึงขอบ(เส้นทึบ)เป็นคอนเวกซ์ ข. รูปไตไม่เป็นคอนเวกซ์เนื่องจากมีส่วนของเส้นตรงที่ลากเชื่อมจุดสองจุดในเซตและไม่ได้อยู่ในเซต ค. รูปสี่เหลี่ยมที่ประกอบด้วยขอบเพียงบางส่วนไม่เป็นคอนเวกซ์ . . . . .	12
2.5 แสดงเปลือกหุ้มเชิงคอนเวกซ์ของเซตสองเซตที่อยู่ใน $\mathbb{R}^2$ ก. เปลือกหุ้มเชิงคอนเวกซ์ของจุดจำนวน 15 จุดคือรูปหกเหลี่ยม(ที่แรเงา) ข. เปลือกหุ้มเชิงคอนเวกซ์ของรูปไตที่แสดงในรูป 2.4ข. ก็คือส่วนที่แรเงาทั้งหมด . . . . .	12
2.6 แสดงกราฟของฟังก์ชันคอนเวกซ์ เส้นพาด(ส่วนของเส้นตรง)ระหว่างจุดสองจุดใดๆของกราฟจะอยู่เหนือกราฟ . . . . .	13
4.1 รูปก. แสดงภาพที่ได้รับจากกล้องพร้อมทั้งแกนอ้างอิงของภาพและ รูปข.แสดงข้อมูลเชิงมุมที่หาได้จากภาพ(แทนด้วยลูกศรชี้ออกจากจุดศูนย์กลางของวงกลม) . . . . .	26
4.2 แสดงข้อมูลที่หุ่นยนต์แต่ละตัวรับรู้ . . . . .	27
4.3 แสดงแผนภาพขั้นตอนของระบบ กล้องสี่เข็มแสดงถึงส่วนที่งานวิจัยนี้จะนำเสนอ . . . . .	27
4.4 แสดงตำแหน่งของหุ่นยนต์และเส้นการรับรู้ . . . . .	28
4.5 รูป ก. แสดงให้เห็นถึงกรณีที่กลุ่มหุ่นยนต์เคลื่อนที่ไปในทิศทางเดียวกันด้วยระยะทางที่เท่ากัน ข้อมูลที่วัดได้ก็จะไม่เปลี่ยนแปลง รูป ข. แสดงในกรณีที่หุ่นยนต์ทุกตัววิ่งห่างออกจากกันด้วยอัตราส่วนเท่าๆกัน ข้อมูลที่วัดได้ก็ไม่เปลี่ยนแปลงเช่นเดียวกัน . . . . .	28
4.6 แสดงพิกัดของระบบ . . . . .	29
4.7 แสดงขั้นตอนวิธีการ triangulation โดยที่รู้ตำแหน่งของจุดสองจุด(จุดด้านล่าง) และรู้ตำแหน่งเชิงมุมของจุดที่สามเทียบกับจุดทั้งสองจุดนั้น . . . . .	29
4.8 แสดงขั้นตอนวิธีการ triangulation โดยรู้มุมที่สัมพันธ์กันระหว่างจุดสองจุด(จุดด้านล่าง 2 จุด) และรู้ตำแหน่งเชิงมุมของจุดที่สามเทียบกับจุดทั้งสองจุดนั้น . . . . .	30
4.9 รูปก. จุดสีดำแสดงจุดตัดที่เกิดขึ้นจากการตัดกันของข้อมูลของหุ่นยนต์สองตัว รูปข. จุดสีเทาแสดงตำแหน่งของหุ่นยนต์จริงๆ ส่วนจุดขาวแสดงถึงจุดผี รูปค. แสดงให้เห็นวิธีการกำจัดจุดผีเมื่อรู้ตำแหน่งของจุดที่สามเพิ่มขึ้นมา เส้นประแสดงถึงข้อมูลของจุดที่สามที่นำมาใช้กำจัดจุดผี . . . . .	30
4.10 แสดงกรณีที่หุ่นยนต์สามตัวที่ใช้กำจัดจุดผี(จุดสีเข้ม)อยู่ในแนวเส้นตรงเดียวกัน แล้วจะทำให้ไม่สามารถหาตำแหน่งของจุดที่ 5 ได้ . . . . .	30

รูปที่	หน้า
4.11 แสดงการจับคู่ระหว่างข้อมูลของหุ่นยนต์ . . . . .	31
4.12 แสดงจุดและเส้นที่เป็นเปลือกหุ้มเชิงคอนเวกซ์ของกลุ่มหุ่นยนต์ ลูกศรที่ชี้ออกนั้นแสดงถึงทิศทางตั้งฉากของเส้นขอบนั้นๆ . . . . .	31
4.13 แสดงส่วนของเส้นตรง $A$ และ $B$ ซึ่งมีทิศทางตั้งฉากเดียวกันแต่ไม่อยู่ในแนวเส้นตรงเดียวกัน . . . . .	32
4.14 แสดงให้เห็นถึงมุมซ้ายสุดและมุมขวาสุดของหุ่นยนต์ $A, B, D$ และ $E$ หุ่นยนต์ $C$ นั้นไม่ได้เป็นส่วนหนึ่งของกรอบเนื่องจากไม่ตรงตามเงื่อนไข ส่วนที่แรเงาแสดงถึงบริเวณที่ข้อมูลเชิงมุมที่เหลือน้อย . . . . .	33
4.15 แสดงให้เห็นขั้นตอนที่ 1 และ 2 . . . . .	33
4.16 ในรูปก. จะแสดงให้เห็นหลังจากหาจุดที่สามได้แล้ว แต่ยังไม่รู้ระยะห่างระหว่างจุดที่สามและจุดแรกหรือก็คือยังไม่รู้ความสัมพันธ์ระหว่างจุดที่สองและจุดที่สาม ในรูปข. จะแสดงให้เห็นว่าจุดสามจุดนี้จะวางตัวเป็นรูปสามเหลี่ยมโดยมีจำนวนจุดที่อยู่ด้านในเท่ากับ $S$ ซึ่งถ้าเราเรียงข้อมูลมุมของจุดที่สองและสาม โดยเริ่มจากจุดแรกตามทิศทางวนเข็มและตามเข็มนาฬิกาตามลำดับแล้ว ข้อมูลที่จะเป็นความสัมพันธ์ระหว่างจุดที่สองและจุดที่สามคือข้อมูลลำดับที่ $S + 1$ . . . . .	34
4.17 แสดงจุดตัดทั้งหมดที่เกิดขึ้นจากจุดแรกและจุดที่สอง . . . . .	34
4.18 แสดงจุดตัดที่นำมาเรียงมุมเทียบกับจุดสามตามเข็มนาฬิกาโดยเริ่มตั้งแต่มุมซ้ายสุดของจุดสาม . . . . .	35
4.19 แสดงชุดข้อมูลสองชุดที่จะนำมาเทียบกัน โดยมุมที่แสดงถึงความสัมพันธ์ระหว่างจุดสามและจุดแรกจะไม่นำมาคิด . . . . .	35
4.20 แสดงกรณีที่สามารถทำให้การหาตำแหน่งด้วยวิธีนี้ผิดพลาดได้ จุดสีขาวยังแสดงตำแหน่งที่เกิดจากการตัดกันของเส้นสามเส้น จุดสีเทาแสดงตำแหน่งจริงๆของหุ่นยนต์ . . . . .	36
4.21 ในรูปก. แสดงตำแหน่งของหุ่นยนต์สามตัว และ ข-ง แสดงข้อมูลที่หุ่นยนต์แต่ละตัวรับได้ . . . . .	37
4.22 ในรูปก. แสดงตำแหน่งของหุ่นยนต์สามตัวเมื่อทำการเปลี่ยนตำแหน่งแล้ว และ ข-ง แสดงข้อมูลที่หุ่นยนต์แต่ละตัวรับได้ . . . . .	37
4.23 แสดงเวกเตอร์รวมข้อมูลเชิงมุมของหุ่นยนต์แต่ละตัวในกลุ่มหุ่นยนต์ที่เราสนใจ โดยเส้นที่เป็นลูกศรจะแสดงเวกเตอร์รวมข้อมูลเชิงมุม . . . . .	37
4.24 แสดงเวกเตอร์รวมข้อมูลเชิงมุมของหุ่นยนต์แต่ละตัวใน “กลุ่มหุ่นยนต์ทำซ้ำ” . . . . .	38
4.25 เส้นประแต่ละเส้นแสดงเวกเตอร์การเคลื่อนที่สำหรับหุ่นยนต์แต่ละตัว . . . . .	38
5.1 แสดงรูปแบบ ตำแหน่ง ของ กลุ่ม หุ่นยนต์(A)และ ตำแหน่ง จริงๆของ กลุ่ม หุ่นยนต์(B) ซึ่งสามารถมองให้อยู่ในลักษณะของรูปภาพได้ และการหาตำแหน่งที่แท้จริงของหุ่นยนต์ก็คือการเคลื่อนย้ายและขยายขนาดรูป $A$ ให้กลายเป็นรูป $B$ . . . . .	44
6.1 กราฟแสดงค่าความผิดพลาดของขั้นตอนวิธี แกน $x$ แสดงจำนวนหุ่นยนต์ แกน $y$ แสดงค่าความน่าจะเป็นที่จะผิดพลาดรูป ก. พื้นที่ทดลองมีขนาด $640 \times 480$ รูป ข. พื้นที่ทดลองมีขนาด $320 \times 240$ รูป ค. แสดงถึงผลต่างระหว่างรูป ก. และ รูป ข. . . . .	49
6.2 กราฟแสดงค่าความผิดพลาดของขั้นตอนวิธี แกน $x$ แสดงจำนวนหุ่นยนต์ แกน $y$ แสดงค่าความน่าจะเป็นที่จะผิดพลาดในรูปก. พื้นที่ทดลองมีขนาด $1000 \times 10$ ในรูปข. พื้นที่ทดลองมีขนาด $116 \times 87$ และรูปค. แสดงถึงผลต่างระหว่างรูป ก. และ รูป ข. . . . .	50

รูปที่	หน้า
6.3 แสดงจำนวนขั้นตอนที่ใช้ในการแก้ปัญหา แกน $x$ แสดงจำนวนหุ่นยนต์ แกน $y$ แสดงจำนวนขั้นตอน กราฟ ก. สำหรับวิธีการแก้ปัญหาแบบไต่ทางลาดกราฟ ข. สำหรับวิธีการแก้ปัญหาโดยวิธี interior point . . . . .	51
6.4 แสดงค่าเฉลี่ยความคลาดเคลื่อนของมุม แกน $x$ แสดงจำนวนหุ่นยนต์ แกน $y$ แสดงค่าความผิดพลาดหน่วยเป็นเรเดียน กราฟ ก. สำหรับวิธีการแก้ปัญหาแบบไต่ทางลาดกราฟ ข. สำหรับวิธีการแก้ปัญหาโดยวิธี interior point . . . . .	52
6.5 แสดงค่าเฉลี่ยความคลาดเคลื่อนของมุม แกน $x$ แสดงจำนวนหุ่นยนต์ แกน $y$ แสดงค่าความผิดพลาดหน่วยเป็นเรเดียน กราฟ ก. สำหรับวิธีการแก้ปัญหาแบบไต่ทางลาดกราฟ ข. สำหรับวิธีการแก้ปัญหาโดยวิธี interior point . . . . .	53
6.6 แสดงตำแหน่งของหุ่นยนต์แต่ละตัว . . . . .	54
6.7 แสดงตำแหน่งของหุ่นยนต์ที่เป็นคำตอบ รูป ก. สำหรับวิธีการแก้ปัญหาแบบไต่ทางลาดรูป ข. สำหรับวิธีการแก้ปัญหาโดยวิธี interior point . . . . .	55
7.1 แสดงรูปแบบการบังกันของหุ่นยนต์ จะเห็นว่าหุ่นยนต์หมายเลข 1 จะสามารถมองเห็นหุ่นยนต์หมายเลข 2 และ 3 ได้เท่านั้น . . . . .	57

# บทที่ 1

## บทนำ

### 1.1 ปัญหา

โดยทั่วไปการที่หุ่นยนต์เคลื่อนที่ได้(mobile robot)จะสามารถทำงานได้อย่างอัตโนมัติและถูกต้องนั้น หุ่นยนต์จำเป็นที่จะต้องรู้ตำแหน่งของตนเองว่าอยู่ที่ใดในสภาพแวดล้อม ความสามารถในการระบุตำแหน่งของตนเองนั้นทางศาสตร์ด้านหุ่นยนต์เรียกว่า Localization การระบุตำแหน่งเป็นสิ่งที่จำเป็นในงานหลายๆงานในศาสตร์ทางด้านหุ่นยนต์ ยกตัวอย่างเช่น หุ่นยนต์นำเที่ยว[1]ทำหน้าที่นำกลุ่มนักท่องเที่ยวไปยังบริเวณที่จัดแสดงสิ่งของต่างๆและบรรยายถึงสิ่งของที่จัดแสดงไว้ในบริเวณนั้นของงานได้อย่างถูกต้อง โดยในขณะที่กำลังนำเที่ยวอยู่นั้นหุ่นยนต์จำเป็นที่จะต้องหลบหลีกสิ่งกีดขวางต่างๆ อาทิเช่นนักท่องเที่ยว และสิ่งของที่จัดแสดงไว้ในพิพิธภัณฑ์ หุ่นยนต์พยาบาล[2]นำอาหารและยาส่งถึงห้องของคนไข้ต่างๆได้อย่างอัตโนมัติและปลอดภัย หุ่นยนต์ดูดฝุ่น[3]ก็สามารถที่จะจัดการกวาดพื้นบ้านทั้งหมดและยังสามารถที่จะเคลื่อนที่กลับแท่นชาร์จเพื่อชาร์จแบตเตอรี่ของตัวเองได้อย่างอัตโนมัติ ในขณะที่การระบุตำแหน่งสำหรับหุ่นยนต์หนึ่งตัวนั้นมีการพัฒนามาเป็นอย่างดีและให้ผลลัพธ์เป็นที่น่าพึงพอใจพอสมควร ความท้าทายจากปัญหาใหม่ๆก็เกิดขึ้นจากการความต้องการที่จะระบุตำแหน่งของหุ่นยนต์หลายตัวพร้อมกัน

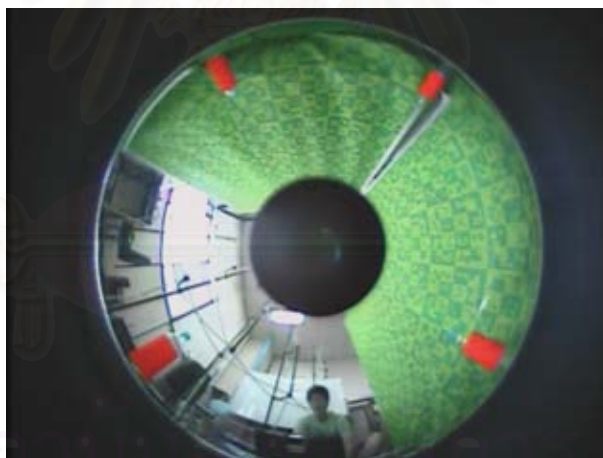
การทำงานของระบบหุ่นยนต์หลายตัวมีข้อดีกว่าหุ่นยนต์ตัวเดียวในหลายๆด้าน โดยเฉพาะงานบางประเภทก็จำเป็นที่จะต้องใช้หุ่นยนต์หลายๆตัว เช่น การร่วมมือกันเพื่อเคลื่อนย้ายสิ่งของซึ่งมีขนาดหรือน้ำหนักเกินขีดความสามารถของหุ่นยนต์ตัวเดียว ระบบหุ่นยนต์หลายตัวยังมีประสิทธิภาพมากกว่าหุ่นยนต์ตัวเดียวอย่างเห็นได้ชัดโดยเฉพาะงานบางอย่างที่หุ่นยนต์ทุกตัวสามารถที่จะทำงานได้พร้อมๆกันอย่างเป็นอิสระจากกัน ยกตัวอย่างเช่น การกู้ภัย การสร้างแผนที่ และการกู้ระเบิด เป็นต้น การพัฒนาทางด้านอุปกรณ์และเครื่องมือต่างๆในปัจจุบันทำให้สามารถที่จะผลิตหุ่นยนต์จำนวนมากได้ในต้นทุนที่ถูกลงและง่ายขึ้น ดังนั้นงานวิจัยทางด้านระบบหุ่นยนต์หลายตัวจึงได้รับความสนใจเป็นอย่างมากและมีความเป็นไปได้ที่จะได้รับการยอมรับกันอย่างกว้างขวางในอนาคต ปัญหาที่ท้าทายมากที่สุดปัญหาหนึ่งในระบบหุ่นยนต์หลายตัวนั้นก็คือการระบุตำแหน่งของหุ่นยนต์หลายๆตัวอย่างถูกต้องและมีประสิทธิภาพ ถึงแม้ว่าในความเป็นจริงจะสามารถที่จะนำวิธีการระบุตำแหน่งที่ใช้สำหรับหุ่นยนต์หนึ่งตัวมาประยุกต์ใช้ได้ก็ตาม แต่ค่าใช้จ่ายทั้งหมดสำหรับการที่จะต้องใช้ติดตั้งอุปกรณ์การระบุตำแหน่งให้กับหุ่นยนต์ทุกๆตัวนั้นสูงมาก วิธีการแก้ปัญหาแบบหนึ่งก็คือแบ่งกลุ่มหุ่นยนต์ทั้งหมดออกเป็นสองส่วนโดยในส่วนแรกนั้นจะเป็นหุ่นยนต์พิเศษที่สามารถระบุพิกัดตนเองได้อย่างถูกต้องเทียบกับสภาพแวดล้อมและหุ่นยนต์ส่วนที่เหลือนั้นสามารถที่จะทำได้เพียงระบุตำแหน่งของตนเองเทียบกับหุ่นยนต์ตัวอื่นๆ วิธีการนี้เรียกว่า การร่วมกันระบุตำแหน่ง (Cooperative Localization)

อย่างไรก็ตามความท้าทายต่างๆก็ยังคงมีอยู่ในปัญหาการร่วมกันระบุตำแหน่ง โดยเฉพาะอย่างยิ่งการที่จะระบุตำแหน่งอ้างอิงกับหุ่นยนต์พิเศษอย่างมีประสิทธิภาพและใช้การประมวลผลอย่างคุ้มค่า ปัญหาโดยทั่วไปเกิดจากสัญญาณรบกวนและข้อจำกัดทางด้านความละเอียดของตัวรับรู้ ระบบการร่วมกันระบุตำแหน่ง ส่วนมากจะใช้สัญญาณวิทยุในการติดต่อสื่อสารและยังใช้เป็นเครื่องมือในการระบุตำแหน่งด้วย เนื่องจากลักษณะการเดินทางของสัญญาณวิทยุนี้ไม่สามารถที่จะคาดเดาได้ง่ายๆเพราะลักษณะ



เฉพาะตัวบางอย่างเช่น การรบกวนที่เกิดจากการสะท้อน การลดทอนของสัญญาณ และการรบกวนจากสภาพแวดล้อม ผลลัพธ์ที่ได้จากการคำนวณระยะทางโดยอาศัยความแรงของสัญญาณนั้นย่อมมีค่าความผิดพลาดที่ค่อนข้างมาก การร่วมกันระบุตำแหน่งบางวิธีใช้กล้องเพื่อทำการระบุตำแหน่ง ส่วนมากทำได้โดยการหาจุดสนใจเพื่อนำมาหาตำแหน่งอ้างอิงของหุ่นยนต์กับสภาพแวดล้อม โดยหุ่นยนต์แต่ละตัวจำเป็นที่จะต้องมีความสนใจที่แตกต่างกันออกไปเพื่อที่จะทำให้สามารถค้นหาและแยกแยะหุ่นยนต์แต่ละตัวได้ ยกตัวอย่างเช่น สีที่แตกต่างกันออกไปในแต่ละตัว ดังนั้นเมื่อกลุ่มหุ่นยนต์ที่มีจำนวนสมาชิกมากก็จำเป็นที่จะต้องใช้จำนวนจุดสนใจที่แตกต่างกันออกไปมากขึ้นด้วย ยิ่งไปกว่านั้นการที่จะหาระยะทางระหว่างกันได้นั้นจำเป็นที่จะต้องรู้ขนาดของจุดสนใจนั้นด้วย ความแม่นยำในการระบุตำแหน่งและความสามารถในการที่จะแยกแยะในกลุ่มหุ่นยนต์จำนวนมากนั้นขึ้นอยู่กับคุณภาพของรูปภาพซึ่งขึ้นอยู่กับปัจจัยหลายๆด้าน เช่น สภาพแวดล้อม และการออกแบบและประกอบของกล้อง ในงาน [4] หุ่นยนต์ของพวกเขาถูกติดตั้งด้วยแถบสีต่างๆ สีของรูปภาพจะถูกรบกวนโดยแหล่งกำเนิดแสงหลายๆแหล่งและการแยกแยะสีจำนวนมากนั้นเป็นเรื่องที่ยาก

ในวิทยานิพนธ์ฉบับนี้ เราจะนำเสนอขั้นตอนวิธีแบบใหม่และการทดลองสำหรับวิธีนี้ วิธีที่นำเสนอนี้ใช้ในการระบุตำแหน่งอ้างอิงของหุ่นยนต์กับหุ่นยนต์ตัวอื่น ระบบหุ่นยนต์หลายตัวที่เราสนใจนั้นใช้หุ่นยนต์ที่ได้ทำการติดตั้งเข็มทิศและกล้องออปติคัล ซึ่งกล้องชนิดนี้เป็นตัวรับรู้ที่มีมุมการรับรู้กว้างถึง 360 องศา ซึ่งหมายความว่าหุ่นยนต์แต่ละตัวจะสามารถรับรู้สภาพแวดล้อม รอบๆตัวได้ทั้งหมดจากภาพเพียงภาพเดียว รูปที่ [1.1] แสดงตัวอย่างภาพที่ได้จากกล้องชนิดนี้ ภาพที่ได้จากกล้องจะถูกนำมาประมวลผลหา

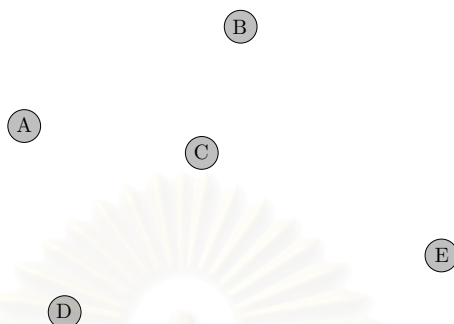


รูปที่ 1.1: ตัวอย่างภาพที่ได้จากกล้องมองเห็นรอบด้าน

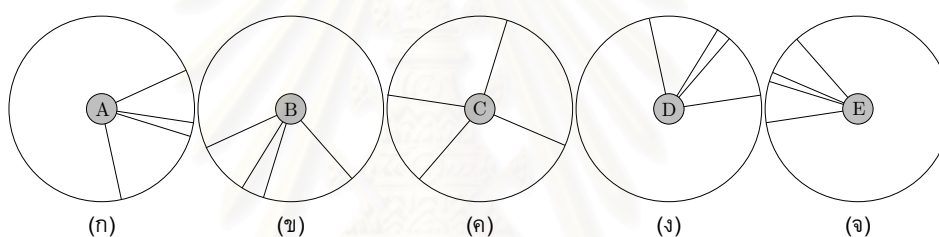
ตำแหน่งของจุดสนใจในภาพ (สำหรับงานวิจัยนี้ก็คือตำแหน่งของหุ่นยนต์ตัวอื่นๆ) ตำแหน่งของจุดสนใจในภาพจะสามารถแปลงให้เป็นระยะทางและทิศทางระหว่างจุดสนใจกับกล้องได้โดยการใช้สมการรูปแบบการหักเหแสงของกล้อง แต่จากข้อจำกัดหลายๆประการอาทิเช่น สภาพแสง ความละเอียดของภาพความคลาดเคลื่อนที่เกิดจากการผลิตกล้องและเลนส์ ฯลฯ ทำให้เราไม่สามารถนำข้อมูลระยะทางมาใช้งานได้ในทางปฏิบัติ และนอกจากนั้น ข้อจำกัดเหล่านี้ยังส่งผลต่อความทนทานของขั้นตอนวิธีที่ใช้ในการแยกแยะจุดสนใจหลายๆแบบอีกด้วย ในงานวิจัยนี้จึงใช้จุดสนใจแบบเดียวสำหรับหุ่นยนต์ทุกตัว

จากที่กล่าวมาข้างต้นส่งผลให้ขั้นตอนวิธีที่เสนอจำเป็นต้องใช้ข้อมูลเชิงมุมเพียงอย่างเดียว และต้องทำงานได้กับข้อมูลที่ไม่มีความสมนัยระหว่างจุดสนใจกับหุ่นยนต์ หุ่นยนต์แต่ละตัวจะคำนวณหา

ตำแหน่งของหุ่นตัวอื่นๆในภาพและเปลี่ยนตำแหน่งนั้นให้เป็นค่ามุมเทียบกับแกนของกล้อง หลังจากนั้นนำข้อมูลจากเข็มทิศมาปรับเพื่อให้มีแกนอ้างอิงเดียวกันทั้งหมดยกตัวอย่างระบบที่มีหุ่นยนต์ทั้งหมด 5 ตัว และหุ่นยนต์แต่ละตัวอยู่ในตำแหน่งต่างๆดังรูป [1.2] ข้อมูลเชิงมุมที่หุ่นยนต์แต่ละตัวคำนวณได้แสดงดังรูป [1.3] เส้นที่อยู่ในแนวรัศมีจะแสดงถึงตำแหน่งเชิงมุมของหุ่นยนต์ตัวอื่นๆ ขั้นตอนวิธีที่นำเสนอจะนำ



รูปที่ 1.2: แสดงตำแหน่งของหุ่นยนต์



รูปที่ 1.3: แสดงข้อมูลที่หุ่นยนต์แต่ละตัวรับรู้

ข้อมูลเหล่านี้ไปคำนวณเพื่อหารูปแบบการวางตัวของกลุ่มหุ่นยนต์

## 1.2 การนำเสนอและลำดับเนื้อหาในวิทยานิพนธ์

เนื้อหาของวิทยานิพนธ์ฉบับนี้แบ่งออกได้เป็น 3 ส่วนหลักๆ

- ส่วนแรกเป็นส่วนที่อธิบายถึงความรู้และหลักการพื้นฐานที่ใช้ในงานการร่วมกันระบุตำแหน่ง (บทที่ 2) และจะกล่าวถึงงานวิจัยที่เกี่ยวข้องกับการร่วมกันระบุตำแหน่ง (บทที่ 3) ซึ่งงานวิจัยที่เกี่ยวข้องกับปัญหานี้สามารถแบ่งได้เป็น 2 สาขาใหญ่ๆคือ ศาสตร์ทางด้านระบบหุ่นยนต์หลายตัว และศาสตร์ทางด้านเครือข่ายการรับรู้ไร้สาย
- ส่วนที่สองจะอธิบายถึงปัญหาและนำเสนอขั้นตอนวิธีที่ใช้ในการแก้ปัญหา (บทที่ 4) และเสนอขั้นตอนการทดลองรวมถึงผลการทดลอง (บทที่ 6) ในบทที่ 5 เราจะนำเสนอวิธีที่จะนำคำตอบของปัญหาในบทที่ 4 มาเปลี่ยนเป็นตำแหน่งที่แท้จริงของหุ่นยนต์แต่ละตัว
- ในส่วนสุดท้ายบทที่ 7 จะสรุปงานวิจัยชิ้นนี้และสรุปผลการทดลอง รวมถึงกล่าวถึงแนวทางในการวิจัยในขั้นถัดไป

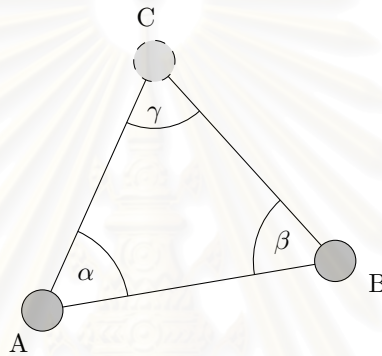
## บทที่ 2

### ทฤษฎีที่เกี่ยวข้อง

ในบทนี้จะอธิบายทฤษฎีที่เกี่ยวข้องกับการร่วมกันระบุตำแหน่งโดยสังเขป

#### 2.1 Triangulation

ในทางเรขาคณิต เมื่อทราบตำแหน่งของจุดสองจุดในสามเหลี่ยมและมุมภายในของสามเหลี่ยมที่จุดทั้งสองนั้นเราสามารถคำนวณพิกัดของจุดที่สามได้ กระบวนการนี้เรียกว่า triangulation ยกตัวอย่างเช่น จากรูป[2.1] ถ้าเรารู้ตำแหน่งของ  $A$  และ  $B$  และรู้มุม  $\alpha$  และ  $\beta$  เราจะสามารถหาความยาวของด้านทั้ง



รูปที่ 2.1: แสดงตำแหน่งของหุ่นยนต์และเส้นการรับรู้

สามของสามเหลี่ยมได้โดยใช้กฎของ sine

$$\frac{\sin(\alpha)}{BC} = \frac{\sin(\beta)}{AC} = \frac{\sin(\gamma)}{AB} \quad (2.1)$$

โดยมุม  $\gamma$  สามารถหาได้จาก  $180 - \alpha - \beta$  ดังนั้นเราจึงสามารถหาตำแหน่งของ  $C$  จาก

$$C = A + Rot(\alpha) \cdot \overline{AB} \frac{\sin(\beta)}{\sin(\gamma)} \quad (2.2)$$

โดย  $Rot(\alpha)$  คือ rotation matrix ที่แทนการหมุนทวนเข็มนาฬิกาเป็นมุม  $\alpha$

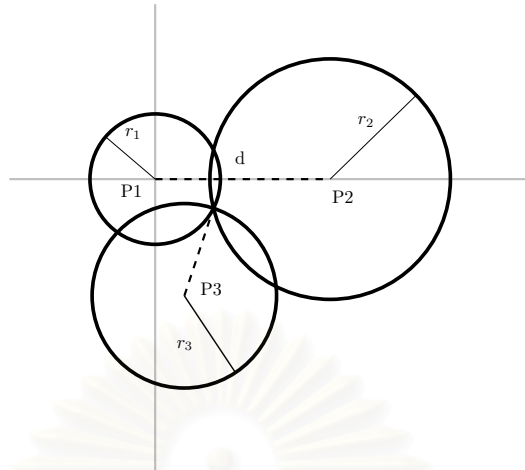
#### 2.2 Trilateration

Trilateration เป็นวิธีที่ใช้สำหรับหาตำแหน่งโดยอาศัยเรขาคณิตของสามเหลี่ยม ซึ่งคล้ายกับวิธี triangulation แต่ต่างกันที่ trilateration จะใช้ตำแหน่งของจุดและระยะทางระหว่างจุดที่สนใจกับจุดอ้างอิง ในกรณีที่เป็นสภาพแวดล้อม 2 มิติจำเป็นที่จะต้องใช้อย่างน้อย 3 จุดเพื่อคำนวณพิกัดของจุดที่สนใจได้

สมการทางคณิตศาสตร์ที่ใช้สำหรับแก้ปัญหา trilateration นั้นสามารถเขียนขึ้นมาจากสมการของวงกลมสามวง(ในกรณีที่เป็นระนาบสองมิติ) ก่อนอื่นทำการเลื่อนตำแหน่งของวงกลมทั้งสามวงด้วยจุดศูนย์กลางของวงกลมวงใดวงหนึ่งเพื่อทำให้จุดศูนย์กลางของวงกลมนั้นอยู่ที่จุด  $(0,0)$  หลังจากนั้นทำ



การหมุนวงกลมอีกสองวงที่เหลือเพื่อให้วงกลมวงใดวงหนึ่งอยู่บนแกน  $x$  ดังรูป[2.2] เมื่อเราหาตำแหน่ง



รูปที่ 2.2: แสดงการจัดเรียงวงกลมเพื่อแก้ปัญหา trilateration

ของจุดที่สนใจได้แล้ว จะทำการหมุนและเลื่อนตำแหน่งกลับเพื่อให้ได้ตำแหน่งจริงๆของจุดนั้น เริ่มจากสมการของวงกลมทั้งสาม ต้องการที่จะหาจุด  $x, y$  ที่ทำให้สมการสามสมการนี้ถูกต้อง

$$r_1^2 = x^2 + y^2 \quad (2.3)$$

$$r_2^2 = (x - d)^2 + y^2 \quad (2.4)$$

$$r_3^2 = (x - i)^2 + (y - j)^2 \quad (2.5)$$

จากนั้นทำการลบสมการ (2.3) ด้วย (2.4) จะได้

$$x = \frac{r_1^2 - r_2^2 + d^2}{2d} \quad (2.6)$$

นำค่า  $x$  ที่ได้ไปแทนในสมการ (2.3) จะได้ค่า  $y$

$$y^2 = r_1^2 - \frac{(r_1^2 - r_2^2 - d^2)^2}{4d^2} \quad (2.7)$$

หลังจากนั้นนำค่าทั้งหมดที่ได้ไปแทนในสมการ(2.5) จะได้

$$y = \frac{r_1^2 - r_3^2 - x^2 + (x - i)^2 + j^2}{2j} = \frac{r_1^2 - r_3^2 + i^2 + j^2}{2j} - \frac{i}{j}x \quad (2.8)$$

หลังจากได้คำตอบของทั้งสาม สมการแล้ว จะสามารถตรวจสอบได้ว่าตำแหน่งที่หามาได้นั้นถูกต้องหรือไม่(ในกรณีที่วงกลมสามวงไม่ตัดกันที่จุดเดียวกัน ก็จะถือว่าไม่มีคำตอบ) โดยการตรวจสอบจาก

$$\sqrt{r_1^2 - x^2 - y^2} \quad (2.9)$$

ถ้าค่าภายใน square root ด้านบนมีค่าไม่ใช่ 0 คำตอบนั้นก็จะไม่ถูกต้อง

### 2.3 Multilateration

Multilateration หรือรู้จักในอีกชื่อหนึ่งว่า hyperbolic positioning นั้นเป็นวิธีที่จะใช้หาตำแหน่งโดยอาศัยการคำนวณข้อมูลความแตกต่างของเวลาที่มาถึง (time difference of arrival) ของสัญญาณซึ่งส่งจากตัวส่งสัญญาณมายังตัวรับจำนวนสามตัวขึ้นไป ข้อแตกต่างระหว่าง multilateration กับ trilateration คือลักษณะของข้อมูลที่นำมาใช้ trilateration นั้นจะใช้ระยะทางสัมบูรณ์ ส่วน multilateration นั้นจะใช้ส่วนต่างของเวลา

ถ้าวัตถุส่งสัญญาณออกมาแล้วตัวรับสัญญาณจะสามารถรับสัญญาณนี้ได้ในเวลาที่แตกต่างกัน T-DOA ก็คือความแตกต่างระหว่างเวลาที่มาถึงของสัญญาณในการใช้งานจริงนั้นเราจะทราบถึงตำแหน่งของตัวรับสัญญาณ ในสภาพแวดล้อมสามมิติ เมื่อรู้ทั้งตำแหน่งของตัวรับสัญญาณสองตัวและ TDOA แล้วพิกัดของตัวส่งสัญญาณที่จะเป็นไปได้นั้นจะต้องอยู่บนพื้นผิวของทรงไฮเพอร์โบลาคี่หนึ่ง และเมื่อรู้ตำแหน่งของจุดรับสัญญาณอันที่สามและ TDOA ที่สัมพันธ์กันก็จะสามารถคำนวณพื้นผิวของทรงไฮเพอร์โบลาคี่ได้เพิ่มมาอีกชั้น และเมื่อนำไฮเพอร์โบลาคี่ทั้งสองมาตัดกันจะได้โค้งที่เป็นพิกัดที่เป็นไปได้ทั้งหมดของตัวส่งสัญญาณ เมื่อรวมข้อมูลพิกัดของจุดที่สี่และ TDOA ที่สัมพันธ์กัน ก็จะทำได้พิกัดที่แท้จริงของตัวส่งสัญญาณในสามมิติ ในกรณีที่อยู่ในสภาพแวดล้อม 2 มิติจำนวนตัวรับสัญญาณน้อยที่สุดที่ต้องใช้คือ 3 ตัว กรณีทั่วไปในสภาพแวดล้อม  $N$  มิติจำนวนตัวรับสัญญาณน้อยที่สุดที่ต้องใช้ในการหาตำแหน่งเท่ากับ  $N + 1$  เมื่อจำนวนของจุดมีมากกว่านั้นก็สามารถนำข้อมูลที่เกินนั้นมาช่วยเพิ่มความแม่นยำให้กับคำตอบ

หัวใจสำคัญของการแก้ปัญหา TDOA นั้นคือสมการระยะทางระหว่างจุดสองจุด

$$D = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2} \quad (2.10)$$

กำหนดให้  $R_i$  แทนระยะทางระหว่างตัวส่งสัญญาณกับตัวรับสัญญาณตัวที่  $i$  และ  $c$  แสดงความเร็วของสัญญาณ

$$ct_i = R_i = \sqrt{(x_i - x)^2 + (y_i - y)^2 + (z_i - z)^2} \quad (2.11)$$

$$ct_j = R_j = \sqrt{(x_j - x)^2 + (y_j - y)^2 + (z_j - z)^2} \quad (2.12)$$

$$ct_k = R_k = \sqrt{(x_k - x)^2 + (y_k - y)^2 + (z_k - z)^2} \quad (2.13)$$

$$ct_l = R_l = \sqrt{(x_l - x)^2 + (y_l - y)^2 + (z_l - z)^2} \quad (2.14)$$

โดยที่  $x_i, y_i, z_i, x_j, y_j, z_j, x_k, y_k, z_k$  และ  $x_l, y_l, z_l$  แสดงตำแหน่งของตัวส่งสัญญาณ กำหนดให้  $R_{ij}$  แสดงถึง TDOA ระหว่างจุด  $i$  และจุด  $j$

$$R_{ij} = R_i - R_j = \sqrt{(x_i - x)^2 + (y_i - y)^2 + (z_i - z)^2} - \sqrt{(x_j - x)^2 + (y_j - y)^2 + (z_j - z)^2} \quad (2.15)$$

$$R_{ik} = R_i - R_k = \sqrt{(x_i - x)^2 + (y_i - y)^2 + (z_i - z)^2} - \sqrt{(x_k - x)^2 + (y_k - y)^2 + (z_k - z)^2} \quad (2.16)$$

$$R_{kj} = R_k - R_j = \sqrt{(x_k - x)^2 + (y_k - y)^2 + (z_k - z)^2} - \sqrt{(x_j - x)^2 + (y_j - y)^2 + (z_j - z)^2} \quad (2.17)$$

$$R_{kl} = R_k - R_l = \sqrt{(x_k - x)^2 + (y_k - y)^2 + (z_k - z)^2} - \sqrt{(x_l - x)^2 + (y_l - y)^2 + (z_l - z)^2} \quad (2.18)$$

หลังจากนั้นทำการย้าย sqrt ตัวหนึ่งจากด้านขวามาทางด้านซ้าย แล้วยกกำลังสองทั้งสองฝั่งและทำการแทนที่  $x_j - x_i$  ด้วย  $x_{ji}$  และตัวอื่นๆในทำนองนี้ แล้วจัดรูปใหม่จะได้

$$\sqrt{(x_i - x)^2 + (y_i - y)^2 + (z_i - z)^2} = \frac{R_{ij}^2 + x_i^2 - x_j^2 + y_i^2 - y_j^2 + z_i^2 - z_j^2 + 2x_{ji}x + 2y_{ji}y + 2z_{ji}z}{2R_{ij}} \quad (2.19)$$

$$\sqrt{(x_i - x)^2 + (y_i - y)^2 + (z_i - z)^2} = \frac{R_{ik}^2 + x_i^2 - x_k^2 + y_i^2 - y_k^2 + z_i^2 - z_k^2 + 2x_{ki}x + 2y_{ki}y + 2z_{ki}z}{2R_{ik}} \quad (2.20)$$

$$\sqrt{(x_k - x)^2 + (y_k - y)^2 + (z_k - z)^2} = \frac{R_{kj}^2 + x_k^2 - x_j^2 + y_k^2 - y_j^2 + z_k^2 - z_j^2 + 2x_{jk}x + 2y_{jk}y + 2z_{jk}z}{2R_{kj}} \quad (2.21)$$

$$\sqrt{(x_k - x)^2 + (y_k - y)^2 + (z_k - z)^2} = \frac{R_{kl}^2 + x_k^2 - x_l^2 + y_k^2 - y_l^2 + z_k^2 - z_l^2 + 2x_{lk}x + 2y_{lk}y + 2z_{lk}z}{2R_{kl}} \quad (2.22)$$

จากนั้นนำสมการ (2.19) มาเท่ากับสมการที่ (2.20) แล้วจัดให้อยู่ในรูปสมการเชิงระนาบ

$$y = Ax + Bz + C \quad (2.23)$$

และในทำนองเดียวกัน นำสมการที่ (2.21) มาเท่ากับสมการที่ (2.22) แล้วจัดให้อยู่ในรูปสมการเชิงระนาบ

$$y = Dx + Ez + F \quad (2.24)$$

จากนั้นนำสมการที่ (2.23) มาเท่ากับสมการที่ (2.24) จะได้

$$Ax + Bz + C = Dx + Ez + F \quad (2.25)$$

$$x = Gz + H \quad (2.26)$$

โดยที่

$$G = \frac{E - B}{A - D} \quad (2.27)$$

และ

$$H = \frac{F - C}{A - D} \quad (2.28)$$

นำสมการที่ (2.26) แทนลงในสมการ (2.23) จะได้สมการเชิงเส้นสำหรับ  $y$  ในรูปของ  $z$

$$y = A(Gz + H) + Bz + C \quad (2.29)$$

$$y = Iz + J \quad (2.30)$$

โดยที่

$$I = AG + B \quad (2.31)$$

และ

$$J = AH + C \quad (2.32)$$

นำสมการที่ (2.26) และ (2.30) แทนกลับไปในสมการที่ (2.20) เพื่อหาค่า  $z$  จะได้

$$2R_{ik}\sqrt{(G^2z^2 - 2Gz(x_i - H) + (x_i - H)^2) + (I^2z^2 - 2Iz(y_i - J) + (y_i - J)^2) + (z^2 - 2z_i z + z_i^2)} = Lz + K \quad (2.33)$$

โดยที่

$$K = R_{ik}^2 + x_i^2 - x_k^2 + y_i^2 - y_k^2 + z_i^2 - z_k^2 + 2x_{ki}H + 2y_{ki}J \quad (2.34)$$

และ

$$L = 2(x_{ki}G + y_{ki}I + 2z_{ki}) \quad (2.35)$$

นำสมการ (2.33) ยกกำลังสองจะได้

$$4R_{ik}^2(G^2 + I^2 + 1)z^2 - 8R_{ik}^2(G(x_i - H) + I(y_i - J) + z_i)z + 4R_{ik}^2((x_i - H)^2 + (y_i - J)^2 + z_i^2) = L^2z^2 + 2KLz + K^2 \quad (2.36)$$

จากนั้นจัดรูปสมการ (2.36) ให้อยู่ในรูปสมการพหุนาม

$$Mz^2 - Nz + O = 0 \quad (2.37)$$

โดยที่

$$M = 4R_{ik}^2(G^2 + I^2 + 1) - L^2 \quad (2.38)$$

และ

$$N = 8R_{ik}^2(G(x_i - H) + I(y_i - J) + z_i) + 2LK \quad (2.39)$$

และ

$$O = 4R_{ik}^2((x_i - H)^2 + (y_i - J)^2 + z_i^2) - K^2 \quad (2.40)$$

จะได้คำตอบของ  $z$  คือ

$$z = \frac{N}{2M} \pm \sqrt{\left(\frac{N}{2M}\right)^2 - \frac{O}{M}} \quad (2.41)$$

จากนั้นนำค่า  $z$  ที่ได้ไปแทนกลับในสมการ (2.26) และ (2.30) เพื่อหาค่า  $x$  และ  $y$  ของวัตถุ

## 2.4 การหาค่าเหมาะที่สุดเชิงคอนเวกซ์

ในงานวิจัยหลายงานที่เกี่ยวข้องกับการระบุตำแหน่งหรือการร่วมกันระบุตำแหน่ง ได้เสนอขั้นตอนวิธีที่ใช้หลักการทางด้าน optimization เพื่อหาคำตอบ ซึ่งปัญหา optimization เองนั้นก็สามารถแบ่งออกได้เป็นหลายชนิดเช่นกัน ส่วนที่เกี่ยวข้องกับวิทยานิพนธ์ฉบับนี้นั้นคือ convex optimization ในส่วนนี้เราจะนำอธิบายถึงปัญหา optimization และจะอธิบายถึงในส่วน convex optimization เพียงอย่างเดียว ปัญหา optimization อื่นๆนั้นจะไม่ขอพูดถึง

### 2.4.1 ปัญหาการหาค่าเหมาะที่สุดเชิงคณิตศาสตร์

ปัญหาการหาค่าเหมาะที่สุดเชิงคณิตศาสตร์หรือเรียกสั้นๆว่าปัญหาการหาค่าเหมาะที่สุดนั้นจะมีรูปแบบดังนี้

$$\begin{aligned} & \text{minimize} && f_0(x) \\ & \text{subject to} && f_i(x) \leq b_i, i = 1, \dots, m. \end{aligned} \quad (2.42)$$

ในที่นี้ให้เวกเตอร์  $x$  แทนตัวแปรที่ต้องการจะหาค่าเหมาะที่สุดของปัญหา ฟังก์ชัน  $f_0 : \mathbb{R}^n \rightarrow \mathbb{R}$  เป็นฟังก์ชันวัตถุประสงค์ ฟังก์ชัน  $f_i : \mathbb{R}^n \rightarrow \mathbb{R}, i = 1, \dots, m$  เป็นฟังก์ชันเงื่อนไขบังคับ(อสมการ) และค่าคงที่  $b_1, \dots, b_m$  เป็นขีดจำกัดของเงื่อนไขบังคับ เวกเตอร์  $x^*$  จะเป็นค่าเหมาะที่สุดหรือคำตอบของปัญหาได้ ถ้ามันทำให้ค่าของฟังก์ชันวัตถุประสงค์มีค่าน้อยสุดในกลุ่มของเวกเตอร์ที่ตรงตามเงื่อนไขบังคับ: สำหรับ  $z$  ใดๆ ที่  $f_1(z) \leq b_1, \dots, f_m(z) \leq b_m$ , จะต้อง  $f_0(z) \geq f_0(x^*)$  ปัญหาการหาค่าเหมาะที่สุดที่ใช้กันโดยทั่วไปนั้นสามารถแบ่งย่อยๆได้เป็นหลายประเภทตามฟังก์ชันวัตถุประสงค์และฟังก์ชันเงื่อนไขบังคับ ยกตัวอย่างเช่น ปัญหาการหาค่าเหมาะที่สุด 2.42 จะเรียกว่ากำหนดการเชิงเส้นถ้าฟังก์ชันวัตถุประสงค์และฟังก์ชันเงื่อนไขบังคับเป็นเชิงเส้น นั่นก็คือจะตรงตามเงื่อนไข

$$f_i(\alpha x + \beta y) = \alpha f_i(x) + \beta f_i(y) \quad (2.43)$$

สำหรับทุกๆ  $x, y \in \mathbb{R}^n$  และทุกๆ  $\alpha, \beta \in \mathbb{R}$  ในกรณีปัญหาการหาค่าเหมาะที่สุดไม่เป็นเชิงเส้นนั้นจะเรียกว่า กำหนดการไม่เชิงเส้น ในงานวิจัยนี้จะสนใจเฉพาะปัญหาการหาค่าเหมาะที่สุดชนิดเดียวคือ ปัญหาการหาค่าเหมาะที่สุดแบบคอนเวกซ์ ปัญหาการหาค่าเหมาะที่สุดแบบคอนเวกซ์นั้นจะมีฟังก์ชันวัตถุประสงค์และฟังก์ชันเงื่อนไขบังคับเป็นคอนเวกซ์ ซึ่งหมายความว่าตรงตามเงื่อนไข

$$f_i(\alpha x + \beta y) \leq \alpha f_i(x) + \beta f_i(y) \quad (2.44)$$

สำหรับทุกๆ  $x, y \in \mathbb{R}^n$  และทุกๆ  $\alpha, \beta \in \mathbb{R}$  และ  $\alpha + \beta = 1, \alpha \geq 0, \beta \geq 0$  เมื่อเปรียบเทียบกันระหว่างสองปัญหานี้จะเห็นว่าปัญหาที่เป็นคอนเวกซ์จะสามัญมากกว่าปัญหาเชิงเส้น จากการที่กำหนดการเชิงเส้นนั้นก็เป็นปัญหาการหาค่าเหมาะที่สุดแบบคอนเวกซ์จะนั้นจึงกล่าวได้ว่าปัญหาการหาค่าเหมาะที่สุดแบบคอนเวกซ์นั้นเป็นสามัญการของกำหนดการเชิงเส้น ในส่วนย่อยถัดไปจะอธิบายถึงทฤษฎีที่เกี่ยวข้องกับคอนเวกซ์

### 2.4.2 เซตคอนเวกซ์

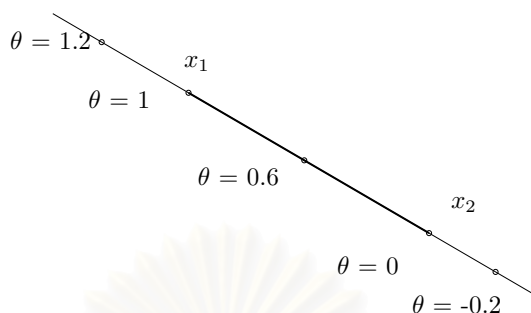
**เส้นตรงและส่วนของเส้นตรง** ให้  $x_1 \neq x_2$  และทั้งสองจุดนี้อยู่ใน  $\mathbb{R}^n$  ส่วนของเส้นตรงที่ลากผ่านจุดสองจุดนี้ก็จะสามารถเขียนได้อยู่ในรูป

$$y = \theta(x_1) + (1 - \theta)(x_2), \quad (2.45)$$

โดยที่  $\theta \in \mathbb{R}$  ในกรณีที่  $\theta = 0$  จะให้ค่า  $y = x_2$  และในกรณีที่  $\theta = 1$  จะให้ค่า  $y = x_1$  หมายความว่าถ้าค่า  $\theta$  อยู่ในช่วง 0 ถึง 1 นั้นจะแสดงถึงส่วนของเส้นตรงที่ลากจากจุด  $x_1$  ไปยัง  $x_2$  หรือเราสามารถเขียนให้อยู่ในอีกรูปแบบหนึ่งได้ดังนี้

$$y = x_2 + \theta(x_1 - x_2) \quad (2.46)$$

ซึ่งในรูปแบบนี้จะหมายถึงผลรวมของ จุดฐานจุดหนึ่ง  $x_2$  และทิศทาง  $x_1 - x_2$  ซึ่งจะขยายตาม  $\theta$  ดังนั้น  $\theta$  จึงแสดงถึงจุดระหว่างทางจาก  $x_2$  ไปยัง  $x_1$  และให้ค่า  $y$  ที่สัมพันธ์กับจุดนั้น เมื่อ  $\theta$  มีค่าเพิ่มขึ้นจาก 0 ไป ถึง 1 จุดจะเลื่อนจาก  $x_2$  ไปยัง  $x_1$  ในกรณีที่  $\theta > 1$  ค่า  $y$  จะเลยจุด  $x_1$  ไปดังในรูป[2.3]



รูปที่ 2.3: แสดงเส้นที่ลากผ่านจุด  $x_1$  และ  $x_2$  โดยอธิบายอยู่ในเชิงพารามิเตอร์โดย  $\theta x_1 + (1 - \theta)x_2$  โดยที่ค่า  $\theta$  นั้นจะเปลี่ยนได้ทั่วๆ ไป ส่วนของเส้นตรงระหว่างจุด  $x_1$  และ  $x_2$  จะสอดคล้องกับค่า  $\theta$  ที่อยู่ในช่วง 0 ถึง 1(สีที่เข้มกว่า)

**เซตสัมพรรค** เซต  $C \subseteq R^n$  จะเป็นเซตสัมพรรค(affine) ได้ก็ต่อเมื่อเส้นตรงที่ลากจากจุดสองจุดที่ต่างกัน ใน  $C$  อยู่ใน  $C$  เช่นสำหรับ  $x_1, x_2 \in C$  และ  $\theta \in \mathcal{R}$  เราจะสามารถเขียน  $\theta x_1 + (1 - \theta)x_2 \in C$  หรือจะกล่าวได้ว่า  $C$  ประกอบด้วยผลรวมเชิงเส้นของจุดสองจุดใดๆ ใน  $C$  โดยที่ผลรวมของสัมประสิทธิ์ทั้งสองตัวนั้นมีค่าเท่ากับ 1 แนวคิดนี้สามารถเขียนให้อยู่ในรูปทั่วไปสำหรับจุดหลายๆจุดได้ในที่นี้หมายถึงจุดที่อยู่ในรูปแบบ  $\theta_1 x_1 + \dots + \theta_k x_k$  โดยที่  $\theta_1 + \dots + \theta_k = 1$  คือผลรวมเชิงสัมพรรคของจุด  $x_1, \dots, x_k$  โดยการใช้การอุปนัยจากนิยามของเซตสัมพรรค(นั่นคือเซตสัมพรรคจะประกอบด้วยทุกๆผลรวมเชิงสัมพรรคของจุดสองจุดในเซต) เราสามารถแสดงให้เห็นว่าเซตสัมพรรคนั้นจะประกอบไปด้วยทุกๆผลรวมเชิงสัมพรรคของจุดในเซต : ถ้า  $C$  เป็นเซตสัมพรรคและ  $x_1, \dots, x_k \in C$  และ  $\theta_1 + \dots + \theta_k = 1$  แล้ว จุดที่เกิดจาก  $\theta_1 x_1 + \dots + \theta_k x_k$  ก็จะมีอยู่ในเซต  $C$  ด้วยถ้า  $C$  เป็นเซตสัมพรรคและ  $x_0 \in C$  แล้วเซต

$$V = C - x_0 = \{x - x_0 | x \in C\}$$

เป็นปริภูมิย่อย นั่นคือ จะมีคุณสมบัติปิดภายใต้การบวกและการคูณด้วยสเกลาร์ สามารถพิสูจน์ได้โดยกำหนดให้  $v_1, v_2 \in V$  และ  $\alpha, \beta \in \mathcal{R}$  แล้วจะได้  $v_1 + x_0 \in C$  และ  $v_2 + x_0 \in C$  และยังได้

$$\alpha v_1 + \beta v_2 + x_0 = \alpha(v_1 + x_0) + \beta(v_2 + x_0) + (1 - \alpha - \beta)x_0 \in C$$

จากการที่  $C$  เป็นเซตสัมพรรค และ  $\alpha + \beta + (1 - \alpha - \beta) = 1$  สามารถที่จะสรุปได้ว่า  $\alpha v_1 + \beta v_2 \in V$  และจากการที่  $\alpha v_1 + \beta v_2 + x_0 \in C$  แล้วเซต  $C$  สามารถเขียนได้อยู่ในรูป

$$C = V + x_0 = \{v + x_0 | v \in V\}$$

นั่นคืออยู่ในรูปของผลรวมของปริภูมิย่อยกับออฟเซต ปริภูมิย่อย  $V$  ที่สัมพันธ์กับเซตสัมพรรค  $C$  ไม่ได้ขึ้นอยู่กับกำเลือก  $x_0$  ดังนั้นสามารถที่จะเลือก  $x_0$  จากจุดใดๆก็ได้ใน  $C$  เราจะนิยาม "มิติ" ของเซตสัมพรรค  $C$



ว่าเป็น มิติของปริภูมิย่อย  $V = C - x_0$  โดยที่  $x_0$  เป็นสมาชิกใดๆในเซต  $C$

เซตของทุกๆผลรวมเชิงสัมพรรคของทุกการจัดกลุ่มของจุดในบางเซต  $C \in \mathbb{R}^n$  จะเรียกว่า เปลือกหุ้มเชิงสัมพรรคของ  $C$  และเขียนแทนด้วย  $\text{aff } C$ :

$$\text{aff } C = \{\theta_1 x_1 + \cdots + \theta_k x_k | x_1, \dots, x_k \in C, \theta_1 + \cdots + \theta_k = 1\}$$

เปลือกหุ้มเชิงสัมพรรคเป็นเซตสัมพรรคที่มีขนาดเล็กที่สุดที่สามารถเก็บ  $C$  ไว้ได้ หรือในอีกมุมมองหนึ่งคือ ถ้า  $S$  เป็นเซตสัมพรรคใดๆที่  $C \subseteq S$  แล้ว  $\text{aff } C \subseteq S$

**มิติเชิงสัมพรรค และภายในสัมพรรค** กำหนดให้มิติเชิงสัมพรรคของเซต  $C$  คือ มิติของเปลือกหุ้มเชิงสัมพรรคของตัวเอง มิติเชิงสัมพรรคจะมีประโยชน์ในเชิงของการวิเคราะห์เชิงคอนเวกซ์และการหาค่าเหมาะสมที่สุดแต่ไม่จำเป็นที่จะต้องสอดคล้องกับนิยามของมิติเชิงอื่นๆ ยกตัวอย่างเช่น วงกลมหนึ่งหน่วยใน  $\mathbb{R}^2$ ,  $\{x \in \mathbb{R}^2 | x_1^2 + x_2^2 = 1\}$  ฉะนั้นเปลือกหุ้มเชิงสัมพรรคของมันคือ  $\mathbb{R}^2$  ดังนั้นมิติเชิงสัมพรรคคือ 2 โดยนิยามทั่วไป วงกลมหนึ่งหน่วยใน  $\mathbb{R}^2$  นั้นจะมีมิติเท่ากับ 1 ถ้ามิติเชิงสัมพรรคของเซต  $C \subseteq \mathbb{R}^n$  น้อยกว่า  $n$  แล้วเซตนี้จะอยู่ในเซตสัมพรรค  $\text{aff } C \neq \mathbb{R}^n$  เรานิยามคำว่า "ภายในสัมพัทธ์(relative interior)" ของเซต  $C$  ว่าเป็นภายในสัมพัทธ์ของ  $\text{aff } C$  แล้วเขียนแทนด้วย  $\text{relint } C$

$$\text{relint } C = \{x \in C | B(x, r) \cap \text{aff } C \subseteq C \text{ for some } r > 0\}$$

โดยที่  $B(x, r) = \{y | \|y - x\| \leq r\}$  วงกลมรัศมี  $r$  และมีจุดศูนย์กลางอยู่ที่  $x$  เรายังสามารถที่จะนิยามขอบเขตสัมพัทธ์(relative boundary)ของเซต  $C$  เป็น  $\text{cl } C$  โดยที่  $\text{cl } C$  เป็นส่วนปิดคลุมของ  $C$

**เซตคอนเวกซ์** เซต  $C$  จะเป็นคอนเวกซ์ ถ้าส่วนของเส้นตรงที่ลากผ่านจุดสองจุดใดๆใน  $C$  อยู่ใน  $C$  หรือพูดอีกอย่างคือ ถ้าสำหรับจุดสองจุด  $x_1, x_2 \in C$  และสำหรับ  $\theta$  ใดๆที่มีค่า  $0 \leq \theta \leq 1$  เราจะได้

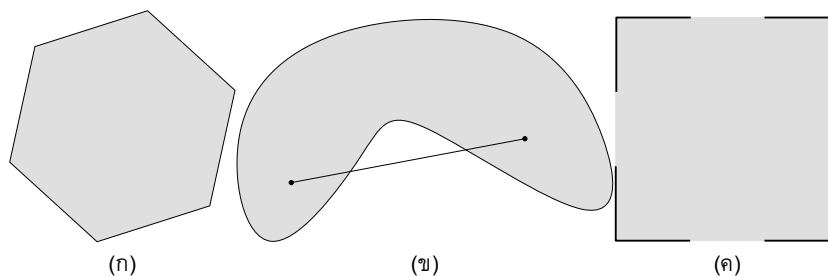
$$\theta x_1 + (1 - \theta)x_2 \in C$$

พูดอย่างง่ายก็คือ เซตจะเป็นคอนเวกซ์ได้ถ้าทุกๆจุดในเซตนั้นสามารถมองเห็นจุดอื่นๆได้ทั้งหมด โดยไม่ถูกกีดขวางเลย การที่ไม่ถูกกีดขวางนั้นหมายถึงเส้นตรงที่มองผ่านนั้นอยู่ในเซตทั้งหมด เซตสัมพรรคทั้งหมดเป็นคอนเวกซ์เนื่องจากมันประกอบด้วยเส้นที่ลากจากคู่จุดทุกคู่จุดอยู่ รูปที่ [2.4] แสดงถึงตัวอย่างของเซตที่เป็นคอนเวกซ์และเซตที่ไม่เป็นคอนเวกซ์ใน  $\mathbb{R}^2$

เราเรียกจุดในรูปแบบ  $\theta_1 x_1 + \cdots + \theta_k x_k$  ซึ่ง  $\theta_1 + \cdots + \theta_k = 1$  และ  $\theta_i \geq 0, i = 1, \dots, k$  ว่าผลรวมเชิงคอนเวกซ์ของจุด  $x_1, \dots, x_k$  และในทำนองเดียวกับเซตสัมพรรคที่สามารถจะแสดงให้เห็นว่า เซตจะเป็นคอนเวกซ์ก็ต่อเมื่อมันประกอบด้วยทุกๆผลรวมเชิงคอนเวกซ์ของจุดของมัน เราสามารถมองผลรวมเชิงคอนเวกซ์ของจุดเป็นการประมาณแบบถ่วงน้ำหนักของกลุ่มจุด โดยให้  $\theta_i$  เป็นตัวถ่วงของ  $x_i$

เปลือกหุ้มเชิงคอนเวกซ์ของเซต  $C$  คือเซตของทุกการจัดกลุ่มของผลรวมเชิงคอนเวกซ์ของจุดใน  $C$



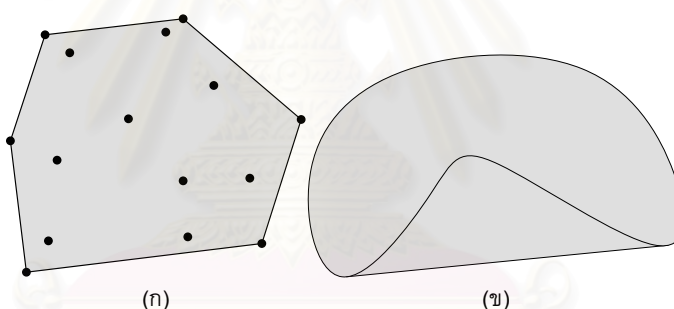


รูปที่ 2.4: แสดงตัวอย่างอย่างง่ายของเซตคอนเวกซ์และเซตเว้า ก. รูปหกเหลี่ยมที่รวมถึงขอบ(เส้นทึบ)เป็นคอนเวกซ์ ข. รูปไตไม่เป็นคอนเวกซ์เนื่องจากมีส่วนของเส้นตรงที่ลากเชื่อมจุดสองจุดในเซตและไม่ได้อยู่ในเซต ค. รูปสี่เหลี่ยมที่ประกอบด้วยขอบเพียงบางส่วนไม่เป็นคอนเวกซ์

เขียนแทนด้วย  $\text{conv } C$

$$\text{conv } C = \{\theta_1 x_1 + \dots + \theta_k x_k \mid x_i \in C, \theta_i \geq 0, i = 1, \dots, k, \theta_1 + \dots + \theta_k = 1\}$$

เปลือกหุ้มเชิงคอนเวกซ์จะเป็นคอนเวกซ์เสมอ และจะเป็นคอนเวกซ์เซตที่เล็กที่สุดที่มี  $C$  อยู่ภายใน : ถ้า  $B$  เป็นคอนเวกซ์เซตใดๆที่มี  $C$  อยู่ด้านใน แล้ว  $\text{conv } C \subseteq B$  รูปที่ [2.5] แสดงตัวอย่างของเปลือกหุ้มเชิง



รูปที่ 2.5: แสดงเปลือกหุ้มเชิงคอนเวกซ์ของเซตสองเซตที่อยู่ใน  $\mathbb{R}^2$  ก. เปลือกหุ้มเชิงคอนเวกซ์ของจุดจำนวน 15 จุดคือรูปหกเหลี่ยม(ที่แรเงา) ข. เปลือกหุ้มเชิงคอนเวกซ์ของรูปไตที่แสดงในรูป 2.4 ข. ก็คือส่วนที่แรเงาทงหมด

คอนเวกซ์

แนวคิดของผลรวมเชิงคอนเวกซ์สามารถนำมาทำให้อยู่ในรูปทั่วไปได้เพื่อจะนำผลรวมของจำนวนอนันต์ตัวมาใช้ ผลรวมของจำนวนอนันต์ตัวนั้นก็เช่นการหาปริพันธ์ และรูปแบบทั่วไปของการกระจายตัวของความน่าจะเป็น กำหนดให้  $\theta_1, \theta_2, \dots$  ซึ่งทำให้

$$\theta_i \geq 0, i = 1, 2, \dots, \sum_{i=1}^{\infty} \theta_i = 1$$

เป็นจริงและ  $x_1, x_2, \dots \in C$  โดยที่  $C \subseteq \mathbb{R}^n$  เป็นคอนเวกซ์ แล้ว

$$\sum_{i=1}^{\infty} \theta_i x_i \in C$$

โดยมีเงื่อนไขคืออนุกรมนี้จะต้องลู่เข้า อาจเขียนให้อยู่ในรูปที่ทั่วไปมากกว่านี้ได้ก็คือ กำหนดให้  $p : \mathfrak{R}^n \rightarrow \mathfrak{R}$  ที่ทำให้  $p(x) \geq 0$  สำหรับทุกๆ  $x \in C$  และ  $\int_C p(x) dx = 1$  โดยที่  $C \subseteq \mathfrak{R}^n$  เป็นคอนเวกซ์ แล้ว

$$\int_C p(x)x dx \in C$$

โดยจะต้องสามารถหาปริพันธ์ได้

ในรูปแบบทั่วไป กำหนดให้  $C \subseteq \mathfrak{R}^n$  เป็นคอนเวกซ์และ  $x$  เป็นเวกเตอร์สุ่ม โดย  $x \in C$  ด้วยความน่าจะเป็นหนึ่ง แล้ว  $\mathbf{E}x \in C$  ซึ่งจริงๆแล้วรูปแบบนี้จะรวมถึงรูปแบบอื่นๆด้วยในลักษณะของกรณีพิเศษ ยกตัวอย่างเช่น กำหนดให้ตัวแปรสุ่ม  $x$  รับข้อมูลเข้าเพียงสองข้อมูล  $x_1$  และ  $x_2$  ด้วยความน่าจะเป็น  $\text{prob}(x = x_1) = \theta$  และ  $\text{prob}(x = x_2) = 1 - \theta$  โดย  $0 \leq \theta \leq 1$  แล้วจะได้  $\mathbf{E}x = \theta x_1 + (1 - \theta)x_2$  และก็กลับมาสู่รูปแบบปกติของผลรวมเชิงคอนเวกซ์ของจุดสองจุด

### 2.4.3 ฟังก์ชันคอนเวกซ์ : คุณสมบัติพื้นฐาน

**นิยาม** ฟังก์ชัน  $f : \mathfrak{R}^n \rightarrow \mathfrak{R}$  จะเป็นคอนเวกซ์ ถ้า  $\text{dom } f$  เป็นคอนเวกซ์เซต และถ้าสำหรับทุกๆ  $x, y \in \text{dom } f$  และ  $\theta$  อยู่ในช่วง  $0 \leq \theta \leq 1$  จะได้ว่า

$$f(\theta x + (1 - \theta)y) \leq \theta f(x) + (1 - \theta)f(y) \quad (2.47)$$

ในเชิงเรขาคณิตแล้ว อสมการนี้จะหมายถึง ส่วนของเส้นตรงที่ลากจากจุด  $x, f(x)$  และ  $y, f(y)$  ใดๆ จะอยู่เหนือกราฟ  $f$  ดังแสดงในรูป [2.6] ฟังก์ชันนี้จะเป็นคอนเวกซ์เข้มงวดถ้าอสมการที่ 2.47 นั้นไม่มีการเท่า



รูปที่ 2.6: แสดงกราฟของฟังก์ชันคอนเวกซ์ เส้นพาด(ส่วนของเส้นตรง)ระหว่างจุดสองจุดใดๆของกราฟจะอยู่เหนือกราฟ

กันเมื่อ  $x \neq y$  และ  $0 \leq \theta \leq 1$  เราจะกล่าวว่า  $f$  นั้นเว้าถ้า  $-f$  นั้นเป็นคอนเวกซ์ และ จะเป็นเว้าเข้มงวดถ้า  $-f$  เป็นคอนเวกซ์เข้มงวด

สำหรับ ฟังก์ชันสัมพรรคเราจะสามารถหาอสมการ 2.47 ได้เสมอ ฉะนั้นสำหรับทุกๆฟังก์ชันสัมพรรค(รวมถึงฟังก์ชันเชิงเส้น)จะเป็นทั้งคอนเวกซ์และเว้า และในทางกลับกันฟังก์ชันใดก็ตามที่เป็นทั้งคอนเวกซ์และเว้า ก็จะเป็นสัมพรรคด้วย

## 2.4.4 ปัญหาการหาค่าเหมาะที่สุดเชิงคอนเวกซ์

ปัญหาการหาค่าเหมาะที่สุด : พื้นฐาน เราใช้สัญกรณ์

$$\begin{aligned} & \text{minimize} && f_0(x) \\ & \text{subject to} && f_i(x) \leq b_i, \quad i = 1, \dots, m \\ & && h_i(x) = 0, \quad i = 1, \dots, p \end{aligned} \tag{2.48}$$

เพื่อใช้ในการอธิบายปัญหาการหาค่า  $x$  ที่ทำให้ค่า  $f_0(x)$  มีค่าน้อยที่สุด จากกลุ่มของ  $x$  ทุกตัวที่ทำให้เงื่อนไข  $f_i(x) \leq b_i, i = 1, \dots, m$  และ  $h_i(x) = 0, i = 1, \dots, p$  นี้เป็นจริง เราเรียก  $x \in \mathcal{X}^n$  ว่าเป็นตัวแปรของการหาค่าเหมาะที่สุด และฟังก์ชัน  $f_0 : \mathcal{X}^n \rightarrow \mathcal{R}$  ว่าเป็นฟังก์ชันจุดประสงค์ หรือฟังก์ชันค่า อสมการ  $f_i(x) \leq 0$  ว่าเป็นอสมการข้อบังคับและฟังก์ชันที่สัมพันธ์กัน  $f_i : \mathcal{X}^n \rightarrow \mathcal{R}$  ว่าเป็นฟังก์ชันอสมการข้อบังคับสมการ  $h_i(x) = 0$  ว่าเป็นสมการข้อบังคับและฟังก์ชัน  $h_i : \mathcal{X}^n \rightarrow \mathcal{R}$  ว่าเป็นสมการข้อบังคับ ถ้าในกรณีที่ไม่มีการข้อบังคับเลย ( $m = p = 0$ ) เราจะเรียกปัญหา 2.48 ว่าไม่มีข้อบังคับ

เราจะเรียกเซตของจุดที่อยู่ในโดเมนของฟังก์ชันจุดประสงค์และข้อบังคับทุกข้อ

$$\mathcal{D} = \bigcap_{i=1}^m \text{dom } f_i \cap \bigcap_{i=1}^p \text{dom } h_i$$

ว่าเป็นโดเมนของปัญหาการหาค่าเหมาะที่สุด 2.48 จุด  $x \in \mathcal{D}$  จะเรียกว่าจุดซึ่งเป็นไปได้ก็ต่อเมื่อ ถ้าสามารถทำให้เงื่อนไข  $f_i(x) \leq 0, i = 1, \dots, m$  และ  $h_i(x) = 0, i = 1, \dots, p$  เป็นจริง ปัญหา 2.48 จะถูกเรียกว่าเป็นไปได้ถ้ามีจุดซึ่งเป็นไปได้อย่างน้อยหนึ่งจุด และจะถูกเรียกว่าเป็นไปไม่ได้ในทางกลับกัน เซตของจุดซึ่งเป็นไปได้นั้นจะถูกเรียกว่าเซตซึ่งเป็นไปได้หรือเรียกว่าเซตข้อบังคับ

ค่าเหมาะที่สุด  $p^*$  สำหรับปัญหา 2.48 จะนิยามได้ดังนี้

$$p^* = \inf\{f_0(x) \mid f_i(x) \leq 0, i = 1, \dots, m, h_i(x) = 0, i = 1, \dots, p\}$$

เรายอมให้ค่า  $p^*$  นั้นมีค่าเป็น  $\pm\infty$  ได้ ถ้าปัญหานั้นเป็นไปไม่ได้จะได้ว่าค่า  $p^* = \infty$  ถ้ามีจุดซึ่งเป็นไปได้  $x_k$  ที่  $f_0(x_k) \rightarrow -\infty$  ในขณะที่  $x_k \rightarrow \infty$  แล้ว  $p^* = -\infty$  และเราจะเรียกว่าปัญหา 2.48 นี้ไม่จำกัดขอบเขตล่าง

**Optimal and locally optimal points** เราจะกล่าวว่า  $x^*$  เป็นค่าเหมาะที่สุดหรือคำตอบของปัญหา 2.48 ถ้า  $x^*$  เป็นไปได้และ  $f_0(x^*) = p^*$  เซตของจุดที่เป็นค่าเหมาะที่สุดทั้งหมดเรียกว่าเซตค่าเหมาะที่สุดและเขียนแทนได้ด้วย

$$X_{opt} = \{x \mid f_i(x) \leq 0, i = 1, \dots, m, h_i(x) = 0, i = 1, \dots, p, f_0(x) = p^*\}$$

ถ้ามีจุดที่เป็นค่าเหมาะที่สุดสำหรับปัญหา 2.48 เราจะกล่าวว่า ค่าเหมาะที่สุดนั้นคงอยู่ หรือจะเรียกว่า ปัญหานี้สามารถที่จะแก้ได้ถ้า  $X_{opt}$  ว่างนั้นเราจะกล่าวว่า ค่าเหมาะที่สุดนั้นไม่คงอยู่หรือปัญหานี้ไม่สามารถแก้

ได้ (ซึ่งโดยส่วนมากจะเกิดขึ้นเมื่อปัญหานี้ไม่จำกัดขอบเขตล่าง) จุดซึ่งเป็นไปได้  $x$  ที่  $f_0(x) \leq p^* + \epsilon$  (โดยที่  $\epsilon > 0$ ) จะเรียกว่า  $\epsilon$ -ค่าเหมาะที่สุดย่อย และเซตของทุก  $\epsilon$ -ค่าเหมาะที่สุดย่อยนั้นจะเรียกว่าเซตค่าเหมาะที่สุดย่อยของปัญหา 2.48

เราจะกล่าวว่าจุดซึ่งเป็นไปได้  $x$  เป็นค่าเหมาะที่สุดเฉพาะแห่ง ถ้ามี  $R > 0$  ที่ทำให้

$$f_0(x) = \inf\{f_0(z) \mid f_i(z) \leq 0, i = 1, \dots, m, h_i(z) = 0, i = 1, \dots, p, \|z - x\|_2 \leq R\}$$

หรือในอีกมุมมองหนึ่งคือ  $x$  สามารถที่จะแก้ปัญหาการหาค่าเหมาะที่สุด

$$\begin{aligned} & \text{minimize} && f_0(z) \\ & \text{subject to} && f_i(z) \leq b_i, \quad i = 1, \dots, m \\ & && h_i(z) = 0, \quad i = 1, \dots, p \\ & && \|z - x\|_2 \leq R \end{aligned}$$

ในรูปตัวแปร  $z$  หรืออาจจะกล่าวอย่างง่าย ๆ ได้คือ  $x$  ทำให้ค่า  $f_0$  น้อยที่สุดในบริเวณจุดรอบๆ ที่เป็นจุดซึ่งเป็นไปได้ ในที่นี้จะใช้คำว่า "ค่าเหมาะสมที่สุดแทนค่าเหมาะสมที่สุดสำหรับส่วนรวม" ซึ่งจะแตกต่างจากคำว่า "ค่าเหมาะสมที่สุดเฉพาะแห่ง"

ถ้า  $x$  เป็นจุดซึ่งเป็นไปได้ และ  $f_i(x) = 0$  เราจะกล่าวว่า อสมการข้อบังคับลำดับที่  $i$   $f_i(x) \leq 0$  นั้นกัมมันต์ (active) ถ้า  $f_i(x) < 0$  เราจะกล่าวว่าข้อบังคับ  $f_i(x) \leq 0$  นั้นอากัมมันต์ (สมการข้อบังคับนั้นจะกัมมันต์ในทุกจุดซึ่งเป็นไปได้) เราจะเรียกข้อบังคับว่าซ้ำซ้อนถ้าลบข้อบังคับนั้นออกไปแล้วเซตซึ่งเป็นไปได้นั้นยังไม่เปลี่ยนแปลง

**Feasibility problems** ถ้าฟังก์ชันวัตถุประสงค์เป็นศูนย์และทำให้ค่าเหมาะสมที่สุดเป็นทั้งศูนย์ (ในกรณีที่เซตซึ่งเป็นไปได้ไม่ว่าง) หรือ  $\infty$  (ถ้าเซตซึ่งเป็นไปได้ว่าง) แล้วเราจะเรียกปัญหานี้ว่า ปัญหาความเป็นไปได้ และอาจจะเขียนได้ดังรูป

$$\begin{aligned} & \text{find} && x \\ & \text{subject to} && f_i(x) \leq b_i, \quad i = 1, \dots, m \\ & && h_i(x) = 0, \quad i = 1, \dots, p \end{aligned}$$

ปัญหาความเป็นไปได้ก็คือการหาจุดที่จะทำให้ข้อบังคับนั้นถูกต้อง และข้อบังคับทุกอย่างสอดคล้องกัน

**รูปแบบมาตรฐานของปัญหาการหาค่าเหมาะที่สุดเชิงคอนเวกซ์** ปัญหาการหาค่าเหมาะสมที่สุดเชิงคอนเวกซ์จะมีลักษณะดัง

$$\begin{aligned} & \text{minimize} && f_0(x) \\ & \text{subject to} && f_i(x) \leq b_i, \quad i = 1, \dots, m \\ & && a_i^T x = b_i, \quad i = 1, \dots, p \end{aligned} \tag{2.49}$$

โดยที่  $f_0, \dots, f_m$  เป็นคอนเวกซ์ฟังก์ชัน เมื่อเปรียบเทียบปัญหา 2.49 กับปัญหาในรูปแบบทั่วไป 2.48 ปัญหาเชิงคอนเวกซ์นั้นมีข้อแตกต่างดังนี้

- ฟังก์ชันวัตถุประสงค์ต้องเป็นคอนเวกซ์
- อสมการข้อบังคับต้องเป็นคอนเวกซ์
- ฟังก์ชันสมการข้อบังคับ  $h_i(x) = a_i^T x_i - b_i$  จะต้องเป็นสัมพรรค

เราจะเห็นคุณสมบัติที่สำคัญอย่างหนึ่งนั่นคือ : เซตของจุดซึ่งเป็นไปได้ของปัญหาการหาค่าเหมาะสมที่สุดเชิงคอนเวกซ์เป็นคอนเวกซ์ เนื่องจากมันเกิดจากการตัดกันของโดเมนของปัญหา

$$D = \bigcap_{i=0}^m \text{dom } f_i$$

ซึ่งเป็นคอนเวกซ์เซตที่มี  $m$  (คอนเวกซ์) เซตชั้นย่อย  $\{x | f_i(x) \leq 0\}$  และ ระบายเกิน  $p \{x | a_i^T x = b_i\}$  (เราสามารถจะกล่าวได้ว่า  $a_i \neq 0$  โดยที่ไม่สูญเสียหลักการทั่วไป เนื่องจากถ้า  $a_i = 0$  และ  $b_i = 0$  สำหรับ  $i$  บางตัวแล้ว สมการข้อบังคับลำดับที่  $i$  นั้นสามารถที่จะลบทิ้งได้ หรือถ้า  $a_i = 0$  และ  $b_i \neq 0$  สมการข้อบังคับในลำดับที่  $i$  จะไม่สอดคล้องกัน และจะทำให้ปัญหานั้นเป็นไปได้) ดังนั้นปัญหาการหาค่าเหมาะสมที่สุดเชิงคอนเวกซ์นั้นจะเป็นการหาค่าต่ำสุดของฟังก์ชันจุดประสงค์บนเซตคอนเวกซ์

**จุดค่าเหมาะสมที่สุดเฉพาะแห่งและจุดค่าเหมาะสมที่สุดของทั้งหมด** คุณสมบัติพื้นฐานของปัญหาการหาค่าเหมาะสมที่สุดเชิงคอนเวกซ์นั่นคือ ในจุดค่าเหมาะสมที่สุดเฉพาะแห่งใดๆจะเป็นจุดค่าเหมาะสมที่สุดของทั้งหมดด้วย เพื่อเป็นการพิสูจน์กำหนดให้  $x$  เป็นจุดค่าเหมาะสมที่สุดเฉพาะแห่งของปัญหาการหาค่าเหมาะสมที่สุดเชิงคอนเวกซ์ ก็คือ  $x$  จะเป็นไปได้และ

$$f_0(x) = \inf \{f_0(z) | z \text{ feasible}, \|z - x\|_2 \leq R\} \quad (2.50)$$

สำหรับ  $R > 0$  บางค่า แล้วกำหนดให้  $x$  ไม่เป็นค่าเหมาะสมที่สุดของทั้งหมด นั่นคือ จะมี  $y$  ที่เป็นไปได้ที่ทำให้  $f_0(y) < f_0(x)$  ซึ่งจะเห็นได้ชัดว่า  $\|y - x\|_2 > R$  เพราะไม่เช่นนั้นแล้ว  $f_0(x) < f_0(y)$  พิจารณาจุด  $z$  ที่ได้จาก

$$z = (1 - \theta)x + \theta y, \theta = \frac{R}{2 \|y - x\|_2}$$

แล้วจะได้  $\|z - x\|_2 = R/2 < R$  และด้วยความเป็นคอนเวกซ์ของกลุ่มซึ่งเป็นไปได้  $z$  ก็จะเป็นไปได้ โดยความเป็นคอนเวกซ์ของ  $f_0$  เราจะได้

$$f_0(z) \leq (1 - \theta)f_0(x) + \theta f_0(y) < f_0(x)$$

ซึ่งจะขัดกับ 2.50 ดังนั้นก็จะมี  $y$  ซึ่งเป็นไปได้ที่ทำให้  $f_0(y) < f_0(x)$  หรือก็คือ  $x$  เป็นค่าเหมาะสมที่สุดของทั้งหมด

สามารถหาข้อมูลเพิ่มเติมเกี่ยวกับปัญหาการหาค่าเหมาะสมที่สุดได้จาก [5]



## บทที่ 3

### งานวิจัยที่เกี่ยวข้อง

ในช่วงเวลาไม่กี่ปีที่ผ่านมาได้มีงานวิจัยที่เกี่ยวข้องกับการร่วมกัน ระบุตำแหน่ง เป็นจำนวนมาก เนื่องจากข้อดีของการทำงานร่วมกันและจากการพัฒนาทางด้านขั้นตอนการผลิตและอุปกรณ์อิเล็กทรอนิกส์ต่างๆทำให้สามารถที่จะผลิตของจำนวนมากโดยใช้ค่าใช้จ่ายน้อยลงได้ ซึ่งงานวิจัยที่เกี่ยวข้องกับปัญหานี้แบ่งออกได้เป็นสามด้านใหญ่ๆนั่นคือ ด้านกลุ่มของหุ่นยนต์ ด้านเครือข่ายการรับรู้ และด้านอื่นๆ โดยต่อไปจะนำเสนองานวิจัยที่เกี่ยวข้องในแต่ละด้านและเสนอถึงข้อเปรียบเทียบของแต่ละงานวิจัยที่อยู่ในด้านเดียวกัน

#### 3.1 งานด้านกลุ่มของหุ่นยนต์

[6] เป็นงานวิจัยแรกที่มีการพูดถึงการช่วยกันระบุตำแหน่งของหุ่นยนต์ โดยได้นำเสนอขั้นตอนวิธีที่ใช้หุ่นยนต์สองกลุ่ม โดยแต่ละกลุ่มนั้นทำหน้าที่สลับกันไปมาระหว่างเคลื่อนที่ไปยังเป้าหมายและทำตัวเป็นจุดอ้างอิง เนื่องจากทีมผู้วิจัยได้เสนอปัญหาว่าการระบุตำแหน่งโดยใช้ dead reckoning นั้นจะมีความคลาดเคลื่อนที่ค่อนข้างสูงเมื่อมีการเคลื่อนที่เป็นระยะทางไกลๆ และอีกปัญหาคือการใช้จุดอ้างอิงในสภาพแวดล้อมเป็นวัฏจักรนั้นก็ไม่สามารถนำไปใช้งานได้ ในสภาพแวดล้อมที่ยังไม่รู้จักนอกจากนี้ยังได้เสนอสมการที่แสดงความคลาดเคลื่อนเมื่อมีการเคลื่อนที่เป็นแบบสามเหลี่ยมต่อกันไปเรื่อย และได้ทดลองในแบบจำลองซึ่งให้ผลลัพธ์ที่ตรงกับสมการที่เสนอไว้

[7]เป็นงานที่ต่อจาก [6] โดยได้เสนอขั้นตอนวิธีคล้ายๆกับงานเดิม แต่มีการเปลี่ยนแปลงคือ ในกลุ่มหุ่นยนต์นั้นจะมีหุ่นหนึ่งตัวที่จะสามารถวัดระยะทางและมุมระหว่างหุ่นยนต์ตัวอื่นๆได้ ขั้นตอนการเคลื่อนที่คือ หุ่นยนต์ตัวอื่นๆที่ไม่ใช่หุ่นหลักจะเคลื่อนที่ไปยังจุดหมายพร้อมๆกันโดยหุ่นยนต์ตัวหลักจะอยู่กับที่ เมื่อหุ่นยนต์ตัวอื่นๆเคลื่อนที่เรียบร้อยแล้วหุ่นยนต์ตัวหลักจึงจะเคลื่อนที่ โดยหุ่นยนต์ตัวที่ทำหน้าที่ระบุตำแหน่งคือหุ่นยนต์ตัวหลักเท่านั้น และในบทความนี้ได้มีการเสนอข้อเสียของการช่วยกันระบุตำแหน่งแบบนี้ นั่นคือ 1.จำเป็นจะต้องมีหุ่นยนต์อย่างน้อยสองตัว 2. จะต้องมีการระบุตำแหน่งอย่างน้อย 1 ตัวที่อยู่กับที่ ในหนึ่งหน่วยเวลา ทำให้เวลาในการทำงานทั้งหมดนั้นเพิ่มขึ้น และ 3.ต้องมีการติดตั้งอุปกรณ์การวัดไว้บนหุ่นยนต์ทุกตัว(สำหรับงาน[6]) และได้มีการปรับปรุงเพิ่มเติมใน [8] โดยได้มีการเสนอกฎหุ่นยนต์แบบใหม่ โดยให้ชื่อว่า CPS-III และในบทความนี้ได้เสนอความคลาดเคลื่อนที่จะเกิดจากการเคลื่อนที่แบบต่างๆสามแบบ

[9],[10] ได้นำเสนอขั้นตอนวิธีการช่วยกันระบุตำแหน่งของหุ่นยนต์โดยใช้วิธีการทางด้านสถิติ ด้วยการให้ probability density function ในการแทนที่ตำแหน่งของหุ่นยนต์และเพื่อป้องกันความซับซ้อนเชิงเวลาที่จะเพิ่มมากขึ้นแบบไม่เป็นพหุนามสำหรับหุ่นยนต์หลายๆตัว ทางทีมวิจัยจึงได้เสนอการแทนตำแหน่งของหุ่นยนต์ทั้งหมดด้วยการที่ให้หุ่นยนต์แต่ละตัวเก็บ เฉพาะค่าความเชื่อมั่นของตัวเองเท่านั้น การที่ใช้วิธีการทางด้านสถิติเข้ามาช่วยนั้นช่วยให้สามารถที่จะระบุตำแหน่งของหุ่นยนต์ได้โดยที่เริ่มจากไม่รู้ตำแหน่งของตัวเองเลยได้ ขั้นตอนวิธีหลักที่ใช้ในการระบุตำแหน่งในงานวิจัยนี้คือการใช้ Monte Carlo Localization(MCL) และได้เสนอวิธีการส่งผ่านความเชื่อมั่นจากหุ่นยนต์ตัวหนึ่งไปยังหุ่นยนต์อีกตัวหนึ่งเมื่อหุ่นยนต์ทั้งสองนี้สามารถตรวจสอบกันเจอ การทดลองในงานนี้ได้ใช้หุ่นยนต์สองตัว แต่ละตัวถูกติดตั้ง

ไว้ด้วย laser range-finder ,กล้องเพื่อตรวจหาหุ่นยนต์อีกตัวหนึ่ง และแถบสีเพื่อให้ง่ายในการตรวจหา ผลการทดลองแสดงให้เห็นว่าการช่วยกันระบุตำแหน่งนั้นทำให้ความรวดเร็วในการระบุตำแหน่งและความแม่นยำของตำแหน่งของหุ่นยนต์ทั้งสองตัวมากขึ้น

[11]นำเสนอการช่วยกันระบุตำแหน่งเพื่อการสำรวจพื้นที่ โดยการใช้อุปกรณ์หาตำแหน่งคล้ายๆกับ[6] คือใช้หุ่นยนต์เป็นจุดอ้างอิงเคลื่อนที่ ในงานนี้ยังได้เสนอวิธีการหาตำแหน่งของสิ่งกีดขวางและเสนอวิธีการหลบหลีกแบบง่ายๆอีกด้วย ขั้นตอนวิธีที่นำเสนอเพื่อใช้ในการสำรวจพื้นที่นั้นแบ่งเป็นหลายๆแบบโดยขึ้นอยู่กับขนาดของพื้นที่ เมื่อพื้นที่มีขนาดเล็ก(หุ่นยนต์ทั้งสองตัวนั้นจะเห็นกันตลอดเวลา) ก็จะใช้วิธี triangulation แต่ถ้าพื้นที่มีขนาดใหญ่ก็จะใช้วิธี trapezoidation เพื่อตัดแบ่งพื้นที่ให้เล็กลงแล้วสำรวจทีละส่วนย่อยๆ ข้อจำกัดสำหรับขั้นตอนวิธีนี้ก็คือจำเป็นที่จะต้องทราบสภาพแวดล้อมที่จะทำการสำรวจล่วงหน้า

[12]ได้เสนอวิธีใหม่เพื่อใช้ในการช่วยกันระบุตำแหน่งของกลุ่มหุ่นยนต์ ทางที่มวิจัยได้ใช้ Kalman filter เป็นวิธีหลักในการระบุตำแหน่ง และได้เสนอวิธีที่จะนำไปประยุกต์ใช้กับกลุ่มของหุ่นยนต์ในแบบ centralized โดยข้อมูลทั้งหมดจะถูกส่งไปยังที่ศูนย์กลางเพื่อคำนวณ นอกจากนั้นยังได้เสนอวิธีการที่เป็นแบบ decentralized ด้วย ซึ่งหุ่นยนต์แต่ละตัวจะทำการคำนวณความไม่แน่นอนของตัวเองจากการเคลื่อนที่ของตัวเองโดยไม่ขึ้นอยู่กับหุ่นยนต์ตัวอื่นแล้วเมื่อมีการตรวจพบหุ่นยนต์ตัวอื่นก็จะมีการแลกเปลี่ยนข้อมูลตำแหน่ง ทิศทางและทำการแลกเปลี่ยนค่า covariance กันระหว่างหุ่นยนต์ทุกตัวในกลุ่มทางผู้เขียนไม่ได้ทำการทดลองเปรียบเทียบผลกับขั้นตอนวิธีอื่นๆทำให้ผลการทดลองยังไม่ค่อยชัดเจนในเรื่องความแม่นยำที่ได้รับเพิ่มขึ้น

[13]ได้นำเสนอการนำ Monte Carlo localization มาใช้ โดยได้อธิบายถึง MCL ทั่วไป ซึ่งจะมีปัญหาในกรณีที่สัญญาณรบกวนในส่วนของข้อมูลที่ใช้นั้นมีค่าน้อยมาก จะส่งผลให้ผลลัพธ์ที่ได้แย่ง ซึ่งขัดกับความรู้สึกโดยทั่วไปทางที่มวิจัยได้อธิบายถึงเหตุผลของปรากฏการณ์นี้ ซึ่งเป็นผลมาจากการที่ข้อมูลที่รับเข้ามานั้นแคบเกินไปผู้เขียนจึงได้เสนอ วิธีการนำการนำ proposal distribution มารวมกันโดยให้ผลลัพธ์ที่ดีกว่าอย่างชัดเจนเมื่อเทียบกับMCL ทั่วไปและ dual

[14]ได้อธิบายถึงการช่วยกันระบุตำแหน่งของหุ่นยนต์เพื่อใช้ในการขนย้ายสิ่งของ โดยได้นำวิธีทางด้านคอมพิวเตอร์วิทัศน์มาใช้เพื่อระบุตำแหน่งสัมพัทธ์และทิศทาง และเพื่อรักษาารูปแบบของการวางตัวของหุ่นยนต์ ซึ่งหุ่นยนต์แต่ละตัวจะถูกติดไว้ด้วยแถบสีเฉพาะแต่ละตัวเพื่อใช้ในการแยกแยะ และตัวรับรู้ที่ใช้สำหรับหาตำแหน่งของหุ่นยนต์ตัวอื่นๆคือ omni-directional camera ที่มวิจัยได้ใช้ข้อมูลทั้งที่เป็นเชิงมุมและเชิง ระยะทางจากกล้องชนิดนี้ หลังจากได้ข้อมูลทั้งหมดแล้วจึงนำมาคำนวณหาตำแหน่งของหุ่นยนต์แต่ละตัวโดยใช้สมการที่ได้คิดขึ้นมา แล้วจึงนำไปเข้าส่วนของการควบคุมรูปแบบการวางตัวต่อไป โดยขั้นตอนทั้งหมดนั้นจะถูกรวมมาคิดที่หุ่นยนต์เพียงตัวเดียว(centralized)

[15]งานวิจัยฉบับนี้ได้อธิบายถึง ขั้นตอนวิธีการระบุตำแหน่ง ของกลุ่มหุ่นยนต์แบบใหม่ซึ่งสามารถทนทานต่อความผิดพลาดที่จะเกิดขึ้นได้งานนี้ได้ใช้หุ่นยนต์ขนาดเล็ก (Millibots) ในการวิจัยหุ่นยนต์ขนาดเล็กแต่ละตัวสามารถวัด ระยะทางระหว่างตัวเองกับตัวอื่นโดยใช้ตัวรับรู้ที่เป็น ultrasonic และจะนำค่าไปรวมกับค่าระยะทางที่ได้จากการเคลื่อนที่โดยใช้วิธี maximum likelihood estimator งานวิจัยชิ้นนี้ได้เน้นไปในส่วนที่การเลือกข้อมูลการวัดที่มีความผิดพลาดสูงออก ซึ่งข้อมูลเหล่านี้มักจะเกิดขึ้นในขั้นตอนการระบุตำแหน่ง สาเหตุที่ทำให้เกิดความผิดพลาดเช่นค่าความผิดพลาดจากการวัดระยะทางเมื่อเกิดการชนเข้า



กับวัตถุที่ตรวจสอบไม่พบ หรือเกิดจากการที่เส้นทางการเคลื่อนที่ของเสียง ultrasonic ที่ไม่เป็นเส้นตรง และสามารถเกิดการสะท้อนได้ ทีมวิจัยได้เสนอขั้นตอนวิธีเพื่อใช้ตรวจสอบความผิดพลาดจากการใช้ข้อมูล ทั้งสองแบบ แล้วตรวจสอบว่ามีความผิดพลาดเกิดขึ้นหรือไม่ถ้าข้อมูลใดที่ผิดพลาดก็จะไม่ใช่ข้อมูลนั้นเพื่อคำนวณตำแหน่ง

[16]งานนี้ได้อธิบายถึงวิธีที่จะช่วยกันระบุตำแหน่งโดยใช้ตัวหุ่นยนต์เองเป็นจุดอ้างอิงวิธีนี้จะทำให้หุ่นยนต์แต่ละตัวสามารถที่จะหาระยะทาง, ทิศทางระหว่างหุ่นยนต์ตัวอื่นๆได้ รวมถึงทิศของตัวเองได้อีกด้วย โดยไม่ใช่จุดอ้างอิงจากภายนอก, GPS และแผนที่ของสภาพแวดล้อมเลย ซึ่งการที่จะใช้วิธีนี้ได้หุ่นยนต์แต่ละตัวจำเป็นต้องสามารถหาระยะทางระหว่างตัวเองและหุ่นยนต์ตัวอื่นๆได้ และจำเป็นที่จะต้องรู้ระยะทางและทิศทางที่เคลื่อนที่ไป ได้มีการนำ Maximum likelihood estimator (MLE) และ numerical optimization มาใช้เพื่อคำนวณหาตำแหน่งของหุ่นยนต์ทุกๆตัวในทีมโดย Howard et al. ได้นำเสนอสมการอย่างง่าย ๆ เพื่อนำมาใช้ในกระบวนการ และมีการทดลองกับหุ่นยนต์จริงจำนวน 4 ตัวอีกด้วย ทางทีมที่วิจัยยังไม่ได้วิเคราะห์ในอีกหลายด้านเช่น ผลกระทบเมื่อมี local minima เกิดขึ้นในขั้นตอนการ optimization และยังไม่ได้ศึกษาการทำงานเมื่อมีหุ่นยนต์จำนวนหลายๆตัว

[17]ได้ทำการทดลองกับขั้นตอนวิธีการช่วยกันระบุตำแหน่งที่ใช้ข้อมูลแบบต่างๆที่มีมาก่อนหน้านี้ โดยมีการทดลองเปรียบเทียบกับ 4 วิธี วิธีแรกคือใช้แค่เพียงเฉพาะข้อมูลที่เป็นระยะทางอย่างเดียวแล้วใช้หุ่นยนต์เป็นจุดอ้างอิงเคลื่อนที่ 2. ใช้เพียงมุมอย่างเดียว 3. ใช้การคำนวณตำแหน่งของหุ่นยนต์ตัวอื่น 4. ใช้ข้อมูลทุกอย่าง นอกจากนี้ในงานวิจัยนี้ยังได้ทำการทดลองและวัดผลเปรียบเทียบระหว่างการเคลื่อนที่สองแบบระหว่างการเคลื่อนที่แบบเป็นระเบียบและแบบสุ่ม ในแต่ละการทดลองนั้นทีมวิจัยได้ทำการทดลองเทียบกับจำนวนหุ่นยนต์หลายๆแบบด้วย ผลการทดลองแสดงให้เห็นว่าเมื่อจำนวนหุ่นยนต์เพิ่มขึ้นนั้นความแม่นยำในการหาตำแหน่งจะเพิ่มสูงขึ้น ยกเว้นแต่ในกรณีที่ใช้เพียงแค่มุมอย่างเดียว และการเคลื่อนที่แบบสุ่มให้ค่าความคลาดเคลื่อนที่น้อยกว่าแบบเป็นระเบียบ

[18]งานนี้ใช้ขั้นตอนวิธีคล้ายๆกับงาน [12] โดยมีการเพิ่มเติมในส่วนของการทดลองมากขึ้น โดยมีการทดลองหลายๆแบบเช่น แบบที่ไม่มีหุ่นยนต์ตัวไหนเลยที่รู้ตำแหน่งสัมบูรณ์ แบบที่สองคือมีหุ่นยนต์อย่างน้อย 1 ตัวที่รู้ตำแหน่งสัมบูรณ์ และอีกแบบคือมีอย่างน้อย 1 ตัวที่อยู่กับที่ ผลการทดลองแสดงให้เห็นว่า การที่แลกเปลี่ยนข้อมูลกันเฉพาะในกรณีที่ไม่มีหุ่นยนต์ตัวไหนรู้ตำแหน่งสัมบูรณ์แสดงให้เห็นว่า การที่แลกเปลี่ยนข้อมูลกันเฉพาะในกรณีที่สามารถตรวจสอบกันเจอเท่านั้นสามารถที่ให้ความแม่นยำดีกว่าในกรณีที่ไม่มีแลกเปลี่ยนข้อมูล แต่แยกว่าในกรณีที่มีการแลกเปลี่ยนข้อมูลกันตลอดเวลา สำหรับในกรณีที่มีหุ่นยนต์อย่างน้อย 1 ตัวที่รู้ตำแหน่งของตัวเองตลอดเวลาหรือในกรณีที่มีหุ่นยนต์หยุดนิ่งอยู่หนึ่งตัวนั้นจะให้ความแม่นยำที่สูงขึ้นซึ่งเปรียบเทียบเสมือนมีการใช้ตัวรับรู้ร่วมกัน

[19]ได้เสนอการช่วยกันระบุตำแหน่งที่ใช้ขั้นตอนวิธีในทำนองเดียวกันกับของ [10] แต่ต่างกันตรงที่ในงานนี้จะไม่มีการอ้างอิงถึงตำแหน่งสัมบูรณ์ของหุ่นยนต์ และไม่มีการใช้ตัวรับรู้ประเภทที่สามารถรับรู้สภาพแวดล้อมได้ ผลลัพธ์ของงานวิจัยนี้จะเป็ตำแหน่งอ้างอิงของหุ่นยนต์แต่ละตัวเทียบกันเองภายในทีมงานนี้ใช้ขั้นตอนวิธีหลักๆแบบเดียวกับ [10] นั่นคือการใช้ MCL

[20]ได้เสนอขั้นตอนวิธีแบบใหม่เพื่อใช้ในการช่วยกันระบุตำแหน่งของกลุ่มหุ่นยนต์โดยวิธีการที่เสนอนั้นเป็นแบบ centralized และจำเป็นที่จะต้องรู้แผนที่ก่อนล่วงหน้า ทางทีมวิจัยได้เสนอวิธีที่ใช้ข้อมูล

เชิงมุมและระยะทางของตำแหน่งของหุ่นยนต์อีกตัวหนึ่งอ้างอิงกับตัวเอง และข้อมูลระยะทางรอบๆตัว โดยอาศัย infrared range sensor และกล้อง วิธีที่นำเสนอแนะอาศัยการหาตำแหน่งของกำแพง และมุม มาเพื่อระบุตำแหน่ง ทีมที่วิจัยไม่ได้นำเสนอการทดลองในกรณีที่มีความคลาดเคลื่อนในระบบและไม่ได้นำเสนอผลการทดลองที่บอกถึงความคลาดเคลื่อนจากการใช้วิธีนี้อย่างชัดเจนนัก และไม่มีเปรียบเทียบกับขั้นตอนวิธีอื่นๆ ข้อจำกัดหลักๆของขั้นตอนวิธีนี้คือ จำนวนของหุ่นยนต์มากขึ้นและความจำเป็นที่จะต้องรู้แผนที่ล่วงหน้า

[4]ได้เสนอวิธีการใหม่สำหรับการช่วยกันระบุตำแหน่ง โดยใช้รูปแบบความผิดพลาดแบบจำกัด Spletzer et al. ได้สร้าง convex polytope ที่เป็นตัวแทนของ configuration space ของหุ่นยนต์ทุกๆตัวซึ่งสัมพันธ์กันกับข้อมูลที่ได้จากตัววัดต่างๆ ค่าตอบที่ต้องการจะทราบนั้น(เช่น ค่าตำแหน่งจริง, ค่าตำแหน่งสัมพันธ์ของสองตัว)จะสามารถหาได้จากการ project polytope นี้ลงบน subspace ที่เหมาะสม และทางผู้เขียนได้เสนอวิธีการประมาณการ projection นี้โดยใช้เทคนิคทางด้าน linear programming วิธีการที่นำเสนอแนะสามารถรับข้อมูลได้ทั้งแบบที่เป็นเชิงมุมและเชิงระยะทาง โดยความซับซ้อนเชิงเวลาของขั้นตอนวิธีนั้นขึ้นเป็นพหุคูณของจำนวนหุ่นยนต์

[21]ได้นำเสนอสองปัญหาที่เกิดขึ้นในการระบุตำแหน่งของกลุ่มหุ่นยนต์ที่ใช้อุปกรณ์ราคาไม่แพง และได้วิเคราะห์ ความซับซ้อนเชิงเวลาของปัญหาและ เสนอวิธีการแก้ปัญหามาแบบ ประมาณ ปัญหาแรกคือการลดจำนวนรอบของการรับข้อมูล ที่ตัวรับรู้ไม่สามารถทำงานพร้อมๆกันสำหรับหุ่นยนต์หลายๆตัวได้(เนื่องจากปัญหาของการ cross-talk) ทางทีมวิจัยได้เสนอการแทนรูปแบบการรับข้อมูลของกลุ่มหุ่นยนต์ด้วย cross-talk graph แล้วได้ทำการพิสูจน์ว่าปัญหาการลดจำนวนรอบนั้นเป็น NP-complete โดยใช้ปัญหาการระบายสีกราฟมาช่วย ส่วนปัญหาที่สองคือปัญหาการลดบริเวณความไม่แน่นอนโดยใช้จำนวนการรับข้อมูลให้น้อยที่สุด บริเวณความไม่แน่นอนแทนความเป็นไปได้ของตำแหน่งของหุ่นยนต์แต่ละตัวซึ่งเกิดขึ้นจากการเคลื่อนที่และวัดโดยใช้ dead-reckoning บริเวณนี้จะถูกแทนด้วยวงรีความไม่แน่นอนซึ่งจะมีลักษณะความไม่แน่นอนแบบ gaussian หลายตัวแปร ทางทีมวิจัยได้เสนอ feature-visibility graph ซึ่งจุดในกราฟนี้จะแทนจุดอ้างอิง หรือตัวหุ่นยนต์และเส้นที่เชื่อมระหว่างจุดนั้นจะแทนการรับรู้ ยิ่งไปกว่านั้นทางทีมวิจัยยังได้พิสูจน์ว่าปัญหาที่สองนี้เป็น NP-complete ด้วยเช่นกัน โดยใช้ปัญหา job-scheduling เข้ามาช่วยทั้งสองปัญหานั้นทางทีมวิจัยได้เสนอวิธีในการแก้ปัญหามาแบบประมาณมาไว้ในบทความชิ้นนี้ด้วย

[22]ในงานวิจัยชิ้นนี้ได้เสนอขั้นตอนวิธีการช่วยกันระบุตำแหน่งของกลุ่มหุ่นยนต์ซึ่งเป็นแบบ distributed นั่นคือจะไม่มีหุ่นยนต์หลักที่ทำหน้าที่ระบุตำแหน่งของหุ่นยนต์ตัวอื่นๆ ทางทีมที่วิจัยได้เสนอเงื่อนไขเพื่อนำไปใช้ในการเลือกข้อมูล ที่หุ่นยนต์แต่ละตัวรับเข้ามาเพื่อลดความไม่แน่นอนของผลลัพธ์ นอกจากนั้นทางทีมยังได้แสดงให้เห็นว่าการเลือกนี้สามารถที่จะนำไปใช้กับระบบแบบ distributed ได้อีกด้วยและยังได้เสนอว่าขั้นตอนวิธีนี้สามารถที่จะนำไปใช้กับระบบที่มีหุ่นยนต์หลายๆชนิดได้(ซึ่งมีตัวรับรู้ที่แตกต่างกันออกไป) ขั้นตอนวิธีการระบุตำแหน่งหลักๆของงานนี้คือการใช้ EKF เพื่อใช้ในการปรับปรุงสถานะของตำแหน่งของหุ่นยนต์ หุ่นยนต์ที่ใช้ในการทดลองนี้จะถูกติดตั้งไว้ด้วยตัววัดระยะทาง(range finder) และตัววัดการเคลื่อนที่(odometric sensor)

[23]ได้นำเสนอขั้นตอนวิธีการช่วยกันระบุตำแหน่งของกลุ่มหุ่นยนต์จำนวน 4 ตัวในสภาพแวดล้อมที่เป็น 3 มิติ และได้เสนอวิธีการเคลื่อนที่ของกลุ่มหุ่นยนต์เพื่อใช้ในการระบุตำแหน่งสองวิธี การเคลื่อนที่ทั้งสองวิธีนั้นได้ถูกวิเคราะห์ และนำไปทดลองใช้งานจริงๆ ทั้งใน simulator และทดลองกับหุ่นยนต์จริงๆ

ผลลัพธ์ที่ได้ขึ้นอยู่กับเงื่อนไขที่ยอมรับได้ ขั้นตอนวิธีที่ใช้สำหรับการหาตำแหน่งนั้นได้ถูกพัฒนาเพิ่มเติมมาจากงานของ [14] ซึ่งได้พัฒนาเพิ่มในส่วนการหา scale factor สำหรับหาตำแหน่งในโลกจริง ในงานของ [14] นั้นไม่ได้มีการหาในส่วนนี้ไว้และการลดข้อจำกัดบางอย่างลงไปอีกด้วย ในงานนี้ได้หาค่านี้มาจากการเคลื่อนที่ของหุ่นยนต์หลายตัว หุ่นยนต์ที่ใช้ในงานนี้แบ่งเป็นหุ่นยนต์ 3 ตัวที่เคลื่อนที่อยู่บนพื้นในระนาบเดียวกันส่วนหุ่นยนต์อีกตัวหนึ่งนั้นเคลื่อนที่อยู่ทีเพดาน ในงานวิจัยนี้ได้เสนอการนำหุ่นยนต์ที่ไม่เหมือนกันมาใช้ นั่นคือหุ่นยนต์แต่ละตัวนั้นจะถูกติดไว้ด้วยตัวรับรู้ที่ต่างชนิดกัน แต่ยังสามารถที่จะทำงานในการหาตำแหน่งร่วมกันได้

[24]ในงานวิจัยชิ้นนี้ได้อธิบายถึงขั้นตอนวิธีในการช่วยกันระบุตำแหน่งของกลุ่มหุ่นยนต์ ซึ่งกลุ่มหุ่นยนต์นี้จะมีเพียงแค่บางตัวเท่านั้นที่มีอุปกรณ์ที่ใช้ในการหาตำแหน่งจริงๆ ของตัวเอง ส่วนหุ่นยนต์ตัวอื่นนั้นจะรับตำแหน่งของหุ่นยนต์ตัวที่รู้ตำแหน่งของตัวเองแล้วหาความแรงของสัญญาณมาเพื่อทำการหาตำแหน่งของตัวเองอีกทีหนึ่ง ในบทความชิ้นนี้ได้เสนอวิธีเพื่อลดการใช้พลังงานของหุ่นยนต์แต่ละตัวที่เกิดจากการส่งข้อมูลระหว่างกัน วิธีที่นำเสนอคือการ synchronize เวลาระหว่างหุ่นยนต์ทุกๆตัวเพื่อจะกำหนดช่วงเวลาที่จะใช้ในการส่งข้อมูล และระหว่างนั้นจะไม่มีมีการส่งข้อมูลตำแหน่งระหว่างกัน ในช่วงที่ไม่ได้รับข้อมูลตำแหน่งของหุ่นยนต์ตัวอื่นนั้น หุ่นยนต์แต่ละตัวจะใช้เพียงข้อมูลระยะทางและทิศทางที่ตัวเองเคลื่อนที่ไปนั้นมาเพื่อหาตำแหน่ง

[25]ได้เสนอการวิเคราะห์อย่างละเอียดสำหรับการช่วยกันระบุตำแหน่งของกลุ่มหุ่นยนต์ที่แต่ละตัวมีคุณสมบัติไม่เหมือนกัน และได้เสนอการเปรียบเทียบกันแบบต่างๆ เช่นจำนวนของหุ่นยนต์ ความแม่นยำของตัวรับรู้ของหุ่นยนต์และ โครงสร้างการรับรู้ของหุ่นยนต์(กราฟที่แสดงให้เห็นถึงการรับรู้ระหว่างหุ่นยนต์) สมการที่ได้แนะนำเสนอนั้นจะช่วยในการประมาณค่าความคลาดเคลื่อนที่จะเกิดขึ้นและจะช่วยทำให้การออกแบบกลุ่มของหุ่นยนต์เพื่อนำไปใช้ในงานต่างๆเป็นไปได้อย่างสะดวกมากขึ้น ขั้นตอนวิธีที่ใช้ในการระบุตำแหน่งที่นำมาวิเคราะห์ความคลาดเคลื่อนนั้นคือ Kalman filter ผลการทดลองแสดงให้เห็นว่าเมื่อถึงจุดสมดุลค่าความคลาดเคลื่อนจะเพิ่มขึ้นโดยไม่ขึ้นอยู่กับตัวรับรู้ของหุ่นยนต์และกราฟแสดงการรับรู้ระหว่างหุ่นยนต์ นอกเหนือจากจำนวนของหุ่นยนต์แล้วยังมีอีกปัจจัยหลักๆอีกอย่างหนึ่งที่มีผลต่อความไม่แน่นอนของระบบนั้นคือ ความแม่นยำของตัวรับรู้การเคลื่อนที่และทิศทางของตัวหุ่นยนต์ จากผลการสังเกตจะเห็นว่าจำนวนของค่าการวัดที่ได้รับนั้นจะส่งผลต่อค่าคงที่ติดอยู่กับ covariance ในขณะที่อัตราความไม่แน่นอนเพิ่มเท่าเดิม นอกจากนั้นยังได้แสดงให้เห็นว่าเมื่อกราฟการรับรู้นั้นไม่สมบูรณ์หรือมีการเปลี่ยนแปลงเนื่องมาจากการบัง, ข้อจำกัดต่างๆของระบบที่ทำให้ไม่สามารถตรวจพบได้ครบทุกตัวนั้นก็จะมีส่งผลกระทบต่อประสิทธิภาพของระบบ

[26]ได้เสนอแนวการช่วยกันระบุตำแหน่งแบบใหม่ของกลุ่มหุ่นยนต์ซึ่งใช้คอมพิวเตอร์วิทัศน์เป็นหลักและสภาพแวดล้อมที่เคลื่อนไหวตลอด ขั้นตอนวิธีที่นำเสนอประกอบไปด้วยการผสานกันของสองขั้นตอนนั้นคือส่วน online ที่ทำหน้าที่หาตำแหน่งของหุ่นยนต์ด้วยการนำข้อมูลที่ได้จากกล้องมาใช้ด้วย ขั้นตอนวิธีการระบุตำแหน่งแบบ Markov แต่เมื่อไม่สามารถที่จะหาจุดอ้างอิงในภาพได้ภายในระยะเวลาที่กำหนดจะมีการนำส่วนที่สองมาใช้ นั่นคือส่วน offline ซึ่งเป็นระบบย่อยที่อาศัยประสบการณ์(experience subsystem) มาเพื่อช่วยในการระบุตำแหน่งโดยการคำนวณหาจุดเด่นในภาพแล้วนำมาเทียบกับข้อมูลที่ทำการประมวลผลไว้ก่อนแล้ว นอกจากวิธีการระบุตำแหน่งสำหรับหุ่นยนต์ตัวเดียวแล้วทางทีมวิจัยยังได้เสนอขั้นตอนวิธีในการช่วยกันระบุตำแหน่งของกลุ่มหุ่นยนต์อีกด้วย โดยการเสนอแนวคิดของวัตถุอ้างอิงเคลื่อนไหวได้มาเพื่อทำการช่วยในการระบุตำแหน่ง เมื่อใดก็ตามที่หุ่นยนต์บางตัวไม่สามารถหาจุด



อ้างอิงจากภาพได้เลยแต่สามารถที่จะมองเห็นจุดอ้างอิงเคลื่อนไหวย่างเดียวกับหุ่นยนต์ตัวอื่นๆก็จะนำค่าตำแหน่งของจุดอ้างอิงเคลื่อนไหวที่ได้จากหุ่นยนต์ตัวอื่นๆมาเพื่อช่วยในการระบุตำแหน่งของตนเอง

### 3.2 งานด้านเครือข่ายการรับรู้

[27]ได้เสนอวิธีการระบุตำแหน่งของเครือข่ายการรับรู้โดยใช้วิธีที่ไม่ซับซ้อนและเป็นแบบ distributed ในเครือข่ายนี้จะมีจุดอ้างอิงซึ่งรู้ตำแหน่งของตัวเองอยู่แล้วแล้วจุดอ้างอิงเหล่านี้จะทำหน้าที่ส่งสัญญาณตำแหน่งของตัวเองให้จุดอื่นๆรับรู้ เมื่อจุดเหล่านี้ได้รับสัญญาณมาก็จะนำข้อมูลที่ได้รับทั้งหมดมาหาตำแหน่งของตัวเองโดยการนำข้อมูลทั้งหมดมาเฉลี่ยตำแหน่งออกมา เนื่องจากเป็นวิธีที่ออกแบบมาใช้สำหรับตัวรับรู้ที่ราคาถูกและไม่ซับซ้อนมากนักจะเห็นได้ว่าความแม่นยำของวิธีการนี้ขึ้นอยู่กับปัจจัยจำนวนมากทั้งที่ควบคุมได้และควบคุมไม่ได้ ในบทความชิ้นนี้ไม่ได้นำเสนอในส่วนของการทดลองที่มีการควบคุมตัวแปรตัวอื่นๆมากนักทำให้ไม่ทราบถึงความแม่นยำที่แท้จริงของขั้นตอนวิธี

[28]ได้เสนอขั้นตอนวิธีเพื่อใช้ในการหาตำแหน่งที่จะใช้วางจุดอ้างอิงเพิ่มเติมเพื่อเพิ่มความแม่นยำให้กับการระบุตำแหน่งของเครือข่ายตัวรับรู้ขั้นตอนวิธีที่นำเสนอมานั้นเป็นแบบ distributed ซึ่งจะช่วยลดการใช้พลังงานของแต่ละจุดลง เพราะมีการลดปริมาณข้อมูลก่อนที่จะมีการส่งวิธีที่นำเสนอนี้มีชื่อว่า HEAP ซึ่งขั้นตอนการระบุตำแหน่งที่ใช้อ้างอิงในงานวิจัยนี้คืองาน [27] ทางทีมผู้วิจัยได้นำเสนอวิธีการหาตำแหน่งสำหรับวางมาสองวิธีนั่นคือ HEAP-Max และ HEAP-Grid และได้ทำการทดลองโดยเทียบผลกันในแบบต่าง ซึ่งก็ให้ข้อดีข้อเสียที่แตกต่างกันออกไป แต่ที่น่าสังเกตคือทางผู้ทดลองได้ทำการเปรียบเทียบผลกับการวางตำแหน่งแบบสุ่มด้วย ซึ่งให้ผลลัพธ์ที่ดีกว่าวิธีที่ทางผู้ทดลองได้เสนอในงานวิจัยชิ้นนี้ผู้วิจัยได้เสนอแนวคิดอีกอย่างหนึ่งที่มีชื่อว่า STROBE มาเพื่อใช้สำหรับเลือกปิดหรือเปิดสัญญาณของจุดอ้างอิงบางจุดเพื่อเพิ่มอายุการใช้งานของจุดอ้างอิงในสภาพแวดล้อมที่มีพลังงานจำกัด โดยที่ไม่ทำให้ความแม่นยำในการระบุตำแหน่งลดลงไปมาก

[29]เสนอวิธีการระบุตำแหน่งของเครือข่ายการรับรู้ วิธีการที่เสนอนี้เป็นวิธีแบบ centralized นั่นคือจะมีจุดที่ทำหน้าที่ระบุตำแหน่งของจุดทุกจุดอยู่ข้อมูลที่ใช้ในการระบุตำแหน่งของงานนี้คือ connectivity ระหว่างจุด, ระยะทางมากที่สุดที่สัญญาณจะไปถึง หรือ อาจจะมีมุมของสัญญาณที่ได้รับจากจุดต่างๆ ประกอบอยู่ด้วย ขั้นตอนวิธีที่ใช้ในการคำนวณตำแหน่งของจุดต่างๆนั้นมาจากการใช้ linear หรือ semidefinite program ในงานวิจัยชิ้นนี้ได้เสนอวิธีการสร้าง constraints ต่างๆเพื่อใช้ในการคำนวณ constraints ที่งานวิจัยชิ้นนี้ได้เสนอคือ constraint ของ connectivity ต่างๆ และ constraint ของมุมการรับข้อมูล ในส่วนของ constraint ที่เป็น connectivity นั้นผู้วิจัยได้เสนอไว้สองแบบคือ แบบระยะทางคงที่ และแบบระยะทางแปรผันได้ซึ่งได้ทำการทดลองและแสดงผลว่าระยะทางแบบแปรผันได้นั้นทำให้ได้ความแม่นยำของคำตอบมากกว่า การวางจุดอ้างอิงที่รู้ตำแหน่งของตัวเองไว้รอบๆบริเวณของจุดทั้งหมดนั้นจะทำให้ค่าความคลาดเคลื่อนลดลงด้วย นอกจากนี้ทางทีมผู้วิจัยยังได้เสนอ ขั้นตอนเพื่อลดความคลาดเคลื่อนลงอีกแต่เพิ่มการคำนวณมากขึ้นด้วยการนำสี่เหลี่ยมมาใช้ในการกำหนดขอบเขตของบริเวณที่เป็นไปได้ของคำตอบ ในส่วนของการใช้ข้อมูลเชิงมุมนั้นงานนี้ได้แสดงให้เห็นว่าเมื่อความคลาดเคลื่อนเชิงมุมลดลงก็จะส่งผลให้ความคลาดเคลื่อนของคำตอบลดลงไปด้วย และระยะทางจำกัดของข้อมูลเชิงมุมนี้ก็ยังคงส่งผลต่อประสิทธิภาพโดยรวมคือ เมื่อระยะทางจำกัดมากความคลาดเคลื่อนก็จะมากขึ้นตาม สุดท้ายในงานวิจัยนี้ได้ทดลองให้เห็นว่าเมื่อ connectivity มากขึ้นก็จะส่งผลให้ความคลาดเคลื่อนน้อยลง

[30]เสนอวิธีการช่วยกันระบุตำแหน่งของเครือข่ายการรับรู้(WSN) ซึ่งใช้ข้อมูลเชิงระยะทางเป็นหลักด้วยการวัด TOA ของ ultrasound ขั้นตอนวิธีที่เสนอนั้นเป็นแบบ distributed เนื่องจากทางทีมผู้วิจัยให้เหตุผลมาว่า 1.การที่ต้องส่งข้อมูลกลับไปยังส่วนกลางที่จะใช้ในการระบุตำแหน่งนั้นแต่ละจุดจำเป็นต้องรู้เส้นทางที่จะส่งไปหาส่วนกลางก่อน โดยอาจจะใช้วิธีอื่นที่ไม่อาศัยข้อมูลตำแหน่ง 2. จำเป็นที่จะต้องมีการ synchronize เวลากันและเมื่อแต่ละจุดมีการเปลี่ยนตำแหน่งจะยังไม่สามารถรู้ตำแหน่งของตัวเองได้ในทันที 3. ตำแหน่งที่วางของจุดกลางนั้นจำเป็นต้องมีการวางแผนไว้ล่วงหน้าบ้างแล้วเพื่อให้จุดอื่นๆเข้าถึงได้ง่ายและจุดอื่นๆที่อยู่ใกล้ๆจุดกลางนั้นจะใช้พลังงานมากกว่าจุดอื่นๆอีกด้วย 4. ความทนทานของระบบจะแยลงเมื่อเส้นทางส่งข้อมูลมาที่จุดกลางเสียหาย 5.จำเป็นที่จะต้องมีการส่งข้อมูลให้น้อยลง ขั้นตอนการระบุตำแหน่งที่แสดงไว้ในบทความนี้แบ่งออกเป็นสองส่วนหลักๆ 1. ส่วนที่ทำหน้าที่หาตำแหน่งเมื่อมีจุดอ้างอิงเพียงพอและเมื่อจุดนั้นทำการหาตำแหน่งของตัวเองเรียบร้อยแล้วก็จะทำหน้าที่เป็นจุดอ้างอิงต่อไป แล้วจุดอื่นๆก็จะทำงานเช่นนี้ไปเรื่อยๆจนครบและ 2. ส่วนที่ทำการช่วยกันระบุตำแหน่งเมื่อจุดอ้างอิงไม่เพียงพอ ในส่วนแรกนั้นสามารถทำได้โดยการใช้วิธี multilateration ธรรมดา และในส่วนที่สองนั้นได้มีการกำหนดเงื่อนไขในการที่จะใช้ในการช่วยกันระบุตำแหน่ง เมื่อเงื่อนไขนั้นครบถ้วนก็สามารถที่จะระบุตำแหน่งได้แต่ก็ยังมีความเป็นไปได้ว่าอาจจะไม่สามารถหาตำแหน่งของจุดบางจุดได้เนื่องจากความไม่สมบูรณ์ของกราฟ

[31]ได้เสนอวิธีการช่วยกันระบุตำแหน่งแบบใหม่ ซึ่งไม่อาศัยข้อมูลเชิงระยะทางเลย ใช้เพียงแคข้อมูลตำแหน่งของจุดอ้างอิงและข้อมูลความแรงของสัญญาณเท่านั้นขั้นตอนวิธีที่ได้อธิบายนั้นชื่อว่า Approximated Point-In-Triangulation Test (APIT) ซึ่งเป็นการประมาณมาจาก PIT อีกทีหนึ่ง ระบบโดยรวมจะมีจุดอ้างอิงซึ่งจะมีอุปกรณ์สำหรับระบุตำแหน่งเทียบกับโลกจริง(GPS) ซึ่งจะกระจายอยู่ในเครือข่าย ขั้นตอนวิธีที่นำเสนอเป็นแบบ distributed นั่นคือจุดแต่ละจุดจะทำหน้าที่หาตำแหน่งของตัวเองวิธีการที่ใช้หลักคือการทดสอบว่าจุดนั้นอยู่ในสามเหลี่ยมของจุดอ้างอิงหรือไม่ จุดอ้างอิงที่จะนำมาใช้สร้างสามเหลี่ยมนั้นคือจุดอ้างอิงที่สามารถรับข้อมูลตำแหน่งได้หลังจากนั้นจะประมาณการทดสอบด้วยการเทียบความแรงของสัญญาณกับจุดที่ไม่ใช่จุดอ้างอิงรอบๆตัว หลังจากนั้นก็จะคำนวณหาจุดที่อยู่ในสามเหลี่ยมที่เกิดจากการสลับเปลี่ยนจุดอ้างอิงทุกอัน มากที่สุด ทางทีมผู้วิจัยได้ทำการทดสอบเปรียบเทียบกับขั้นตอนวิธีการช่วยกันระบุตำแหน่งแบบอื่นๆ และได้สรุปว่า APIT นั้นสามารถทำงานได้ดีที่สุดเมื่อลักษณะของสัญญาณวิทยุไม่เป็นระเบียบ และจุดต่างๆนั้นมีการกระจายตัวแบบสุ่ม และทางทีมผู้วิจัยได้นำไปทดสอบกับการใช้งานที่จำเป็นต้องรู้ตำแหน่งของจุดด้วย และได้สรุปว่าความแม่นยำที่ได้นั้นเพียงพอสำหรับการใช้งานนั้นมีเพียงประสิทธิภาพที่ลดลงเพียงเล็กน้อยเท่านั้น

[32]นำเสนอขั้นตอนวิธีการระบุตำแหน่งของเครือข่ายการรับรู้(WSN) โดยใช้อุปกรณ์ที่มีราคาไม่แพง แนวความคิดหลักๆที่ใช้คือการใช้การติดต่อสื่อสารระหว่างจุดต่อจุดโดยมีการกระจายข้อมูลจากจุดอ้างอิงที่มีอุปกรณ์ระบุตำแหน่งสมบูรณ์(เช่น GPS) แล้วข้อมูลเหล่านี้จะส่งจากจุดหนึ่งไปยังจุดหนึ่ง ในระหว่างที่ส่งนั้นจะมีการนับจำนวนครั้งที่ส่งไปด้วย หลังจากที่จุดๆหนึ่งรับข้อมูลของจุดอ้างอิงมากกว่า 3 จุดแล้วจุดนั้นก็สมารถที่จะหาตำแหน่งของตัวเองได้โดยวิธีการ multilateration ซึ่งใช้จำนวนครั้งที่ส่งมาคำนวณหาตำแหน่ง ขั้นตอนวิธีนี้เป็นขั้นตอนวิธีแบบ distributed ซึ่งจุดแต่ละจุดนั้นจะคำนวณหาตำแหน่งของตัวเองโดยตรง ในงานวิจัยชิ้นนี้ได้กล่าวถึงขั้นตอนเพิ่มเติมสำหรับปรับปรุงคุณภาพของตำแหน่งที่ได้รับวิธีแรกคือการนำข้อมูลจำนวนครั้งที่ส่งจากจุดรอบๆตัวมาประมาณ และวิธีที่สองคือการนำความแรงของสัญญาณมาใช้เพื่อความละเอียดให้กับการวัดระยะการส่งซึ่งให้ผลลัพธ์ที่ดีขึ้นอย่างเห็นได้ชัด จะเห็นได้ว่าขั้นตอนวิธีที่เสนอมานี้ใช้การติดต่อสื่อสารค่อนข้างมากซึ่งจะทำให้สิ้นเปลืองพลังงาน ถ้าในกรณีที่จุดแต่ละ

จุดไม่มีการเคลื่อนที่ก็สามารถที่จะหาตำแหน่งเพียงครั้งเดียวได้ แต่ถ้ามีจุดใดจุดหนึ่งเคลื่อนที่ได้ นั่นจำเป็นที่จะต้องหาตำแหน่งใหม่ตลอดซึ่งเป็นการสิ้นเปลืองพลังงานเป็นอย่างมาก

[33] ทีมวิจัยได้นำเสนอวิธีการระบุตำแหน่งของ WSN ขั้นตอนวิธีนี้ได้ทำให้สามารถที่จะระบุตำแหน่งของจุดซึ่งไม่มี GPS โดยอาศัยข้อมูลที่ได้จากจุดที่มี GPS ด้วยการส่งผ่านข้อมูลกันระหว่างจุดต่อกันภายในเครือข่าย การระบุตำแหน่งในงานนี้ได้อาศัยการใช้การผสมผสานกันระหว่าง ข้อมูลระยะทางเสมือนที่เกิดจากการส่งข้อมูลระหว่างจุดต่อกัน และการใช้วิธี triangulation ที่ใช้สำหรับการหาตำแหน่งเมื่อมีข้อมูลความแรงของสัญญาณ งานนี้ได้นำเสนอข้อดีคือ การที่ระบบเป็นแบบ distributed นั่นคือไม่จำเป็นที่จะต้องมีหน่วยที่ทำหน้าที่หาตำแหน่งของจุดทุกจุด ถัดมาคือไม่จำเป็นต้องมีการวางโครงสร้างภายในของสภาพแวดล้อมไว้ก่อน ในงานนี้ได้เสนอรูปแบบข้อมูลที่จะใช้ส่งหลายๆแบบ ซึ่งแต่ละแบบจะมีข้อดีข้อเสียที่แตกต่างกันออกไปในแต่ละด้านเช่นด้านความแม่นยำ, ด้านความซับซ้อนของข้อมูลสัญญาณ, ด้านการครอบคลุมพื้นที่และด้านความเหมือนกันของเครือข่าย "DV-hop" และ "DV-distance" ให้ผลลัพธ์ที่น่าพอใจในหลายๆด้าน และยังมีความซับซ้อนของข้อมูลน้อยอีกด้วย ทาง "Euclidean" นั้นจะให้ผลลัพธ์ที่ดีกว่าในกรณีที่คุณสมบัติของเครือข่ายไม่สมมาตรและยังคาดเดาประสิทธิภาพได้มากกว่าอีกด้วย "DV-coordinate" นั้นใช้การสื่อสารที่มากกว่าแบบอื่นแต่ผลลัพธ์ที่ได้ก็ไม่ได้เด่นชัดกว่า แต่ได้ให้รูปแบบใหม่ซึ่งสามารถจะยังนำไปปรับปรุงต่อไป

[34] ได้นำเสนอขั้นตอนวิธีสำหรับการช่วยกันระบุตำแหน่งของ WSN วิธีใหม่ชื่อ MDS-MAP ซึ่งใช้เพียงแค่ความสามารถในการรับรู้ถึงจุดรอบข้าง และยังสามารถนำข้อมูลที่เป็นระยะทางในกรณีที่สามารถวัดได้มาใช้งานร่วมกันได้อีกด้วย จุดเด่นของงานนี้อยู่ที่เมื่อมีจุดอ้างอิงที่รู้ตำแหน่งน้อยๆ ก็ยังสามารถที่จะทำงานได้อยู่ งานนี้ใช้ขั้นตอนวิธีหลักๆจาก MDS เดิม โดยได้มีการเพิ่มเติมขั้นตอนเข้าไปก่อนที่จะนำไปใช้ต่อกับ MDS ขั้นตอนที่เพิ่มเข้ามาคือการหาระยะทางสิ้นสุดของทุกคู่จุดก่อนและจึงนำระยะทางนี้ไปใช้ต่อใน MDS แล้วหลังจากนั้นจึงนำจุดอ้างอิงที่รู้ตำแหน่งอยู่แล้วไปใส่ให้เพื่อที่จะได้ตำแหน่งที่แท้จริงของแต่ละจุด

[35] ได้เสนอวิธีการช่วยกันระบุตำแหน่งบนเครือข่ายตัวรับสัญญาณไร้สายแบบใหม่ชื่อว่า CLOQ โดยอาศัยการวัดระยะทางจากเวลาที่มาถึง (TOA) และใช้ UWB ทางที่ทีมวิจัยได้เสนอการออกแบบการลดทอนของสัญญาณระหว่างทาง และการออกแบบความคลาดเคลื่อนของระยะทางแบบใหม่ และได้นำไปใช้ในงานชิ้นนี้ด้วย งานวิจัยนี้เสนอการนำค่าในบริเวณที่ไม่ได้เกิดจากสัญญาณที่มาทางตรงมาคิดคำนวณด้วย นอกจากนี้ทางที่ทีมวิจัยยังได้เสนอมาตรฐานในการใช้วัดเพิ่มเติมอีกสองแบบนี้คือมาตรฐานของการเชื่อมต่อ ซึ่งใช้ในการบอกว่าการเชื่อมต่อระหว่างจุดสองจุดนั้นมีคุณภาพมากขนาดไหน และอย่างที่สองคือมาตรฐานของการหาตำแหน่งซึ่งบอกถึงคุณภาพของตำแหน่งที่คำนวณออกมาได้ เมื่อทีมวิจัยได้เสนอมาตรฐานทั้งสองอย่างนี้มาแล้วจึงได้มีการทดลองเพื่อทดสอบว่ามาตรฐานไหนที่จะมีผลกระทบต่อคุณภาพของตำแหน่งที่คำนวณออกมาได้มากกว่ากัน และจากการทดลองทำให้เห็นว่ามาตรฐานของการเชื่อมต่อนั้นมีผลต่อคุณภาพของตำแหน่งที่คำนวณได้มากกว่า จึงได้ออกแบบขั้นตอนวิธีที่ใช้ในการช่วยกันระบุตำแหน่งขึ้นมาโดยขึ้นอยู่กับมาตรฐานสองอย่างนี้ ขั้นตอนวิธีที่ออกแบบมานั้นเป็นแบบ distributed ผลลัพธ์ที่ได้จากการจำลองนั้นออกมาค่อนข้างแม่นยำ ทางที่ทีมวิจัยสรุปว่าเมื่อใช้ขั้นตอนวิธีที่คิดคำนวณขึ้นมา และทำการเพิ่มจำนวนจุดอ้างอิงในระบบให้มากขึ้นจะทำให้ความแม่นยำของผลลัพธ์ลดลงแต่หากเพิ่มจำนวนจุดอื่นๆที่ไม่ใช่จุดอ้างอิงให้เพิ่มขึ้นนั้นจะทำให้ความแม่นยำเพิ่มขึ้น ซึ่งในจุดนี้เองเป็นจุดที่แตกต่างกับขั้นตอนวิธีแบบอื่นๆที่เมื่อมีการเพิ่มจำนวนจุดอ้างอิงจะทำให้ผลลัพธ์ดีขึ้น

[36]งานวิจัยชิ้นนี้คล้ายๆกับงาน [35] แต่ในงานนี้อาศัยการส่งผ่านรูปแบบความไม่แน่นอนของตำแหน่งซึ่งเป็นแบบ Gaussian มาใช้งาน และในงานชิ้นนี้ไม่มีการทำแบบวนซ้ำนั่นคือเมื่อจุดใดจุดหนึ่งสามารถหาตำแหน่งได้แล้วจะทำตัวเป็นจุดอ้างอิงในทันที ความไม่แน่นอนของตำแหน่งและความไม่แน่นอนของการวัดระยะทางจะถูกนำไปใช้ในการหาตำแหน่งซึ่งใช้ weighted least square การทดลองในงานวิจัยทำเทียบกับการใช้ Cramer-Rao Lower Bound ในการหาตำแหน่ง เนื่องจากวิธีที่นำเสนอเป็นแบบ distributed ผลที่ได้จึงยังไม่ดีเท่ากับแบบ CRLB แต่ดีกว่าในกรณีที่ไม่มีกรส่งผ่านรูปแบบความไม่แน่นอน

### 3.3 งานด้านอื่นๆ

[37]ได้อธิบายถึง geometry และ algebra ของการ projection จากข้อมูล 1 มิติหลายๆชุด ซึ่งได้จากการวัดข้อมูลในสองมิติงานวิจัยนี้ได้ใช้กล้อง 1 มิติซึ่งปกติจะถูกนำมาใช้ในงานพวก autonomous guided vehicles (AGV) ที่มผู้วิจัยได้เสนอขั้นตอนวิธีเพื่อใช้สำหรับการแก้ปัญหาเมื่อมีข้อมูลเชิงมุม และทราบถึงตำแหน่งของหุ่นยนต์ที่เคลื่อนที่ไป ข้อมูลเชิงมุมแต่ละอันนั้นไม่ได้มีการระบุหมายเลขไว้ทำให้ข้อมูลทุกๆอันมีลักษณะเหมือนกันไม่สามารถแยกความแตกต่างได้ ข้อมูลนี้วัดได้จากการติดแถบสะท้อนแสงอินฟราเรดไว้ยังตำแหน่งต่างๆในสภาพแวดล้อม หุ่นยนต์จะต้องสามารถทำการวัดระยะทางและทิศทางที่เคลื่อนที่ไปได้ด้วย ทางที่มวิจัยได้แสดงให้เห็นถึงความกำกวมที่จะเกิดขึ้นเมื่อตำแหน่งที่วัดค่า หรือ จำนวนจุดมีน้อยกว่าค่าๆหนึ่ง และได้แสดงให้เห็นถึงวิธีการคำนวณคำตอบเมื่อมีจำนวนข้อมูลมากพอและได้มีการอธิบายเพิ่มเติมเกี่ยวกับความกำกวมทั้งหมดไว้ใน[38]

[39]ได้เสนอวิธีการระบุตำแหน่งของกลุ่มตัวส่งสัญญาณหลายๆตัวโดยพร้อมกัน โดยการใช้แถวของตัวรับสัญญาณหลายแถวเพื่อรับค่าความถี่และทิศทางของตัวส่งสัญญาณในกรณีที่ตัวส่งสัญญาณมีหนึ่งตัวนั้น การแก้ปัญหาก็จะเป็นการแก้ตรีโกณธรรมดา แต่เมื่อจำนวนของตัวส่งสัญญาณมากกว่าหนึ่งตัวขึ้นไปจะทำให้เกิดปรากฏการณ์จุดผีขึ้นมาได้จุดผีที่เกิดขึ้นเกิดจากการตัดกันของเส้นตรงของข้อมูลที่แถวรับสัญญาณที่หนึ่งได้รับกับเส้นตรงของแถวสัญญาณจากแถวรับสัญญาณอื่นๆ ในบทความนี้ได้นำเสนอวิธีการแก้ไขปัญหาจุดผีนี้โดยเสนอการใช้แถวรับสัญญาณมากกว่า 3 แถวขึ้นไป ในกรณีที่ไม่มีสัญญาณรบกวนนั้นคำตอบของขั้นตอนวิธีนี้จะทำได้ถูกต้องเสมอ แต่เมื่อมีความคลาดเคลื่อนที่ได้รับจากการวัดค่านั้นจะทำให้ต้องมีการปรับปรุงขั้นตอนเพื่อให้สามารถใช้งานในกรณีที่มสัญญาณรบกวนได้ แต่ที่มที่วิจัยก็ยังไม่สามารถที่จะหาวิธีที่สามารถทำงานได้ทุกความเป็นไปได้ของจุดนั้นคือ ยังมีบางกรณีที่ยังไม่สามารถหาคำตอบได้ด้วยเช่นกัน ขั้นตอนวิธีที่ใช้ในงานวิจัยชิ้นนี้นั้นจำเป็นที่ตัวส่งสัญญาณต้องใช้ความถี่ที่แตกต่างกันออกไปถึงจะทำงานได้อย่างถูกต้อง ถ้ามีจุดที่ความถี่เหมือนกันนั้นจะไม่สามารถใช้งานได้



## บทที่ 4

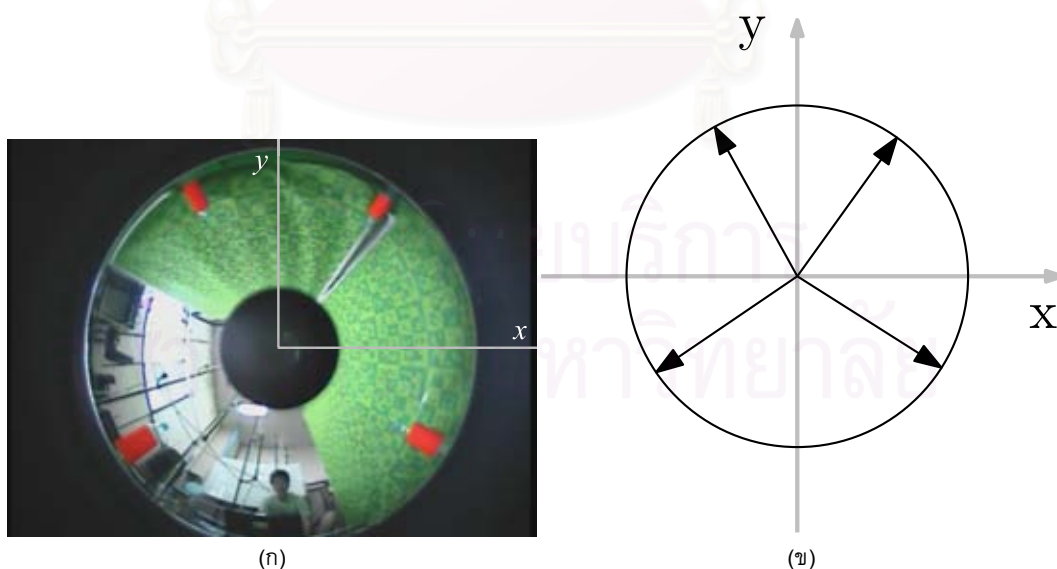
### ปัญหาการร่วมกันระบุตำแหน่งและขั้นตอนวิธีการแก้ปัญหา

เราได้กล่าวถึงที่มาของปัญหาการร่วมกันระบุตำแหน่งและขั้นตอนการทำงานของระบบที่เราสนใจไปพอสมควรแล้วในบทนำในบทนี้เราจะแบ่งเนื้อหาออกเป็น 3 ส่วน ส่วนแรกจะกล่าวถึงขั้นตอนการทำงานของระบบอีกครั้งหนึ่งอย่างละเอียด รวมถึงภาพรวมของปัญหา ส่วนที่สองจะกำหนดสัญลักษณ์ต่างๆที่จะใช้ในบทนี้ และส่วนสุดท้ายจะนำเสนอขั้นตอนวิธีเพื่อใช้ในการแก้ปัญหาสองแบบ

#### 4.1 ภาพรวมของปัญหา

วิทยานิพนธ์ฉบับนี้นำเสนอขั้นตอนวิธีเพื่อใช้ในการระบุตำแหน่งของกลุ่มหุ่นยนต์ โดยที่หุ่นยนต์แต่ละตัวเป็นหุ่นยนต์เคลื่อนที่ได้ ซึ่งถูกติดตั้งด้วยกล้องมองเห็นรอบทิศทาง(Omni-directional camera) เชื่อมทิศ และ แถบสีที่ใช้ช่วยในการหาตำแหน่ง แถบสีสำหรับหุ่นทุกตัวจะมีสีเดียวกันทั้งหมด เนื่องจากการทำให้การแยกแยะแถบสีที่มีสีแตกต่างกันหลายๆแถบสีมีความทันทานสูงยังเป็นเรื่องที่ทำได้ยาก โดยเฉพาะสภาพแวดล้อมที่ไม่ได้มีการจัดเตรียมไว้ล่วงหน้า ดังที่ได้กล่าวไปแล้วในบทนำ

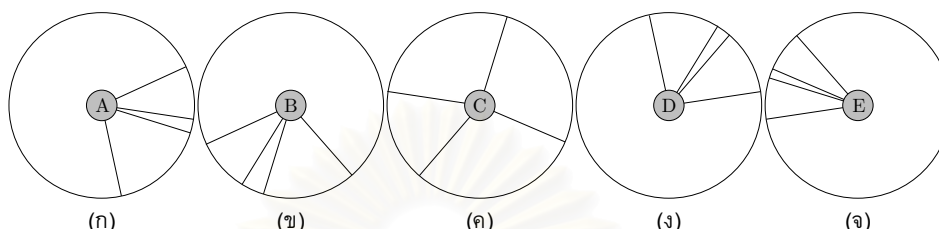
ขั้นตอนการร่วมกันระบุตำแหน่งที่จะนำเสนอ นั้นจะเริ่มจาก หุ่นยนต์แต่ละตัวจะรับภาพจากกล้องและนำมาประมวลผล เพื่อหาตำแหน่งแถบสีของหุ่นยนต์ตัวอื่นๆในภาพ เมื่อได้ตำแหน่งของแถบสีในภาพแล้วจะนำตำแหน่งแถบสีในภาพไปเปลี่ยนเป็นค่ามุมเทียบกับแกนคงที่ในภาพ(เช่นแกน  $x$ ) โดยมีตัวเองเป็นศูนย์กลาง ค่ามุมดังกล่าวนี้จะเรียกแทนด้วย “ข้อมูลเชิงมุม” หรือ “ตำแหน่งเชิงมุม” ตัวอย่างเช่นภาพจากกล้องออมนิจิตรงรูปที่ [4.1ก] ก็จะสามารถคำนวณข้อมูลเชิงมุมได้ดังรูป [4.1ข] และเมื่อนำข้อมูลจาก



รูปที่ 4.1: รูปก. แสดงภาพที่ได้รับจากกล้องพร้อมทั้งแกนอ้างอิงของภาพและ รูปข.แสดงข้อมูลเชิงมุมที่หาได้จากภาพ(แทนด้วยลูกศรชี้ออกจากจุดศูนย์กลางของวงกลม)

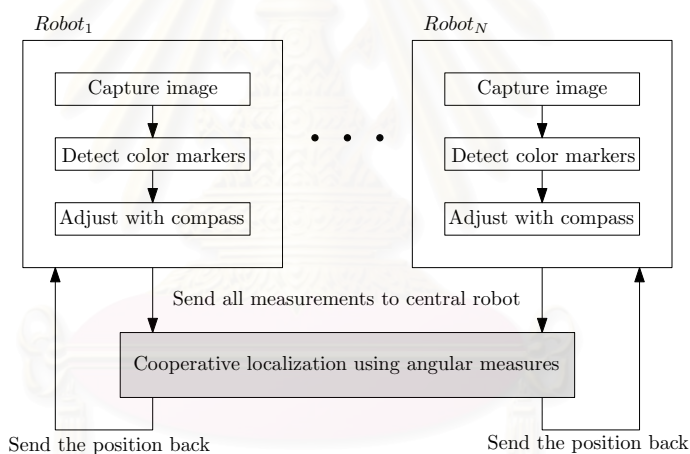
เชื่อมทิศมาปรับ เช่นการหมุนให้แกน  $y$  ของภาพชี้ไปทางทิศเหนือ(นั่นคือมุม 90 องศาของหุ่นทุกตัวก็

จะชี้ไปในทางทิศเหนือด้วย) ก็จะได้ข้อมูลเชิงมุมที่มีแกนอ้างอิงแกนเดียวกันทั้งหมด ยกตัวอย่างระบบที่ประกอบด้วยหุ่นยนต์ทั้งหมด 5 ตัว A,B,C,D และ E สมมติว่าข้อมูลเชิงมุมที่หุ่นยนต์แต่ละตัวหาได้หลังปรับค่าด้วยเข็มทิศเป็นดังแสดงในรูป [4.2] โดยเส้นที่อยู่ในแนวรัศมีของหุ่นยนต์แต่ละตัวแสดงถึงตำแหน่งเชิงมุมของหุ่นยนต์ตัวอื่นๆ สังเกตว่าข้อมูลเชิงมุมทุกๆอันมีลักษณะเหมือนกันเนื่องจากแถบสีที่ใช้เป็นสีเดียวกันทั้งหมด หลังจากที่หุ่นยนต์แต่ละตัวคำนวณหาข้อมูลเชิงมุมและปรับข้อมูลเชิงมุมให้อยู่ในแกน



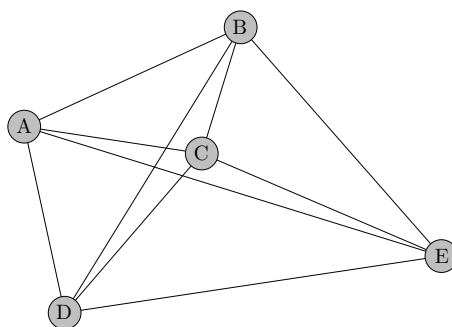
รูปที่ 4.2: แสดงข้อมูลที่หุ่นยนต์แต่ละตัวรับรู้

อ้างอิงเดียวกันเรียบร้อยแล้ว จะทำการส่งข้อมูลเชิงมุมทั้งหมดมายังเซิร์ฟเวอร์ซึ่งทำหน้าที่หาตำแหน่งให้กับหุ่นยนต์ทุกตัวขั้นตอนการทำงานของระบบดังแสดงในรูป[4.3]

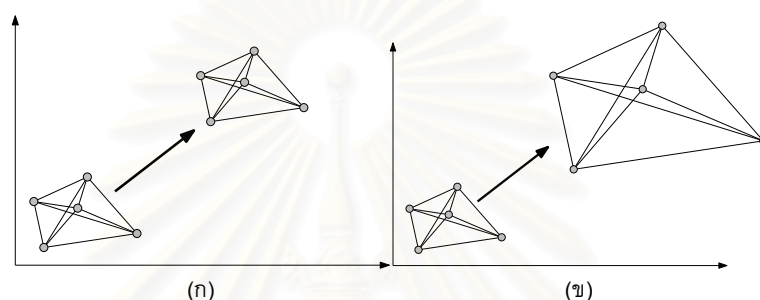


รูปที่ 4.3: แสดงแผนภาพขั้นตอนของระบบ กล้องสีเข้มแสดงถึงส่วนที่งานวิจัยนี้จะนำเสนอ

คำตอบของปัญหานี้คือรูปแบบตำแหน่งของกลุ่มหุ่นยนต์ เช่นในกรณีที่มีข้อมูลเชิงมุมเป็นดังรูป[4.2] เราจะได้คำตอบของปัญหาที่มีลักษณะดังรูป[4.4] สาเหตุที่เรียกคำตอบของปัญหานี้ว่า “รูปแบบตำแหน่งของกลุ่มหุ่นยนต์” ก็เนื่องมาจากจำนวนของคำตอบสำหรับปัญหานี้จะมีได้มากกว่า 1 คำตอบ ยกตัวอย่างเช่น เมื่อเราทำการเคลื่อนย้ายหุ่นยนต์ทุกตัวไปพร้อมๆกันในทิศทางและระยะทางเดียวกันข้อมูลเชิงมุมที่หุ่นยนต์แต่ละตัวได้รับก็จะไม่มีการเปลี่ยนแปลง หรือแม้แต่ในกรณีที่หุ่นยนต์แต่ละตัวเคลื่อนที่ออกจากกันอย่างเป็นอิสระส่วนต่อกันข้อมูลเชิงมุมที่ได้รับก็จะไม่เปลี่ยนแปลงเช่นกัน ดังรูป[4.5] จากเซตของคำตอบทั้งหมดจะเลือกมาเพียงหนึ่งคำตอบเพื่อแทนคำตอบของปัญหานี้และเรียกมันว่า “รูปแบบตำแหน่งของกลุ่มหุ่นยนต์” ในบทที่ 5 จะนำเสนอขั้นตอนวิธีเพื่อใช้ในการหาตำแหน่งที่แท้จริงของหุ่นยนต์แต่ละตัวจากรูปแบบตำแหน่งของกลุ่มหุ่นยนต์และข้อมูลตำแหน่งจริงของหุ่นยนต์บางตัว



รูปที่ 4.4: แสดงตำแหน่งของหุ่นยนต์และเส้นการรับรู้

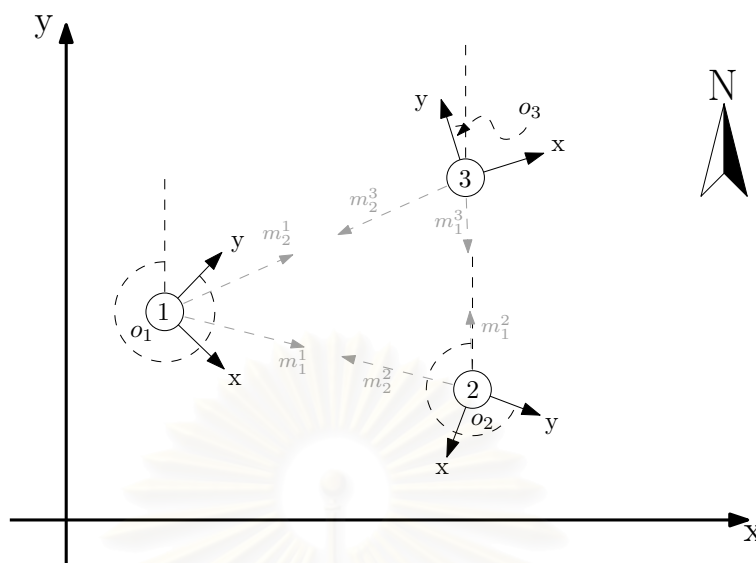


รูปที่ 4.5: รูป ก. แสดงให้เห็นถึงกรณีที่กลุ่มหุ่นยนต์เคลื่อนที่ไปในทิศทางเดียวกันด้วยระยะทางที่เท่ากัน ข้อมูลที่วัดได้ก็จะไม่เปลี่ยนแปลง รูป ข. แสดงในกรณีที่หุ่นยนต์ทุกตัววิ่งห่างออกจากกันด้วยอัตราส่วนเท่าๆกัน ข้อมูลที่วัดได้ก็ไม่เปลี่ยนแปลงเช่นเดียวกัน

## 4.2 ขั้นตอนการแก้ปัญหา

ในส่วนแรกนี้เราจะทำการกำหนดสัญญาณต่างๆที่จะใช้ต่อไปในวิทยานิพนธ์ฉบับนี้กำหนดให้ระบบประกอบด้วยหุ่นยนต์ทั้งหมด  $N$  ตัว ข้อมูลเชิงมุมที่ได้จากภาพของหุ่นยนต์ตัวที่  $i$  ให้แทนด้วยเวกเตอร์  $M_i = (m_1^i, \dots, m_{N-1}^i)$  โดย  $m_j^i \in \mathbb{R}^2, |m_j^i| = 1$  แทนข้อมูลเชิงมุมแต่ละตัวสังเกตว่าหุ่นยนต์แต่ละตัวต้องมีข้อมูลเชิงมุมจำนวน  $N - 1$  ตัวเสมอ เนื่องจากงานวิจัยนี้เสนอวิธีการแก้ปัญหาเฉพาะกรณีที่หุ่นยนต์ทุกตัวสามารถถูกตรวจพบได้ทั้งหมด นั่นคือไม่มีการบังหรือการตรวจไม่พบ ดังที่ได้กล่าวไปแล้วว่าข้อมูลเชิงมุมที่ได้จากภาพจะถูกปรับให้อยู่ในแกนอ้างอิงที่ขนานกันด้วยข้อมูลทิศทางอ้างอิงจากเข็มทิศ กระบวนการปรับมุมนี้ทำได้โดยการหมุนข้อมูลเชิงมุมแต่ละตัวของหุ่นยนต์ตัวที่  $i$  รอบจุดกำเนิดในทิศทางตามเข็มนาฬิกาไปด้วยมุม  $\alpha_i$  โดย  $\alpha_i$  เป็นมุมที่ทิศตะวันออกทำกับแกน  $x$  ของระบบพิกัดของหุ่นยนต์ตัวที่  $i$  ข้อมูลเชิงมุมใหม่ทุกตัวที่ได้จากการหมุนนี้จะเป็นเวกเตอร์ที่เขียนบนพิกัดใหม่ที่มีแกน  $x$  ชี้ไปทางทิศตะวันออก เมื่อกำหนดให้  $A_i = (a_1^i, \dots, a_{N-1}^i)$  โดย  $a_j^i \in \mathbb{R}^D, \|a_j^i\| = 1$  แทนข้อมูลเชิงมุมที่ปรับแล้วของหุ่นยนต์ตัวที่  $i$  เราจะได้ว่าปัญหาที่พิจารณาก็คือการคำนวณรูปแบบตำแหน่งของหุ่นยนต์ในระบบจากข้อมูลเชิงมุม  $A_i, i = 1, \dots, N$  ในวิทยานิพนธ์ฉบับนี้เราขอเสนอวิธีแก้ปัญหาสองวิธีที่มีพื้นฐานแตกต่างกันอย่างสิ้นเชิงวิธีแรกใช้การวิเคราะห์ทางเรขาคณิตเป็นหลัก กระบวนการหารูปแบบตำแหน่งที่ได้นำไปสู่ความเข้าใจปัญหาที่ลึกซึ้งขึ้น ส่วนวิธีที่สองใช้การเปลี่ยนปัญหาให้เป็นปัญหาการหาค่าเหมาะสมที่สุด ถึงแม้วิธีที่นำเสนอจะไม่ได้สะท้อนให้เห็นถึงโครงสร้างปัญหาโดยตรงเหมือนวิธีแรกวิธีที่สองมีความสามารถสูงกว่าวิธีแรกอย่างชัดเจนเรื่องความทนทานต่อความคลาดเคลื่อนจากการวัดข้อมูลเชิงมุม

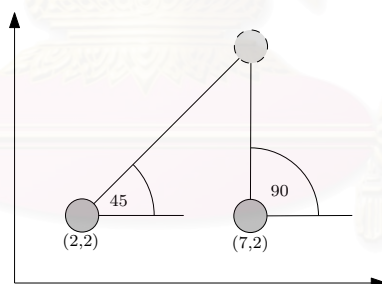
ตัวอย่างระบบพิกัดของกลุ่มหุ่นยนต์ดังรูป [4.6]



รูปที่ 4.6: แสดงพิกัดของระบบ

#### 4.2.1 แนวทางการแก้ปัญหาเชิงเรขาคณิต

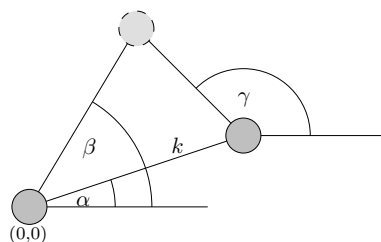
ในกรณีที่มีหุ่นยนต์ทั้งหมดสามตัว และรู้ตำแหน่งของหุ่นยนต์สองตัว จะสามารถหาตำแหน่งของหุ่นยนต์ตัวที่สามได้โดยการอ้างอิงด้วยสามเหลี่ยม (triangulation) เช่นกรณีนี้ที่แสดงในรูป[4.7] หรือใน



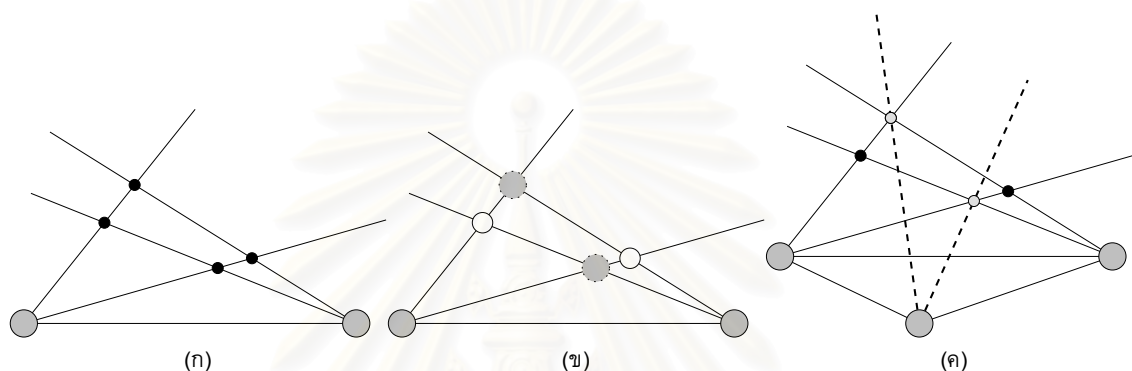
รูปที่ 4.7: แสดงขั้นตอนวิธีการ triangulation โดยที่รู้ตำแหน่งของจุดสองจุด(จุดด้านล่าง) และรู้ตำแหน่งเชิงมุมของจุดที่สามเทียบกับจุดทั้งสองจุดนั้น

กรณีที่ไม่ว่านรู้ตำแหน่งของหุ่นยนต์ทั้งสองตัว แต่รู้ว่าหุ่นยนต์สองตัวทำมุมกันอยู่เท่าไร ก็สามารถหาตำแหน่งของหุ่นยนต์ตัวที่สามได้เช่นกัน ด้วยการกำหนดให้ระยะทางระหว่างคู่หุ่นที่รู้ความสัมพันธ์เป็นค่าคงตัวค่าหนึ่งและกำหนดตำแหน่งให้กับหุ่นยนต์ตัวใดตัวหนึ่งในคู่นั้น ตำแหน่งของหุ่นยนต์ตัวที่สามก็จะสัมพันธ์กับตำแหน่งและค่าคงตัวนั้น ดังตัวอย่างในรูป[4.8]

ปัญหานี้จะมีความซับซ้อนเพิ่มมากขึ้นในกรณีที่กลุ่มของหุ่นยนต์มีขนาดมากกว่า 3 ตัว ดังแสดงในตัวอย่างในรูป[4.9ก] เมื่อนำข้อมูลทั้งหมดจากหุ่นยนต์สองตัวมาหาจุดตัด จะได้จุดตัดขึ้นมาจำนวนหนึ่งในจุดตัดที่ได้มานั้นจะมีเพียงบางส่วนที่เป็นตำแหน่งจริงๆของหุ่นยนต์ ส่วนที่เหลือนั้นจะเป็นจุดที่ไม่ใช่ตำแหน่งจริงๆ เราจะขอเรียกจุดเหล่านี้ว่า "จุดผี" ซึ่งจำนวนของจุดผีมากที่สุดที่จะเกิดขึ้นได้นั้นก็เท่ากับ  $(N - 2) \times (N - 2)$  เมื่อ  $N$  คือจำนวนของหุ่นยนต์ทั้งหมด จะเห็นว่าข้อมูลที่ได้จากหุ่นยนต์สองตัวนั้น



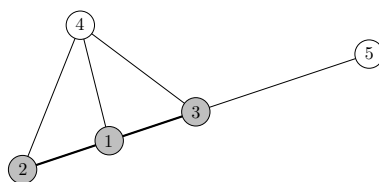
รูปที่ 4.8: แสดงขั้นตอนวิธีการ triangulation โดยรู้มุมที่สัมพันธ์กันระหว่างจุดสองจุด(จุดด้านล่าง 2 จุด) และรู้ตำแหน่งเชิงมุมของจุดที่สามเทียบกับจุดทั้งสองจุดนั้น



รูปที่ 4.9: รูปก. จุดสีดำแสดงจุดตัดที่เกิดขึ้นจากการตัดกันของข้อมูลของหุ่นยนต์สองตัว รูปข. จุดสีเทาแสดงตำแหน่งของหุ่นยนต์จริงๆ ส่วนจุดขาวแสดงถึงจุดผี รูปค. แสดงให้เห็นวิธีการกำจัดจุดผีเมื่อรู้ตำแหน่งของจุดที่สามเพิ่มขึ้นมา เส้นประแสดงถึงข้อมูลของจุดที่สามที่นำมาใช้กำจัดจุดผี

ไม่เพียงพอที่จะกำจัดจุดผีเหล่านี้ออกไปได้ แต่เราสามารถที่จะกำจัดจุดผีนี้ออกไปได้โดยการนำข้อมูลของหุ่นยนต์อีกตัวหนึ่งมาเปรียบเทียบ ซึ่งจำเป็นที่จะต้องรู้ตำแหน่งของหุ่นยนต์ตัวนี้ด้วยโดยเมื่อนำข้อมูลจากหุ่นยนต์ที่เพิ่มขึ้นมานี้มาเปรียบเทียบกับตำแหน่งของจุดตัดทั้งหมด ตำแหน่งของจุดตัดใดที่สอดคล้องกับข้อมูลเชิงมุมของหุ่นยนต์ตัวนี้ก็คือตำแหน่งของหุ่นยนต์จริงๆ ส่วนตำแหน่งของจุดตัดที่ไม่สอดคล้องก็คือจุดผีดังในรูป[4.9ค]

วิธีการที่นำมากำจัดจุดผีนี้ก็เสมือนกับการนำข้อมูลของหุ่นยนต์ทั้งสามตัวมาตัดกันทั้งหมด และดูว่าจุดใดที่เป็นจุดตัดของข้อมูลเชิงมุมจากหุ่นยนต์ 3 ตัว จุดนั้นจึงเป็นตำแหน่งจริงของหุ่นยนต์ เพราะฉะนั้นถ้าหุ่นยนต์สามตัวที่จะนำมาใช้ในการกำจัดจุดผีนั้นอยู่ในแนวเส้นตรงเดียวกันและมีอีกจุดหนึ่งในแนวเส้นตรงนี้แล้วจะทำให้ไม่สามารถหาจุดตัดได้ดังในรูป [4.10] จึงจำเป็นที่จะต้องหาหุ่นยนต์สามตัวที่ไม่อยู่

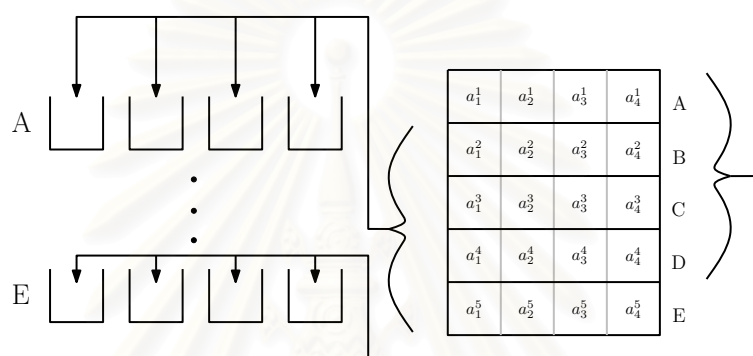


รูปที่ 4.10: แสดงกรณีที่หุ่นยนต์สามตัวที่ใช้กำจัดจุดผี(จุดสีเข้ม)อยู่ในแนวเส้นตรงเดียวกัน แล้วจะทำให้ไม่สามารถหาตำแหน่งของจุดที่ 5 ได้



ในแนวเส้นตรงเดียวกันมาใช้ในการกำจัดจุดฝอกจากวิธีการนี้จึงจำเป็นที่จะต้องรู้ตำแหน่งของหุ่นยนต์อย่างน้อยสามตัวเพื่อใช้ในการกำจัดจุดฝู ซึ่งตำแหน่งของจุดสามจุดนั้นสามารถคำนวณได้จากความสัมพันธ์เชิงมุมระหว่างหุ่นยนต์สามตัวดังที่ได้กล่าวไปแล้วในตอนต้น

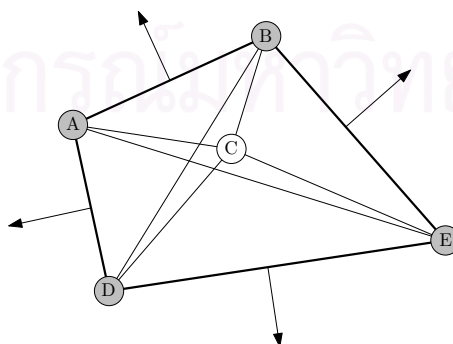
วิธีพื้นฐานที่ใช้ในการหาความสัมพันธ์ระหว่างกลุ่มทุกคู่คือการจับคู่ทุกแบบที่เป็นไปได้แล้วทำการตรวจสอบว่าสามารถจับคู่ได้หรือไม่ยกตัวอย่างเช่น ระบบที่ประกอบด้วยหุ่นยนต์ 5 ตัวดังรูป [4.4] เราจะเริ่มจับคู่จากหุ่นยนต์ A ซึ่งมีข้อมูลเชิงมุมดังรูป[4.2] ข้อมูลของหุ่นยนต์ A มีทั้งหมด 4 ข้อมูล เราจะจับคู่ระหว่างข้อมูลของหุ่นยนต์ A ทุกอันกับข้อมูลของหุ่นยนต์ตัวอื่นๆทุกอัน ก็จะเสมือนกับการเรียงสับเปลี่ยนของข้อมูลหุ่นยนต์ตัวอื่นๆทุกตัวลงในช่อง 4 ช่องที่แทนข้อมูล 4 ข้อมูลของหุ่นยนต์ A และสำหรับหุ่นยนต์ตัว B,C,D และ E ก็เช่นเดียวกันดังรูป [4.11] หลังจากทำการเรียงสับเปลี่ยนครบเรียบร้อยแล้ว



รูปที่ 4.11: แสดงการจับคู่ระหว่างข้อมูลของหุ่นยนต์

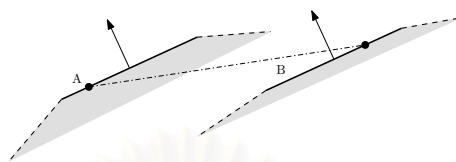
แล้วก็นำข้อมูลที่ใส่ลงช่องของหุ่นยนต์ทุกตัวมาตรวจสอบว่าสามารถจัดเรียงตัวตามแบบนี้ได้หรือไม่ซึ่งวิธีนี้จะสามารถแก้ปัญหาเรื่องความสัมพันธ์ระหว่างหุ่นยนต์ได้แน่นอนแต่ความซับซ้อนเชิงเวลาของขั้นตอนวิธีนี้มากกว่า  $O(N^N)$  ซึ่งทำให้ไม่สามารถนำไปใช้งานได้จริง ต่อไปนี้เราขอเสนอขั้นตอนวิธีในการหาความสัมพันธ์ระหว่างข้อมูลกับหุ่นยนต์ที่ใช้ความซับซ้อนเชิงเวลาน้อยกว่า

หุ่นยนต์สามตัวที่จะนำมาใช้เพื่อกำจัดจุดฝูนั้นได้มาจากการสังเกตรูป[4.12] จะเห็นหุ่นยนต์ที่เป็นกรอบล้อมหุ่นยนต์ตัวอื่นๆ(จุดสีเข้ม) รวมถึงเส้นที่ทำหน้าที่เป็นเส้นขอบ(เส้นเข้ม) นั่นคือเป็นเปลือกหุ้มเชิงคอนเวกซ์ของหุ่นยนต์ทั้งหมด กำหนดให้ทิศทางตั้งฉากของเส้นขอบใดๆคือทิศทางที่ตั้งฉากกับเส้น



รูปที่ 4.12: แสดงจุดและเส้นที่เป็นเปลือกหุ้มเชิงคอนเวกซ์ของกลุ่มหุ่นยนต์ ลูกศรที่ชี้ออกนั้นแสดงถึงทิศทางตั้งฉากของเส้นขอบนั้นๆ

ขอบนั้นๆและมีทิศทางพุ่งออกจากกลุ่มของหุ่นยนต์ตัวอย่างในรูป[4.12] สำหรับเส้นขอบใดๆ จะแบ่งระนาบออกเป็นสองฝั่ง ซึ่งฝั่งที่อยู่ในทิศทางตั้งฉากของเส้นขอบนั้นจะไม่มีหุ่นยนต์ตัวใดๆอยู่เส้นขอบแต่ละเส้นจะมีทิศทางตั้งฉากไม่ซ้ำกับเส้นขอบเส้นอื่นๆยกเว้นในกรณีที่ไม่มีเส้นขอบสองเส้นใดๆที่อยู่ในแนวเส้นตรงเดียวกันซึ่งสามารถพิสูจน์ได้โดยง่าย โดยกำหนดให้มีเส้นขอบสองเส้นที่มีทิศทางตั้งฉากเหมือนกันและไม่อยู่ในแนวเส้นตรงเดียวกัน คือส่วนของเส้นตรง  $A$  และ ส่วนของเส้นตรง  $B$  ดังรูป[4.13] ลากเส้นตรงจาก

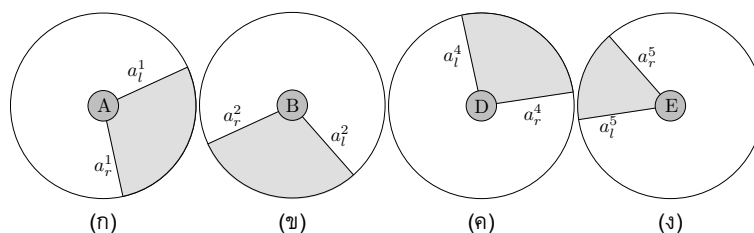


รูปที่ 4.13: แสดงส่วนของเส้นตรง  $A$  และ  $B$  ซึ่งมีทิศทางตั้งฉากเดียวกันแต่ไม่อยู่ในแนวเส้นตรงเดียวกัน

จุดใดๆบน  $A$  ไปยังจุดใดๆ บน  $B$  เส้นตรงที่ลากนี้จะมีโอกาสผ่านได้ 3 แบบคือ ผ่านด้านในกลุ่มหุ่นยนต์ ผ่านด้านนอกกลุ่มหุ่นยนต์ หรือ ผ่านทั้งในและนอกของกลุ่มหุ่นยนต์ ในกรณีสุดท้ายนั้นจะเป็นไปไม่ได้ตามคุณสมบัติของเปลือกหุ้มเชิงคอนเวกซ์ ถ้าผ่านด้านในเพียงอย่างเดียวก็เป็นไปไม่ได้เนื่องจากจะขัดกับทิศทางตั้งฉากและในทำนองเดียวกันถ้าผ่านด้านนอกเพียงอย่างเดียวก็เป็นไปไม่ได้เช่นกัน หมายความว่าถ้ารู้ทิศทางตั้งฉากของเส้นขอบใดๆแล้วจะสามารถหาจุดสองจุดที่ต่ออยู่กับเส้นขอบนั้นได้เสมอ นั่นคือถ้าข้อมูลเชิงมุมใดเป็นเส้นขอบแล้วจะสามารถหาจุดสองจุดที่ต่ออยู่กับข้อมูลนี้ได้เสมอ

จากข้อสังเกตนี้ถ้าเลือกเส้นขอบสองเส้นที่อยู่ติดกันบนเปลือกหุ้มเชิงคอนเวกซ์ ก็จะได้หุ่นยนต์สามตัวและความสัมพันธ์ระหว่างหุ่นยนต์สองเส้น ยกตัวอย่างเช่นในรูป [4.12] ถ้าเลือกเส้นขอบ  $AD$  และ  $DE$  ก็จะสามารถหาจุด  $A, D$  และ  $E$  ได้และความสัมพันธ์สองเส้นนั้นคือ  $AD$  และ  $DE$  นั้นเอง ส่วนความสัมพันธ์อันสุดท้าย(จากตัวอย่างเมื่อครู่ก็คือเส้น  $AE$ ) นั้นสามารถหาได้จากคุณสมบัติโดยทั่วไปของสามเหลี่ยมซึ่งจะอธิบายอย่างละเอียดต่อไป สรุปขั้นตอนโดยสังเขปในการหาตำแหน่งหุ่นยนต์สามตัวคือ เริ่มจากหาเส้นขอบสองเส้นที่ติดกัน ซึ่งหาได้จากการเลือกหุ่นยนต์ที่เป็นขอบขึ้นมาหนึ่งตัวและเลือกเส้นขอบที่ติดอยู่กับหุ่นยนต์นี้ทั้งสองเส้นโดยมีเงื่อนไขคือเส้นขอบสองเส้นนี้จะต้องไม่อยู่ในแนวเดียวกัน เมื่อได้เส้นขอบทั้งสองเส้นแล้วก็จะสามารถหาหุ่นยนต์อีกสองตัวที่เหลือได้ ต่อจากนั้นก็ทำการหาความสัมพันธ์ระหว่างหุ่นยนต์สองตัวที่เหลือนั้น ที่กล่าวมาข้างต้นเป็นเพียงแนวคิดเบื้องต้นเพื่อจะใช้ในการหาตำแหน่งของหุ่นยนต์สามตัว ต่อไปจะอธิบายแต่ละขั้นตอนอย่างละเอียดโดยแบ่งขั้นตอนการหาความสัมพันธ์ของหุ่นยนต์สามตัวนี้ออกเป็น 3 ขั้นตอนแต่ละขั้นตอนเป็นการหาตำแหน่งของหุ่นยนต์ที่ละตัว และมีขั้นตอนสำหรับเตรียมข้อมูล ซึ่งมีไว้เพื่อ หาหุ่นยนต์ตัวที่จะเป็นเปลือกหุ้มเชิงคอนเวกซ์ทั้งหมดและหาเส้นขอบที่สัมพันธ์กับหุ่นยนต์ตัวนั้นๆ

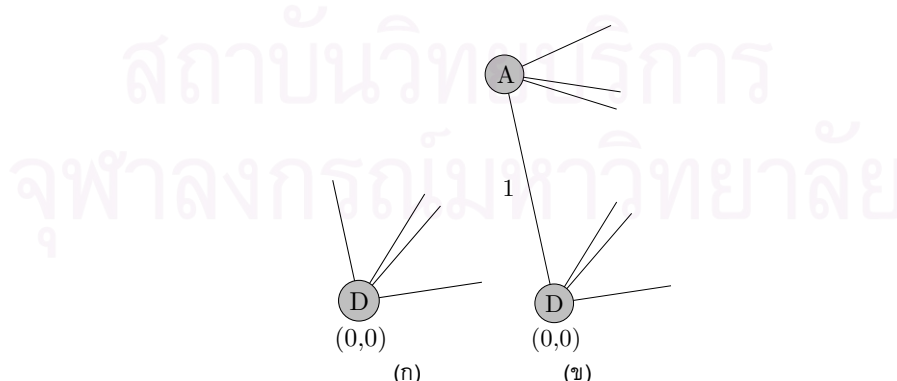
- ขั้นตอนการเตรียมข้อมูล - สำหรับจุด  $i$  ใดๆที่จะเป็นจุดขอบได้นั้นจะต้องมีคุณสมบัติดังต่อไปนี้  $a_i^j \times a_i^k \geq 0$  และสำหรับทุกๆ  $k$  ใดๆ  $a_i^j \times a_i^k \geq 0$  และ  $a_i^k \times a_i^j \geq 0$  กล่าวคือจะมีข้อมูลสองตัวที่ทำมุมกันไม่เกิน 180 องศา และข้อมูลที่เหลือทั้งหมดจะอยู่ระหว่างมุมนี้ จากข้อมูลเชิงมุมของหุ่นยนต์แต่ละตัวในรูป [4.2] จะหาจุดที่เป็นจุดขอบได้ดังรูป[4.14] และกำหนดให้  $a_i^j$  และ  $a_i^k$  เรียกว่ามุมซ้ายสุดและมุมขวาสุด(ก็คือเส้นขอบของเปลือกหุ้มเชิงคอนเวกซ์นั่นเอง)ตามลำดับของแต่ละจุด  $i$  ใดๆที่เป็นจุดขอบ
- ขั้นตอนแรก - เลือกจุดที่ใช้เป็นจุดอ้างอิงตำแหน่ง จากที่กล่าวไปแล้วข้างต้นว่าคำตอบของปัญหานี้



รูปที่ 4.14: แสดงให้เห็นถึงมุมซ้ายสุดและมุมขวาสุดของหุ่นยนต์ A,B,D และ E หุ่นยนต์ C นั้นไม่ได้เป็นส่วนหนึ่งของกรอบเนื่องจากไม่ตรงตามเงื่อนไข ส่วนที่แรเงาแสดงถึงบริเวณที่ข้อมูลเชิงมุมที่เหลืออยู่

มีได้มากกว่าหนึ่งคำตอบซึ่งขึ้นอยู่กับตำแหน่งและอัตราขยาย ในขั้นตอนนี้จะทำการจำกัดจำนวนของคำตอบที่จะเป็นไปได้ลง โดยจะทำการกำหนดพิกัดให้หุ่นยนต์ตัวแรกเช่นกำหนดให้พิกัดเป็น 0,0 ซึ่งจุดที่จะทำการกำหนดตำแหน่งนี้จะเลือกได้จากกลุ่มของจุดที่เป็นขอบทั้งหมด เงื่อนไขที่นำมาใช้เลือกก็คือมุมซ้ายสุดและมุมขวาสุดของจุดนั้นจะต้องไม่อยู่ในแนวเส้นตรงเดียวกัน จากข้อมูลของหุ่นยนต์แต่ละตัว จะสามารถเลือกตัวที่จะนำมาใช้เป็นจุดอ้างอิงตำแหน่งได้ 4 จุดคือ A,B,D และ E ในที่นี้เราจะเลือก D เพื่อให้สะดวกในการอธิบาย [4.15ก]

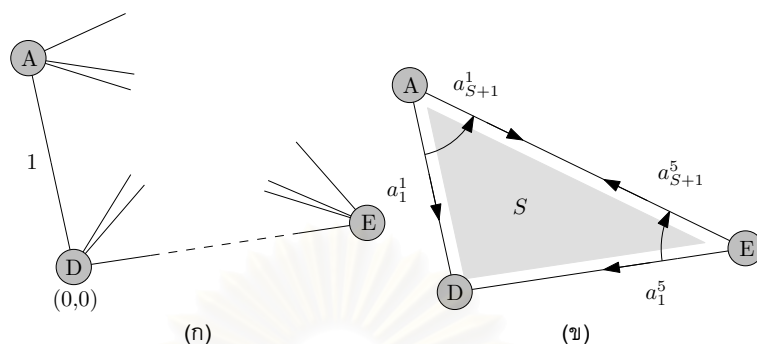
- ขั้นตอนที่สอง - หาจุดที่สัมพันธ์กับจุดอ้างอิงตำแหน่ง ในขั้นนี้เราจะหาตำแหน่งของจุดที่สองซึ่งสัมพันธ์กับจุดแรกและใช้เป็นจุดเพื่อกำหนดอัตราขยาย และจะเรียกจุดนี้ว่าจุดอ้างอิงขนาด โดยจุดที่เราต้องการนั้นจะสัมพันธ์กับมุมซ้ายสุดหรือมุมขวาสุดของจุดอ้างอิงตำแหน่งก็ได้ ในที่นี้เราจะเลือกใช้มุมซ้ายสุด โดยเราจะหาจุดขอบที่มีมุมขวาสุดที่เป็นมุมตรงข้ามกับมุมซ้ายสุดของจุดอ้างอิงตำแหน่งพอดี และเป็นจุดที่ใกล้ที่สุดด้วย(ในกรณีที่มีจุดขอบหลายจุดอยู่ในแนวเดียวกันนั้น) จุดที่ใกล้ที่สุดก็คือจุดที่มีจำนวนข้อมูลที่อยู่ในทิศมุมขวาสุดเพียง 1 ข้อมูล ในที่นี้เราจะได้จุด A เป็นจุดอ้างอิงขนาด หลังจากหาจุดนี้ได้เราจะทำการกำหนดระยะห่างระหว่างสองจุดนี้ให้เป็นค่าคงที่ค่าหนึ่งซึ่งในที่นี้กำหนดให้เท่ากับ 1 หน่วย ก็จะสามารถคำนวณตำแหน่งของจุดนี้จากทิศทางและระยะทางได้และหลังจากขั้นตอนนี้จะทำให้คำตอบของปัญหาการระบุตำแหน่งกลุ่มหุ่นยนต์ลดลงเหลือเพียงหนึ่งคำตอบเท่านั้น[4.15ข]



รูปที่ 4.15: แสดงให้เห็นขั้นตอนที่ 1 และ 2

- ขั้นตอนที่สาม - หาจุดที่สามและความสัมพันธ์ระหว่างทั้งสามจุด จุดที่สามนี้สามารถหาได้ด้วยวิธีการเช่นเดียวกันกับขั้นตอนที่สองแต่เปลี่ยนจากมุมซ้ายสุดของจุดอ้างอิงตำแหน่งเป็นมุมขวาสุด

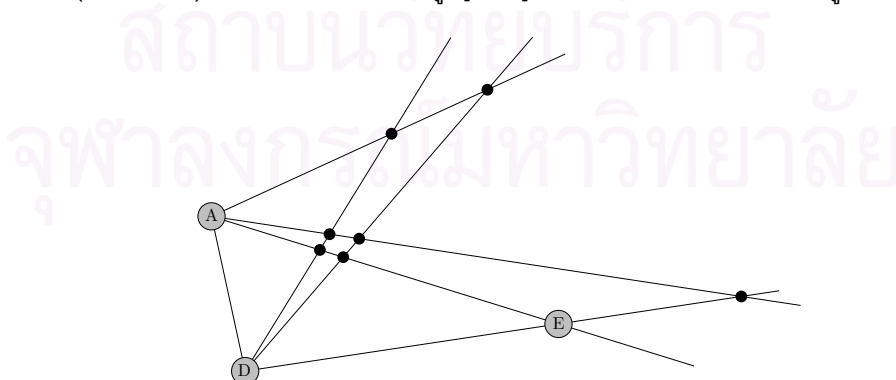
แทน และเมื่อหาความสัมพันธ์ระหว่างจุดอ้างอิงตำแหน่งกับจุดที่สามมาได้แล้วตั้งรูป [4.16ก.] ก็จะต้องหาตำแหน่งของจุดที่สามนี้ ซึ่งจะได้จากความสัมพันธ์ระหว่างจุดที่สามกับจุดที่สอง (A กับ E) เนื่องจากจุดทั้งสามจุดนี้จะวางตัวเป็นสามเหลี่ยมเสมอโดยมีจุดอยู่ใน S จุดโดยที่ S จะมี



รูปที่ 4.16: ในรูปก. จะแสดงให้เห็นหลังจากหาจุดที่สามได้แล้ว แต่ยังไม่รู้ระยะห่างระหว่างจุดที่สามและจุดแรกหรือก็คือยังไม่รู้ความสัมพันธ์ระหว่างจุดที่สองและจุดที่สาม ในรูปข. จะแสดงให้เห็นว่าจุดสามจุดนี้จะวางตัวเป็นรูปสามเหลี่ยมโดยมีจำนวนจุดที่อยู่ด้านในเท่ากับ  $S$  ซึ่งถ้าเราเรียงข้อมูลมุมของจุดที่สองและสาม โดยเริ่มจากจุดแรกตามทิศทวนเข็มนาฬิกาและตามเข็มนาฬิกาตามลำดับแล้ว ข้อมูลที่จะเป็นความสัมพันธ์ระหว่างจุดที่สองและจุดที่สามคือข้อมูลลำดับที่  $S + 1$

ค่าได้ตั้งแต่ 0 ถึง  $N-3$  เพราะฉะนั้นถ้าเราเรียงข้อมูลของจุดที่สองโดยเริ่มจากมุมขวาสุดตามทิศทวนเข็มนาฬิกา ข้อมูลที่จะวัดมาจากจุดที่สามนั้นจะเป็นข้อมูลในตำแหน่งที่  $S+1$  ในทำนองเดียวกันถ้าเรียงข้อมูลของจุดที่สามโดยเริ่มจากมุมซ้ายสุดตามทิศตามเข็มนาฬิกา ข้อมูลที่วัดมาจากจุดที่สองนั้นจะต้องเป็นข้อมูลในตำแหน่งที่  $S+1$  เช่นกัน ดังรูป[4.16ข] จากคุณสมบัตินี้เราจึงทำการเรียงข้อมูลของจุดที่สองและจุดที่สามตามวิธีดังกล่าว แล้วทำการเปรียบเทียบข้อมูลจากอันดับแรกของทั้งสองจุดไปเรื่อยๆจนพบข้อมูลที่มีทิศตรงข้ามกันพอดี ข้อมูลในตำแหน่งนั้นจะเป็นความสัมพันธ์ระหว่างสองจุดนี้ และเมื่อได้ความสัมพันธ์แล้วก็จะสามารถหาตำแหน่งของจุดที่สามได้

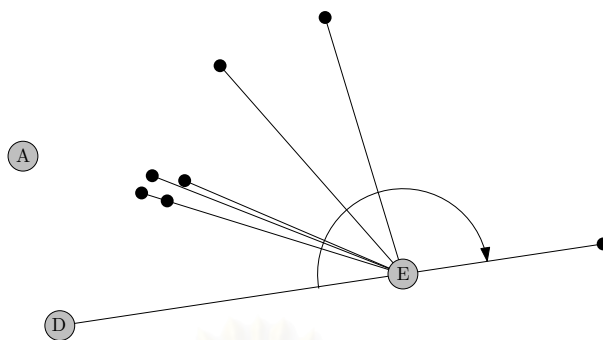
ขั้นตอนถัดไปคือการระบุตำแหน่งของหุ่นยนต์แต่ละตัว โดยเริ่มจากการนำข้อมูลทั้งหมดของจุดแรกและจุดที่สอง (A และ D) มาตัดกันทั้งหมด ดังรูป[4.17] และหลังจากนั้นจึงนำข้อมูลจากจุดที่สาม (E) มา



รูปที่ 4.17: แสดงจุดตัดทั้งหมดที่เกิดขึ้นจากจุดแรกและจุดที่สอง

เทียบ โดยการใช้ข้อมูลที่เรียงแล้วมาเทียบดูว่าจุดตัดไหนเป็นตำแหน่งจริงและจุดตัดไหนเป็นจุดผี ก่อนที่จะนำจุดตัดทั้งหมดมาเทียบนั้นเราจะนำจุดตัดนั้นมาเรียงตามมุมกับจุดสามโดยเริ่มจากมุมซ้ายสุดตาม

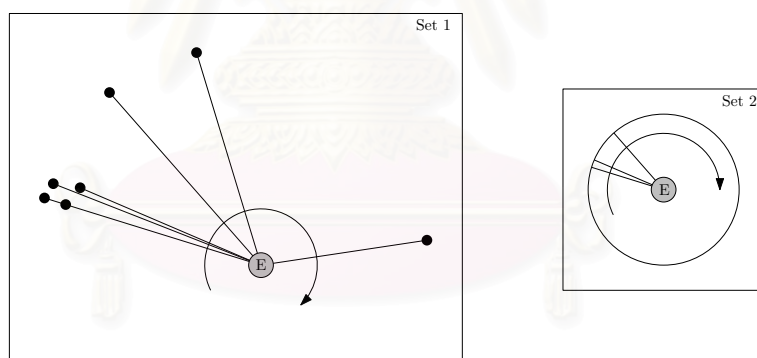
เข็มนาฬิกาเช่นกัน ดังรูป[4.18] หลังจากนั้นจะทำการเทียบมุมระหว่างสองชุดข้อมูลคือ มุมที่ได้จากจุดตัด



รูปที่ 4.18: แสดงจุดตัดที่นำมาเรียงมุมเทียบกับจุดสามตามเข็มนาฬิกาโดยเริ่มตั้งแต่มุมซ้ายสุดของจุดสาม

ทั้งหมดกับมุมที่ได้จากการวัด รูป [4.19] แสดงข้อมูลสองชุดที่จะนำมาเทียบกันโดยผลจากการเทียบมุมจะแบ่งออกได้เป็น 3 กรณี

1. ถ้ามุมใดเท่ากันก็แสดงว่าจุดตัดนั้นเป็นตำแหน่งจริงๆของหุ่นยนต์
2. ถ้ามุมใดที่มีในชุดข้อมูลจุดตัดแต่ไม่มีในชุดข้อมูลการวัดก็แสดงว่าจุดนั้นเป็นจุดผี
3. ถ้ามุมใดมีในชุดข้อมูลการวัดแต่ไม่มีในชุดข้อมูลจุดตัดก็แสดงว่าจะมีจุดจริงๆที่อยู่บนเส้นตรง  $\overline{DA}$

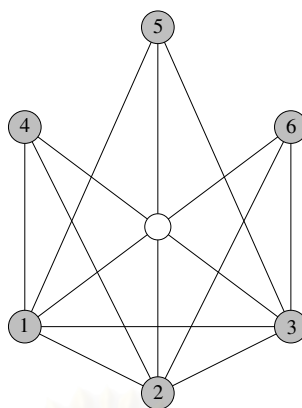


รูปที่ 4.19: แสดงชุดข้อมูลสองชุดที่จะนำมาเทียบกัน โดยมุมที่แสดงถึงความสัมพันธ์ระหว่างจุดสามและจุดแรกจะไม่นำมาคิด

จากขั้นตอนวิธีทั้งหมดจะสามารถหาความซับซ้อนเชิงเวลาได้จากแต่ละขั้นตอนย่อย โดยขั้นตอนเตรียมข้อมูลนั้นมีความซับซ้อนเชิงเวลาเป็น  $O(N)$  เช่นเดียวกับขั้นตอนที่ 1 และ 2 ด้วย ส่วนขั้นตอนที่สามนั้นจำเป็นที่จะต้องมีการเรียงข้อมูลก่อนจึงมีความซับซ้อนเชิงเวลาเป็น  $O(N \log(N))$  ขั้นตอนสุดท้ายที่กำจัดจุดผีออกนั้นใช้เวลามากที่สุดคือ  $O(N^2 \log(N))$  เนื่องจากมีการเรียงของข้อมูลจุดตัดทั้งหมด เพราะฉะนั้นเราจึงได้ความซับซ้อนเชิงเวลาของทั้งระบบเป็น  $O(N^2 \log(N))$

จาก ขั้นตอนวิธี ข้างต้น นั้น จะ สามารถ ระบุ ตำแหน่ง ของ กลุ่ม หุ่นยนต์ ได้ โดย อาศัย เพียงแค่ ข้อมูลเชิงมุมเท่านั้น แต่ขั้นตอนวิธีนี้ยังมีข้อจำกัดอยู่คือในกรณีที่มีจุดผีที่เกิดจากจุดตัดของข้อมูลของหุ่นยนต์สามตัวดังรูป[4.20] อาจจะทำให้ไม่สามารถจะหาคำตอบได้หรือคำตอบที่หามาได้นั้นไม่ถูกต้อง เช่น ถ้าเราเลือกจุดที่สองมาเป็นจุดอ้างอิงตำแหน่ง แล้วจุดสีขาวที่ไม่ใช่จุดตำแหน่งจริงจะกลายเป็นคำตอบของ





รูปที่ 4.20: แสดงกรณีที่สามารถทำให้การหาตำแหน่งด้วยวิธีนี้ผิดพลาดได้ จุดสีเทาแสดงตำแหน่งที่เกิดจากการตัดกันของเส้นสามเส้น จุดสีเทาแสดงตำแหน่งจริงๆของหุ่นยนต์

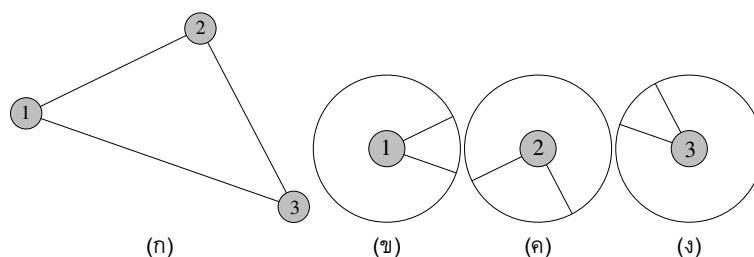
ขั้นตอนวิธี ซึ่งเป็นคำตอบที่ไม่ถูกต้องในบทที่ 6 จะนำเสนอการทดลองสำหรับขั้นตอนวิธีนี้และแสดงผลการทดลอง

ในสภาพแวดล้อมจริง ข้อมูลเชิงมุมที่วัดได้จากกล้องจะมีความคลาดเคลื่อนอยู่ด้วยเสมอ ทำให้ขั้นตอนวิธีแรกที่น่าเสนอไปไม่สามารถทำงานได้อย่างถูกต้องเนื่องจากขั้นตอนที่ใช้ในการตรวจสอบความสอดคล้องกันระหว่างคู่ข้อมูลเชิงมุมจะใช้วิธีการตรวจสอบว่ามุมสองมุมทำมุมกันอยู่ 180 องศา ซึ่งในกรณีที่มีความคลาดเคลื่อนมุมสองมุมที่สอดคล้องกันไม่จำเป็นที่จะต้องทำมุมกัน 180 องศาพอดี ในกรณีนี้ จะสามารถที่จะเปลี่ยนเงื่อนไขการตรวจสอบความสอดคล้องกันระหว่างคู่ข้อมูลเชิงมุมให้เป็นช่วงแทนได้ แต่ความซับซ้อนเชิงเวลาก็จะเปลี่ยนเป็นแบบไม่เชิงพหุนาม วิทยานิพนธ์ฉบับนี้จึงได้เสนอขั้นตอนวิธีอีกขั้นตอนหนึ่งซึ่งสามารถที่จะทนต่อความผิดพลาดในการวัดได้และความซับซ้อนเชิงเวลาอยู่ในรูปเชิงพหุนาม และจะอธิบายในส่วนถัดไป

#### 4.2.2 แนวทางการแก้ปัญหาด้วยวิธีการหาค่าเหมาะที่สุดเชิงคอนเวกซ์

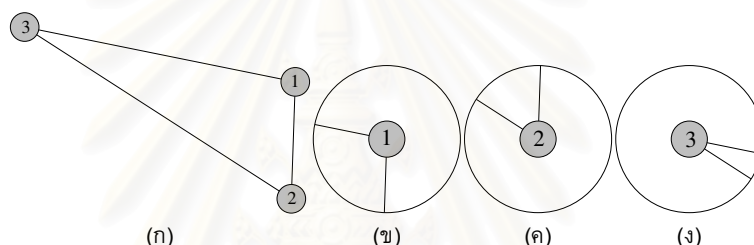
ขั้นตอนวิธีแรกที่น่าเสนอไปอาศัยคุณสมบัติทางเรขาคณิตของสามเหลี่ยมในการหาความสัมพันธ์ระหว่างข้อมูลและใช้เพื่อคำนวณพิกัดของหุ่นยนต์แต่ละตัว แต่สำหรับขั้นตอนวิธีที่จะนำเสนอต่อไปนี้นั้น จะอาศัยการเปลี่ยน “ปัญหาการร่วมกันระบุตำแหน่งโดยใช้ข้อมูลเชิงมุม” ให้กลายเป็น “ปัญหาการหาค่าเหมาะที่สุดเชิงคอนเวกซ์” ซึ่งเป็นปัญหาที่มีขั้นตอนวิธีในการแก้ปัญหาอยู่แล้วเป็นจำนวนมาก [5] และความซับซ้อนเชิงเวลาของการแก้ปัญหาอยู่ในรูปพหุนามของจำนวนข้อมูล และยิ่งไปกว่านั้นจากการเปลี่ยนปัญหานี้เองทำให้สามารถหาคำตอบได้แม้ในกรณีที่มีความคลาดเคลื่อนในการวัดการนำเสนอสำหรับขั้นตอนวิธีนี้จะแบ่งออกได้เป็น 2 ช่วง ในช่วงแรกเราจะอธิบายถึง “ขั้นตอนวิธีแบบทำซ้ำ” ที่ได้คิดขึ้นมา และผลการทดลองขั้นตอนวิธีนี้แสดงให้เห็นถึงความเป็นไปได้ของการแก้ปัญหา และในช่วงที่สองเราจะทำการวิเคราะห์ “ขั้นตอนวิธีแบบทำซ้ำ” ที่ได้คิดขึ้นมาซึ่งจะนำไปสู่การเปลี่ยนรูปแบบของปัญหา

ต่อไปจะนำเสนอตัวอย่างขั้นตอนการทำงานของ “ขั้นตอนวิธีแบบทำซ้ำ” กำหนดให้ระบบหุ่นยนต์ตัวอย่างประกอบด้วยหุ่นยนต์ทั้งหมด 3 ตัวซึ่งมีตำแหน่งดังรูป [4.21ก] และข้อมูลเชิงมุมที่ได้จากหุ่นยนต์แต่ละตัวดังรูป [4.21ข-ง] ขั้นตอนการทำงานเป็นดังนี้



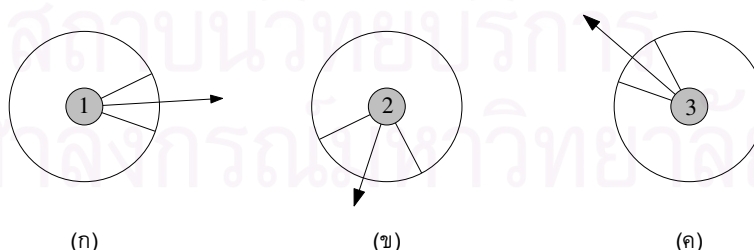
รูปที่ 4.21: ในรูปก. แสดงตำแหน่งของหุ่นยนต์สามตัว และ ข-ง แสดงข้อมูลที่หุ่นยนต์แต่ละตัวรับได้

1. สุ่ม ตำแหน่ง หุ่นยนต์ ทุกตัว ใน “กลุ่มหุ่นยนต์ทำซ้ำ” ซึ่งกลุ่มหุ่นยนต์นี้จะประกอบด้วยหุ่นยนต์จำนวนเท่ากับกลุ่มหุ่นยนต์ที่เราต้องการหาตำแหน่ง และเมื่อเสร็จสิ้นขั้นตอนการทำงานตำแหน่งของหุ่นยนต์แต่ละตัวใน “กลุ่มหุ่นยนต์ทำซ้ำ” จะเป็นคำตอบของปัญหา นี้ ฉะนั้นจึงได้ว่าตำแหน่งของหุ่นยนต์ตัวที่  $i$  ใน “กลุ่มหุ่นยนต์ทำซ้ำ” ก็จะสอดคล้องกับตำแหน่งของหุ่นยนต์ตัวที่  $i$  ในกลุ่มหุ่นยนต์ที่เราสนใจ รูป [4.22ก] แสดงให้เห็นถึงตำแหน่งของ “กลุ่มหุ่นยนต์ทำซ้ำ” หลังจากสุ่มตำแหน่ง



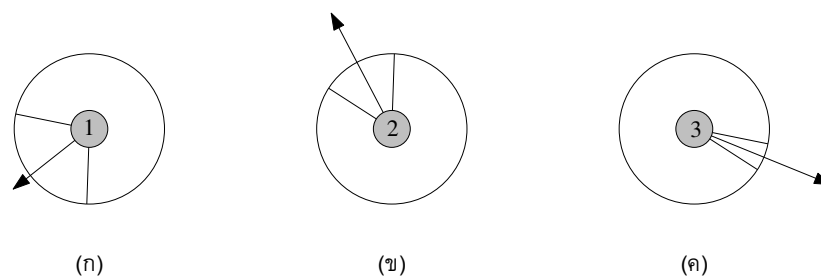
รูปที่ 4.22: ในรูปก. แสดงตำแหน่งของหุ่นยนต์สามตัวเมื่อทำการเปลี่ยนตำแหน่งแล้ว และ ข-ง แสดงข้อมูลที่หุ่นยนต์แต่ละตัวรับได้

2. หา “เวกเตอร์รวมข้อมูลเชิงมุม” ของหุ่นยนต์แต่ละตัวในกลุ่มหุ่นยนต์ที่เราสนใจ โดยเวกเตอร์นี้เกิดจากการนำข้อมูลเชิงมุมทุกอันมาบวกกัน รูป [4.23] แสดงเวกเตอร์รวมข้อมูลเชิงมุมสำหรับหุ่นยนต์แต่ละตัว



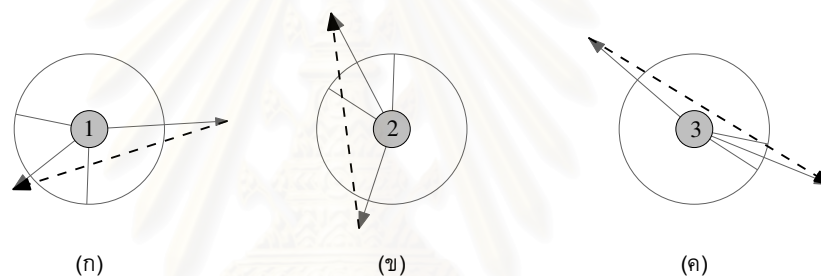
รูปที่ 4.23: แสดงเวกเตอร์รวมข้อมูลเชิงมุมของหุ่นยนต์แต่ละตัวในกลุ่มหุ่นยนต์ที่เราสนใจ โดยเส้นที่เป็นลูกศรจะแสดงเวกเตอร์รวมข้อมูลเชิงมุม

3. หาข้อมูลเชิงมุมของหุ่นยนต์แต่ละตัวที่อยู่ใน “กลุ่มหุ่นยนต์ทำซ้ำ” ข้อมูลเชิงมุมของหุ่นยนต์แต่ละตัวจะเป็นดังรูป [4.22ข-ง]
4. หาเวกเตอร์รวมข้อมูลเชิงมุมของหุ่นยนต์แต่ละตัวใน “กลุ่มหุ่นยนต์ทำซ้ำ” ซึ่งจะได้ดังรูป [4.24]



รูปที่ 4.24: แสดงเวกเตอร์รวมข้อมูลเชิงมุมของหุ่นยนต์แต่ละตัวใน “กลุ่มหุ่นยนต์ทำซ้ำ”

5. คำนวณหาเวกเตอร์การเคลื่อนที่สำหรับหุ่นยนต์แต่ละตัว ซึ่งเวกเตอร์นี้จะเกิดจากการนำเวกเตอร์รวมข้อมูลเชิงมุมของ “กลุ่มหุ่นยนต์ทำซ้ำ” ลบด้วยเวกเตอร์รวมข้อมูลเชิงมุมของกลุ่มหุ่นยนต์ที่เราสนใจ จะได้เวกเตอร์การเคลื่อนที่ดังรูป [4.25]



รูปที่ 4.25: เส้นประแต่ละเส้นแสดงเวกเตอร์การเคลื่อนที่สำหรับหุ่นยนต์แต่ละตัว

6. คำนวณตำแหน่งใหม่ให้กับหุ่นยนต์แต่ละตัวใน “กลุ่มหุ่นยนต์ทำซ้ำ” โดยการบวกตำแหน่งของหุ่นยนต์แต่ละตัวกับเวกเตอร์การเคลื่อนที่ที่สอดคล้องกันสำหรับหุ่นยนต์แต่ละตัว
7. หาข้อมูลเชิงมุมของหุ่นยนต์แต่ละตัวใน “กลุ่มหุ่นยนต์ทำซ้ำ” ใหม่และทำการเปรียบเทียบกับข้อมูลเชิงมุมของกลุ่มหุ่นยนต์ที่เราสนใจ ถ้ามีความคล้ายคลึงกันมากก็จะหมายความว่าคำตอบที่ได้นั้นถูกต้องแล้ว แต่ถ้ายังไม่คล้ายคลึงกันก็กลับไปทำขั้นตอนที่ 4-6 ใหม่

เราสามารถเขียนขั้นตอนการทำงานข้างต้นให้อยู่ในรูป pseudocode ได้ดัง pseudocode 1

จากการที่เราไม่ทราบถึงความสมนัยระหว่างข้อมูลเชิงมุมกับตัวหุ่นยนต์ ทำให้การตรวจสอบเงื่อนไขการหยุดการทำงานของ “ขั้นตอนวิธีแบบทำซ้ำ” (ที่จะหยุดเมื่อข้อมูลเชิงมุมมีความคล้ายคลึงกันมาก) ไม่สามารถที่จะทำได้ภายในเวลาอันสั้น ในวิทยานิพนธ์นี้เราจึงได้เสนอเงื่อนไขการหยุดที่สามารถทำงานได้รวดเร็วซึ่งเป็นการประมาณการหยุดของเงื่อนไขการหยุดที่แท้จริงเงื่อนไขหยุดการทำงานที่เสนอใน pseudocode นี้คือถ้าผลต่างระหว่างอัตราส่วนระหว่างระยะทางเคลื่อนที่มากที่สุดกับระยะทางเคลื่อนที่มากที่สุดอันดับสองกับอัตราส่วนนี้ของการเคลื่อนที่ครั้งที่แล้ว น้อยกว่าค่าคงตัวค่าหนึ่ง ก็จะหยุดทำงานถ้ามากกว่าก็จะทำงานต่อ

---

**Algorithm 1** Iterative method
 

---

```

1: for  $i = 1$  to  $N$  do
2:    $\{N$  is total number of robot $\}$ 
3:    $RobotPos[i] \leftarrow RandomPoint2D(640, 480)$   $\{\text{Random position for all robots}\}$ 
4:    $MovedPos[i] \leftarrow RandomPoint2D(640, 480)$   $\{\text{Random moved position for all robots}\}$ 
5: end for
6: for  $i = 1$  to  $N$  do
7:    $k \leftarrow 0$ 
8:   for  $j = 1$  to  $N$  do
9:     if  $i \neq j$  then
10:       $RobotMeasurement[i][k] \leftarrow normalize(RobotPos[j] - RobotPos[i])$   $\{\text{Compute angular measurement for all robots}\}$ 
11:       $RobotMeasurement2[i][k] \leftarrow normalize(MovedPos[j] - MovedPos[i])$   $\{\text{Compute angular measurement for all moved robots}\}$ 
12:       $k \leftarrow k + 1$ 
13:     end if
14:   end for
15: end for
16: for  $i = 1$  to  $N$  do
17:    $RobotSumVect[i] \leftarrow zero - vector$   $\{\text{Initialize summation vector}\}$ 
18:   for  $j = 1$  to  $N - 1$  do
19:      $RobotSumVect[i] = RobotSumVect[i] + RobotMeasurement[i][j]$   $\{\text{Sum all measurement vectors}\}$ 
20:   end for
21: end for
22:  $MovingThreshold1 \leftarrow -1$ 
23: repeat
24:    $MovingThreshold2 \leftarrow MovingThreshold1$ 
25:    $MaxMovingDist1 \leftarrow -1$ 
26:    $MaxMovingDist2 \leftarrow -1$ 
27:   for  $i = 1$  to  $N$  do
28:      $RobotSumVect2[i] \leftarrow ZeroVector$   $\{\text{Initialize summation vector}\}$ 
29:     for  $j = 1$  to  $N - 1$  do
30:        $RobotSumVect2[i] = RobotSumVect2[i] + RobotMeasurement2[i][j]$   $\{\text{Sum all measurement vectors}\}$ 
31:     end for
32:      $DifVect[i] \leftarrow RobotSumVect2[i] - RobotSumVect[i]$   $\{\text{Calculate different vector}\}$ 
33:     if  $MaxMovingDist1 < |DifVect[i]|$  then
34:        $MaxMovingDist2 \leftarrow MaxMovingDist1$   $\{\text{Find 1st and 2nd max distance}\}$ 
35:        $MaxMovingDist1 \leftarrow |DifVect[i]|$ 
36:     end if
37:      $MovedPos[i] \leftarrow MovedPos[i] + DifVect[i]$   $\{\text{Move robot with } DifVect[i]\}$ 
38:   end for
39:   for  $i = 1$  to  $N$  do
40:      $k \leftarrow 0$ 
41:     for  $j = 1$  to  $N$  do
42:       if  $i \neq j$  then
43:          $RobotMeasurement2[i][k] \leftarrow normalize(MovedPos[j] - MovedPos[i])$ 
44:          $k \leftarrow k + 1$ 
45:       end if
46:     end for
47:   end for
48:    $MovingThreshold1 \leftarrow \frac{MaxMovingDist2}{MaxMovingDist1}$ 
49: until  $|MovingThreshold1 - MovingThreshold2| < THRESHOLD$ 

```

---

เมื่อนำขั้นตอนวิธีนี้ไปทำการทดลองใน simulator ผลที่ได้จากการทดลองแสดงให้เห็นว่าหุ่นยนต์แต่ละคู่ที่สอดคล้องกันระหว่าง “กลุ่มหุ่นยนต์ทำซ้ำ” และ “กลุ่มหุ่นยนต์ที่สนใจ” มีลักษณะข้อมูลเชิงมุม

คล้ายกันอย่างมาก การทดลองอย่างละเอียดสามารถติดตามได้ในบทที่ 6 ในส่วนหลังจากนี้จะทำการวิเคราะห์ขั้นตอนวิธีการนี้โดยการใช้สมการทางคณิตศาสตร์เพื่อพิสูจน์ความถูกต้องของขั้นตอนวิธี

กำหนดให้  $N$  คือจำนวนของหุ่นยนต์ทั้งหมดและตำแหน่งของหุ่นยนต์แต่ละตัว  $R$

$$R = \{r_1, r_2, \dots, r_N \mid r_i \in \mathbb{R}^2\} \quad (4.1)$$

ข้อมูลจริง ๆ ที่ควรจะได้จากตำแหน่งของหุ่นยนต์แต่ละตัว  $D$

$$D = \{(d_{12}, d_{13}, \dots, d_{1N}), (d_{21}, d_{23}, \dots, d_{2N}), \dots, (d_{N1}, d_{N2}, \dots, d_{N(N-1)}) \mid d_{ij} = \frac{r_j - r_i}{|r_j - r_i|}\} \quad (4.2)$$

ข้อมูลที่ได้จากการวัดจริง ๆ

$$V = \{(v_{12}, v_{13}, \dots, v_{1N}), (v_{21}, v_{23}, \dots, v_{2N}), \dots, (v_{N1}, v_{N2}, \dots, v_{N(N-1)}) \mid v_{ij} = d_{ij} + e_{ij}\} \quad (4.3)$$

$$e_{ij} = (d_{ij} \cdot ((\cos(\delta_{ij}), -\sin(\delta_{ij})), (\sin(\delta_{ij}), \cos(\delta_{ij}))) \cdot d_{ij}; \delta_{ij} \sim \mathbf{N}(0, \sigma^2) \quad (4.4)$$

ตำแหน่งของหุ่นยนต์ในแต่ละขั้นของวิธีที่นำเสนอ

$$X_t = ((x_1^t, y_1^t), (x_2^t, y_2^t), \dots, (x_N^t, y_N^t)) \quad (4.5)$$

$$X_t = (p_1^t, p_2^t, \dots, p_N^t \mid p_i^t = (x_i^t, y_i^t)) \quad (4.6)$$

สมการของการอัปเดตตำแหน่งขั้นถัดไป

$$X_{t+1} = X_t + f(X_t) \quad (4.7)$$

กำหนดให้เวกเตอร์รวมของข้อมูลของหุ่นยนต์ตัวที่  $i$  คือ

$$s_i = \sum_{j=1 \mid j \neq i}^N v_{ij} \quad (4.8)$$

จะได้ว่า

$$f(X_t) = (\delta_1^t, \delta_2^t, \dots, \delta_N^t \mid \delta_i^t = \sum_{j=1 \mid j \neq i}^N \frac{p_j^t - p_i^t}{|p_j^t - p_i^t|} - s_i) \quad (4.9)$$

และจากผลการทดลองตาม pseudocode 1 จะเห็นว่าคู่หุ่นยนต์ที่สอดคล้องกันระหว่าง “กลุ่มหุ่นยนต์ทำซ้ำ” และ “กลุ่มหุ่นยนต์ที่สนใจ” จะมีข้อมูลเชิงมุมที่คล้ายคลึงกันมากเสมอ นอกจากนี้จะเห็นว่าสมการ (4.7) มีลักษณะคล้ายกับขั้นตอนวิธีการหาค่าเหมาะที่สุดชนิดหนึ่งคือขั้นตอนการไล่ทางลาด (Gradient descent)[40] ซึ่งจะมีรูปแบบทั่วไปดังนี้ กำหนดให้  $G(X) : \mathbb{R}^n \rightarrow \mathbb{R}$  และสามารถที่จะหาอนุพันธ์ของ  $G(X)$  รอบๆจุด  $a$  ใดๆได้ สิ่งที่เราต้องการที่จะหาคือจุดที่ทำให้  $G(X)$  มีค่าน้อยที่สุด จะได้สมการเพื่อใช้สำหรับปรับดังนี้

$$a_{n+1} = a_n - \gamma_n \nabla G(a_n), n \geq 0 \quad (4.10)$$



ซึ่งค่า  $\gamma_i > 0, i = 1 \dots n$  และมีค่าน้อยๆ เมื่อทำการเปรียบเทียบสมการ (4.10) กับ (4.7) จะได้ว่า  $-f(X_t) \equiv \nabla G(x_n)$  และ  $\gamma_i = 1, i = 1 \dots n$  หลังจากนั้นจึงทำการหา  $F(X)$  ที่ทำให้  $\frac{\partial}{\partial p_i^t} F(X_t) = -f_i(X_t) = -\delta_i^t, i = 1 \dots n$  จะได้ว่า

$$F_i(X_t) = - \int f_i(X_t) dp_i = - \int \left( \sum_{j=1|j \neq i}^N \frac{p_j^t - p_i^t}{|p_j^t - p_i^t|} - s_i \right) dp_i \quad | \quad i = 1 \dots N \quad (4.11)$$

กำหนดให้  $\psi(p_i)$  คือฟังก์ชันใดๆที่ขึ้นอยู่กับ  $p_i$  เพียงอย่างเดียว จะได้ว่า

$$- \int f_i(X_t) dp_i = - \sum_{j=1|j \neq i}^N (|p_j - p_i| + \psi(p_j)) - s_i p_i + c_i \quad (4.12)$$

$$\int f_i(X_t) dp_i = \sum_{j=1|j \neq i}^N (|p_j - p_i| + \psi(p_j)) + s_i p_i - c_i \quad (4.13)$$

เมื่อได้  $F_i(X_t)$  ทั้งหมดมาแล้วเราสามารถหา  $F(X_t)$  ได้โดยนำ  $F_i(X_t)$  ทั้งหมดมาคูณกันและพจน์ที่เกี่ยวข้องกับฟังก์ชัน  $\psi(p_i)$  จะถูกกำจัดออกไปเนื่องจากพจน์นี้นั้นจะประกอบอยู่ใน  $F_i(X_t)$  ตัวใดตัวหนึ่งอยู่แล้วฉะนั้นจะได้

$$F(X_t) = \sum_{i=1}^N \sum_{j=i+1}^N |p_j^t - p_i^t| + \sum_{i=1}^N s_i \cdot p_i^t \quad (4.14)$$

และจากผลการทดลองจะเห็นว่าฟังก์ชันนี้จะลู่เข้าสู่คำตอบที่ถูกต้องได้เสมอ จึงมีการตั้งสมมติฐานว่าฟังก์ชันนี้น่าจะเป็นคอนเวกซ์ คอนเวกซ์ฟังก์ชันสามารถที่จะหาค่าเหมาะสมที่สุดได้ในเวลาเชิงพหุนามด้วยวิธีที่มีประสิทธิภาพ ยิ่งไปกว่านั้นการหาค่าเหมาะสมที่สุดสามารถทำได้โดยใช้วิธี gradient descent และจะลู่เข้าสู่คำตอบที่ถูกต้องได้เสมอ ยกตัวอย่างเช่นการขยับหุ่นยนต์ในรูปแบบที่เราได้เสนอ การพิสูจน์ว่าฟังก์ชันใดๆเป็นคอนเวกซ์สามารถทำได้โดยการแสดงว่า hessian matrix ของฟังก์ชันนั้นเป็น positive semi-definite และจาก (4.14) จะสามารถแยกตรวจสอบทีละฟังก์ชันได้เนื่องจากผลลัพธ์ที่ได้จากการบวกของคอนเวกซ์ฟังก์ชันนั้นยังเป็นคอนเวกซ์อยู่ ฉะนั้นจึงแยกออกเป็นฟังก์ชันย่อยๆได้สองฟังก์ชันคือ  $|p_j^t - p_i^t|$  และ  $s_i \cdot p_i^t$  ฟังก์ชันหลังนั้นเป็นผลรวมเชิงเส้นซึ่งเป็นคอนเวกซ์ฟังก์ชันอยู่แล้ว จึงพิจารณาแค่เฉพาะฟังก์ชันแรก อันดับแรกทำการหา Hessian matrix ของฟังก์ชันโดย

$$\frac{\partial}{\partial x_i} (|p_j^t - p_i^t|) = - \frac{x_j^t - x_i^t}{|p_j^t - p_i^t|} \quad (4.15)$$

$$\frac{\partial}{\partial y_i} (|p_j^t - p_i^t|) = - \frac{y_j^t - y_i^t}{|p_j^t - p_i^t|} \quad (4.16)$$

$$\frac{\partial}{\partial x_j} (|p_j^t - p_i^t|) = \frac{x_j^t - x_i^t}{|p_j^t - p_i^t|} \quad (4.17)$$

$$\frac{\partial}{\partial y_j} (|p_j^t - p_i^t|) = \frac{y_j^t - y_i^t}{|p_j^t - p_i^t|} \quad (4.18)$$

$$\begin{aligned} \frac{\partial}{\partial x_k} (|p_j^t - p_i^t|) &= 0 \\ \frac{\partial}{\partial y_k} (|p_j^t - p_i^t|) &= 0 \end{aligned} \quad | \quad k \neq i, k \neq j, k = 1, \dots, N \quad (4.19)$$

และเมื่อทำการจัดเรียงให้พจน์ที่มีค่าเป็น 0 อยู่ต่อท้ายพจน์ที่มีค่าทั้งหมดก็จะได้ Hessian matrix เป็นดังนี้

$$\nabla^2 F = \begin{bmatrix} \frac{(y_j - y_i)^2}{Z} & -\frac{(y_j - y_i)^2}{Z} & -\frac{(x_j - x_i)(y_j - y_i)}{Z} & \frac{(x_j - x_i)(y_j - y_i)}{Z} & 0 & \dots & 0 \\ -\frac{(y_j - y_i)^2}{Z} & \frac{(y_j - y_i)^2}{Z} & \frac{(x_j - x_i)(y_j - y_i)}{Z} & -\frac{(x_j - x_i)(y_j - y_i)}{Z} & 0 & \dots & 0 \\ -\frac{(x_j - x_i)(y_j - y_i)}{Z} & \frac{(x_j - x_i)(y_j - y_i)}{Z} & \frac{(x_j - x_i)^2}{Z} & -\frac{(x_j - x_i)^2}{Z} & 0 & \dots & 0 \\ \frac{(x_j - x_i)(y_j - y_i)}{Z} & -\frac{(x_j - x_i)(y_j - y_i)}{Z} & -\frac{(x_j - x_i)^2}{Z} & \frac{(x_j - x_i)^2}{Z} & 0 & \dots & 0 \\ 0 & 0 & 0 & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & 0 & \dots & 0 \end{bmatrix} \quad (4.20)$$

โดยที่

$$Z = ((x_j - x_i)^2 + (y_j - y_i)^2)^{\frac{3}{2}} \quad (4.21)$$

หลังจากนั้นนำ Sylvester criterion [41] มาเพื่อใช้ในการตรวจสอบ determinant ของ matrix ซึ่ง Sylvester criterion จะตรวจสอบ determinant ของทุกเมตริกซ์ย่อยบนซ้ายขนาด  $n \times n$ ,  $n = 1, \dots, 4N$  ของเมตริกซ์ที่เราสนใจเท่านั้นเราจึงสามารถที่จะตรวจสอบเพียงแค่เมตริกซ์ขนาด  $4 \times 4$  บนซ้ายของ  $\nabla^2 F$  ก็เพียงพอ และผลที่ได้จากการทดสอบตาม Sylvester criterion จะได้ว่าเมตริกซ์นี้เป็น positive semi-definite ซึ่งทำให้ฟังก์ชันนี้เป็นคอนเวกซ์ฟังก์ชัน แสดงให้เห็นว่าเราสามารถหาค่าตอบได้เสมอและจะมีค่าต่ำสุดเพียงค่าเดียวแต่ค่าตอบที่มาจากค่าต่ำสุดนี้อาจจะอยู่ติดกันเป็นกลุ่ม เมื่อเราทราบว่าเป็นคอนเวกซ์ฟังก์ชันนั้นก็ยังสามารถนำขั้นตอนวิธีที่ใช้สำหรับแก้ปัญหาการหาค่าเหมาะสมที่สุดเชิงคอนเวกซ์ต่างๆมาใช้งานได้สิ่งที่เราต้องการที่จะหาคือจุดที่ทำให้ฟังก์ชันจุดประสงค์มีค่าน้อยที่สุด (4.14) โดยไม่มีข้อบังคับอื่นๆ หลังจากทำการลองจัดรูปของสมการ (4.14) ใหม่เพื่อนำความคลาดเคลื่อนมาแสดงแทนจะได้สมการใหม่ดังรูป

$$F(X_t) = \sum_{i=1}^N \sum_{j=1|j \neq i}^N (|p_j^t - p_i^t| (1 - \cos \angle(p_j^t - p_i^t, d_{ij}^t))) + \sum_{i=1}^N (p_i^t \cdot \sum_{j=1|j \neq i}^N e_{ij}^t) \quad (4.22)$$

จากสมการนี้จะเห็นว่าค่าความคลาดเคลื่อนนั้นจะอยู่ในพจน์หลังเพียงพจน์เดียว เพราะฉะนั้นในกรณีที่ไม่มีความคลาดเคลื่อนเลยจะทำให้พจน์หลังหายไปเหลือเพียงแค่

$$F(X_t) = \sum_{i=1}^N \sum_{j=1|j \neq i}^N (|p_j^t - p_i^t| (1 - \cos \angle(p_j^t - p_i^t, d_{ij}^t))) \quad (4.23)$$

และจากสมการนี้จะเห็นว่ากรณีที่  $F(X_t)$  นั้นจะมีค่าน้อยที่สุดจะไม่มีทางต่ำกว่า 0 อย่างแน่นอนเนื่องจากระยะระหว่างจุดไม่มีทางต่ำกว่า 0 และผลลัพธ์จากฟังก์ชัน  $\cos$  นั้นไม่มีทางมากกว่า 1 เพราะฉะนั้นค่า  $F(X_t)$  ที่ต่ำที่สุดเท่าที่จะเป็นไปได้คือ 0 และจะเกิดขึ้นได้จากสองกรณี

1. ค่าระยะทางระหว่างทุกคู่จุดเป็น 0
2. มุมระหว่าง  $p_j^t - p_i^t, d_{ij}^t$  มีค่าเป็น 0 สำหรับทุกคู่มุม

สำหรับกรณีที่หนึ่งนั้นก็หมายถึงทุกจุดนั้นอยู่ ณ ตำแหน่งเดียวกันทั้งหมด ซึ่งในกรณีนี้จะทำให้ไม่สามารถหาค่ามุมระหว่างคู่จุดใดๆได้เลย ในงานวิจัยชิ้นนี้เราจะไม่สนใจในกรณีนี้เนื่องจากในโลกจริงนั้นไม่มีทาง

ที่หุ่นยนต์สองตัวจะอยู่ในตำแหน่งเดียวกันได้ ส่วนกรณีที่สองนั้นเป็นกรณีที่น่าสนใจ เพราะความหมายของกรณีนี้คือตำแหน่งมุมของหุ่นยนต์ทุกตัวจะต้องตรงกันกับข้อมูลมุมที่วัดได้ และยิ่งไปกว่านั้นจะเห็นว่าตำแหน่งของกลุ่มหุ่นยนต์ที่จะทำให้  $F(X_t)$  มีค่าเป็น 0 นั้นจะขึ้นอยู่กับมุมเพียงอย่างเดียว ไม่ขึ้นอยู่กับระยะทางหรือว่าตำแหน่งของกลุ่มหุ่นยนต์เลย แสดงว่าตำแหน่งของกลุ่มหุ่นยนต์ทั้งหมดที่ให้ค่า  $F(X_t)$  ต่ำสุดจะแตกต่างกันที่ขนาดและตำแหน่ง นั่นก็คือฟังก์ชันจุดประสงค์จะให้ค่าไม่แปรผันตามการขยายขนาดและการเปลี่ยนตำแหน่ง (translation invariance and scale invariance)

ขั้นตอนวิธีที่นำเสนอทั้งสองแบบจะให้ผลลัพธ์ออกมาเป็นรูปแบบตำแหน่งของกลุ่มหุ่นยนต์ซึ่งไม่ใช่ตำแหน่งจริงๆของหุ่นยนต์ ในบทถัดไปเราจะนำเสนอวิธีที่จะใช้ในการหาตำแหน่งที่แท้จริงของหุ่นยนต์โดยอาศัยตำแหน่งจริงๆของหุ่นยนต์บางตัว และในบทที่ 6 จะนำเสนอการทดลองสำหรับขั้นตอนวิธีทั้งสองแบบนี้ การทดลองนั้นจะทดลองในส่วนของความถูกต้องของขั้นตอนวิธีค่าความคลาดเคลื่อนที่เกิดขึ้นจากขั้นตอนวิธี และเพื่อทดสอบขั้นตอนวิธีที่นำเสนอในการใช้งานจริง



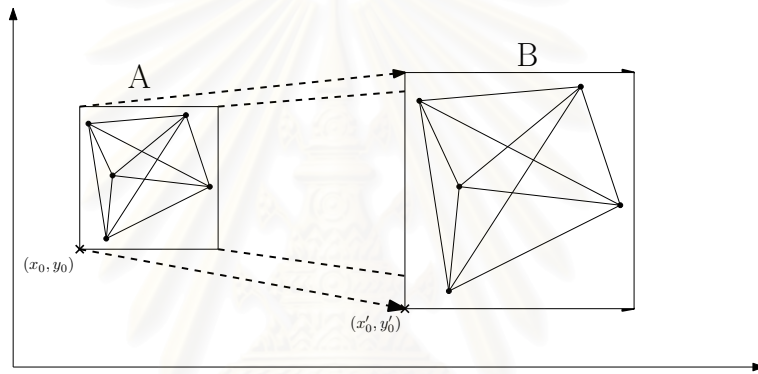
สถาบันวิทยบริการ  
จุฬาลงกรณ์มหาวิทยาลัย

## บทที่ 5

### การหาตำแหน่งที่แท้จริงจากรูปแบบตำแหน่งของกลุ่มหุ่นยนต์

จากขั้นตอนวิธีที่นำเสนอไปในบทที่ 4 จะได้คำตอบออกมาคือรูปแบบตำแหน่งของกลุ่มหุ่นยนต์แต่ในการใช้งานจริงบางครั้งสิ่งที่เราต้องการจะเป็นตำแหน่งในโลกจริงของหุ่นยนต์ ในบทนี้จะเสนอขั้นตอนวิธีที่ใช้หาตำแหน่งในโลกจริงของหุ่นยนต์จากรูปแบบตำแหน่งของกลุ่มหุ่นยนต์และตำแหน่งที่แท้จริงของหุ่นยนต์บางตัว

สมมติให้รูปแบบตำแหน่งของกลุ่มหุ่นยนต์ที่ได้จากขั้นตอนวิธีในบทที่ 4 เป็นดังตำแหน่ง A ในรูป [5.1] และตำแหน่งที่แท้จริงของกลุ่มหุ่นยนต์เป็นดังตำแหน่ง B เราสามารถที่จะเปลี่ยนตำแหน่ง A ให้เป็นตำแหน่ง B ได้โดยการเคลื่อนที่และขยายขนาด กล่าวโดยทั่วไปคือขั้นตอนวิธีในการหาตำแหน่งที่แท้



รูปที่ 5.1: แสดงรูปแบบตำแหน่งของกลุ่มหุ่นยนต์(A)และตำแหน่งจริงๆของกลุ่มหุ่นยนต์(B) ซึ่งสามารถมองให้อยู่ในลักษณะของรูปภาพได้ และการหาตำแหน่งที่แท้จริงของหุ่นยนต์ก็คือการเคลื่อนย้ายและขยายขนาดรูป A ให้กลายเป็นรูป B

จริงของกลุ่มหุ่นยนต์นั้นก็เหมือนกันกับการหาเวกเตอร์การเคลื่อนที่ของกลุ่มหุ่นยนต์ และอัตราขยายที่เหมาะสมเพื่อให้ตำแหน่งของรูปแบบกลุ่มหุ่นยนต์นั้นตรงกับตำแหน่งจริงๆมากที่สุด ถัดไปจะทำการนิยามปัญหาและเสนอขั้นตอนวิธีในการหาตำแหน่งจริงๆจากรูปแบบตำแหน่ง

#### 5.1 นิยามของปัญหา

กำหนดให้รูปแบบการวางตัว  $C = (c_1, \dots, c_N | c_i \in \mathbb{R}^2, i = 1 \dots N)$  (ในกรณีที่สภาพแวดล้อมเป็นระนาบสองมิติ) และกำหนดให้  $A$  เป็นเวกเตอร์ตำแหน่งที่แท้จริงของหุ่นยนต์  $A = (a_1, \dots, a_N | a_i \in \mathbb{R}^2, i = 1 \dots N)$  โดยถ้าหุ่นยนต์ตัวที่  $i$  นั้นไม่รู้ตำแหน่งที่แท้จริงค่า  $a_i = \infty$  และ กำหนดให้  $C' = (c'_1, \dots, c'_M)$  เป็นตำแหน่งหุ่นยนต์ในรูปแบบการวางตัวที่รู้ตำแหน่งที่แท้จริงโดย  $M$  นั้นคือจำนวนของหุ่นยนต์ที่รู้ตำแหน่งที่แท้จริงและให้  $A' = (a'_1, \dots, a'_M)$  เป็นตำแหน่งที่แท้จริงของหุ่นยนต์ที่สอดคล้องกับ  $C'$  นั่นก็คือ  $a'_i$  จะเป็นตำแหน่งที่แท้จริงของหุ่นยนต์ตัวที่  $i$  และ  $c'_i$  จะเป็นตำแหน่งหุ่นยนต์ตัวที่  $i$  ในรูปแบบการวางตัวของกลุ่มหุ่นยนต์ สิ่งที่เราต้องการจะหาก็คือเวกเตอร์การเคลื่อนที่และอัตราขยายที่ทำให้ความแตกต่างระหว่าง  $a'_i$  และ  $c'_i$  น้อยที่สุดกำหนดให้  $T, T \in \mathbb{R}^2$  คือเวกเตอร์การเคลื่อนที่และให้  $\eta, \eta \in \mathbb{R}, \eta \neq 0$

เป็นอัตราขยาย ถ้าเราให้อัตราขยายนั้นเทียบกับจุดกำเนิดจะได้

$$c_i^* = \eta c_i' + T$$

และให้  $C^* = (c_1^*, \dots, c_M^*)$  และสิ่งที่เราต้องการจะหาคือ

$$\text{minimize } \omega(\eta, T) \quad (5.1)$$

โดยฟังก์ชัน  $\omega$  นั้นเป็นฟังก์ชันที่ใช้เปรียบเทียบ ระหว่าง กลุ่ม จุด หุ่นยนต์ ( $C^*$ ) และ ตำแหน่ง จริง ( $A'$ ) ถ้าใกล้เคียงกันมากจะให้ค่าน้อย ซึ่งเราจะต้องหาค่า  $\eta$  และ  $T$  ที่ทำให้ค่าฟังก์ชันน้อยที่สุด จะเห็นได้ว่าจำนวนตัวแปรที่เราต้องทำการหาค่านั้นมีเท่ากับ 3 ตัวฉะนั้นเราจึงจำเป็นต้องได้ ข้อมูล ตำแหน่ง จริง ของหุ่นยนต์อย่างน้อยสองตัว เพื่อที่ ทำให้สามารถหาคำตอบได้ ในส่วนถัดไปเราจะอธิบายถึงฟังก์ชันที่เลือกมาใช้และขั้นตอนวิธีเพื่อใช้ในการหาค่า  $\eta$  และ  $T$

## 5.2 ขั้นตอนการหาเวกเตอร์การเคลื่อนที่และอัตราขยาย

ฟังก์ชันที่สามารถนำมาใช้เปรียบเทียบ ( $\omega$ ) นั้นมีได้หลายฟังก์ชัน แต่ในวิทยานิพนธ์ฉบับนี้เราจะเลือกฟังก์ชันระยะทางกำลังสองมาใช้งาน เนื่องจากฟังก์ชันนี้สามารถที่จะหาค่าต่ำสุดได้ง่ายโดยมีรูปแบบปิด ทำให้ความซับซ้อนเชิงเวลาน้อย ลักษณะของฟังก์ชันระยะทางกำลังสองคือ

$$\text{minimize } f(x) = \|Ax - b\|^2 = \sum_{i=1}^k (a_i^T x - b_i)^2 \quad (5.2)$$

โดยที่  $A \in \mathbb{R}^{k \times n}$  และ  $(k \geq n)$   $a_i^T$  เป็นแถวของ  $A$  และเวกเตอร์  $x \in \mathbb{R}^n$  เป็นตัวแปรค่าเหมาะที่สุดเมื่อเปรียบเทียบกับปัญหา 5.1 เราก็สามารถเขียนใหม่ได้ดังนี้

$$\text{minimize } \omega(\eta, T) = \sum_{i=1}^M (\|\eta c_i' + T - a_i'\|^2) \quad (5.3)$$

หลังจากนั้นก็ใช้ขั้นตอนการแก้ปัญหาแบบกำลังสองน้อยสุดมาใช้ดังต่อไปนี้ จากปัญหา 5.3 เราจะเขียนให้อยู่ในรูปของ  $x$  และ  $y$  ได้ดังนี้

$$\text{minimize } \omega(\eta, T) = \sum_{i=1}^M ((\eta x_i' + x^T - x^{a'})^2 + (\eta y_i' + y^T - y^{a'})^2) \quad (5.4)$$

โดยที่  $x_i'$  และ  $y_i'$  เป็นส่วน  $x$  และ  $y$  ของ  $c_i'$ ,  $x^T$  และ  $y^T$  เป็นส่วน  $x$  และ  $y$  ของ  $T$  และ  $x^{a'}$  และ  $y^{a'}$  เป็น  $x$  และ  $y$  ของ  $a_i'$  ต่อไปทำการหาค่าอนุพันธ์ย่อยต่างๆ

$$\frac{\partial}{\partial \eta} \omega = 2 \left( \sum_{i=1}^M (x^T + \eta x_i' - x^{a'}) (x_i') + \sum_{i=1}^M (y^T + \eta y_i' - y^{a'}) (y_i') \right) \quad (5.5)$$

$$\frac{\partial}{\partial x^T} \omega = 2 \sum_{i=1}^M (x^T + \eta x_i' - x^{a'}) \quad (5.6)$$



$$\frac{\partial}{\partial y^T} \omega = 2 \sum_{i=1}^M (y^T + \eta y'_i - y^{a'}) \quad (5.7)$$

ต่อไปทำการหาจุดที่ทำให้อนุพันธ์ย่อยทั้งสามตัวมีค่าเป็น 0 จะได้

$$\sum_{i=1}^M (x^T + \eta x'_i - x^{a'}) (x'_i) + \sum_{i=1}^M (y^T + \eta y'_i - y^{a'}) (y'_i) = 0 \quad (5.8)$$

$$\sum_{i=1}^M (x^T + \eta x'_i - x^{a'}) = 0 \quad (5.9)$$

$$\sum_{i=1}^M (y^T + \eta y'_i - y^{a'}) = 0 \quad (5.10)$$

จัดรูปสมการ 5.8 และ 5.9 ใหม่จะได้

$$x^T = \frac{\sum x^{a'} - \eta \sum x'_i}{M} \quad (5.11)$$

$$y^T = \frac{\sum y^{a'} - \eta \sum y'_i}{M} \quad (5.12)$$

แล้วนำสมการ 5.11 และ 5.12 ไปแทนลงใน 5.10 จะได้

$$\eta = \frac{1}{\sum (x'_i)^2 + \sum (y'_i)^2 - \frac{\sum^2 x'_i + \sum^2 y'_i}{M}} \left( \sum x^{a'} x'_i + \sum y^{a'} y'_i - \frac{\sum x^{a'} \sum x'_i + \sum y^{a'} \sum y'_i}{M} \right) \quad (5.13)$$

และเมื่อนำไปแทนค่าลงในสมการ 5.11 และ 5.12 ก็จะได้คำตอบทั้งหมด หลังจากนั้นก็ค่าที่ได้ไปใส่ลงในสมการของ  $c_i^*$  ก็จะได้ตำแหน่งที่แท้จริงของหุ่นยนต์แต่ละตัว

ในกรณีที่ความแม่นยำในการระบุตำแหน่งที่แท้จริงของหุ่นยนต์แต่ละตัวไม่เท่ากันและเราทราบถึงอัตราส่วนของความแม่นยำระหว่างหุ่นยนต์แต่ละตัว เราก็สามารถที่จะใช้ขั้นตอนวิธีระยะทางกำลังสองน้อยสุดแบบถ่วงน้ำหนักได้ ซึ่งจะมีรูปแบบสมการดังนี้

$$\text{minimize } \omega(\eta, T) = \sum_{i=1}^M w_i ((\eta x'_i + x^T - x^{a'})^2 + (\eta y'_i + y^T - y^{a'})^2)$$

โดยที่  $w_i$  เป็นค่าถ่วงน้ำหนักสำหรับหุ่นยนต์แต่ละตัวและจะได้

$$\eta = \frac{1}{\sum w_i (x'_i)^2 + \sum w_i (y'_i)^2 - \frac{\sum^2 x'_i w_i + \sum^2 y'_i w_i}{\sum w_i}} \left( \sum x^{a'} x'_i w_i + \sum y^{a'} y'_i w_i - \frac{\sum x^{a'} w_i \sum x'_i w_i + \sum y^{a'} w_i \sum y'_i w_i}{\sum w_i} \right) \quad (5.14)$$

$$x^T = \frac{\sum w_i x^{a'} - \eta \sum w_i x'_i}{\sum w_i} \quad (5.15)$$

$$y^T = \frac{\sum w_i y^{a'} - \eta \sum w_i y'_i}{\sum w_i} \quad (5.16)$$

ในบทนี้เราได้ นำเสนอ ขั้นตอนวิธีที่ใช้หาตำแหน่งที่แท้จริงของหุ่นยนต์จาก รูปแบบ ตำแหน่ง ของ กลุ่ม

หุ้ยนน็ดแลลตำแหน่งที่แท้จริงของหุ้ยนน็ดอย่างน้อยสองตัว โดยอาศัยขั้นตอนวิธีระยะทางกำลังสองน้อยที่สุดทั้งแบบที่ไม่มีการถ่วงน้ำหนักและแบบที่มีการถ่วงน้ำหนัก จะเห็นได้ว่าความซับซ้อนเชิงเวลาของขั้นตอนวิธีที่นำเสนอั้นมีค่าเท่ากับ  $O(N)$  เมื่อ  $N$  คือจำนวนของหุ้ยนน็ดทั้งหมด



สถาบันวิทยบริการ  
จุฬาลงกรณ์มหาวิทยาลัย

## บทที่ 6

### การทดลองและผลการทำงาน

ในบทนี้เราจะทำการทดลองขั้นตอนวิธีที่ได้นำเสนอไปแล้วในบทที่ 4 การทดลองนี้ทำขึ้นเพื่อทดสอบความถูกต้องของขั้นตอนวิธีที่นำเสนอและวัดค่าความคลาดเคลื่อนที่อาจเกิดขึ้นจากขั้นตอนวิธีโดยเราจะทำการทดลองในสภาพแวดล้อมสองแบบนั่นคือในสภาพแวดล้อมแบบจำลอง และสภาพแวดล้อมจริง

#### 6.1 สภาพแวดล้อมจำลอง

เรากำหนดให้สภาพแวดล้อมจำลองที่จะใช้ในการทดลองของขั้นตอนวิธีทั้งสองแบบนี้มีลักษณะเป็นระนาบสองมิติที่มีพื้นที่เท่ากับ  $w \times h$  ตารางหน่วยโดยในกรณีที่ไม่ว่ากล่าวถึงพื้นที่ทำการทดลองนั้นจะหมายถึงพื้นที่ทำการทดลองมีค่าเท่ากับ  $640 \times 480$  ตารางหน่วยและกำหนดให้หุ่นยนต์แต่ละตัวมีลักษณะเป็นจุดไม่มีมิติ นั่นคือหุ่นยนต์สองตัวใดๆสามารถที่เห็นกันได้เสมอ อุปกรณ์การรับรู้ของหุ่นยนต์จะมีเพียงกล้องที่สามารถรับข้อมูลตำแหน่งเชิงมุมของหุ่นยนต์ตัวอื่นๆและเข็มทิศเท่านั้นข้อมูลที่วัดได้จะมีการปนเปื้อนด้วยความคลาดเคลื่อน ซึ่งค่าความคลาดเคลื่อนสำหรับข้อมูลแต่ละอันจะขึ้นอยู่กับค่าความแปรปรวนของอุปกรณ์ที่ใช้ในการวัด ยกตัวอย่างเช่น การวัดด้วยกล้องนั้นอาจจะมีค่าความแปรปรวนเท่ากับ  $\delta_c$  และข้อมูลที่วัดด้วยเข็มทิศอาจจะมีค่าความแปรปรวนเท่ากับ  $\delta_p$  และความแปรปรวนสำหรับอุปกรณ์ที่เหมือนกันอาจจะมีค่าไม่เท่ากันได้ แต่เพื่อความสะดวกในการทดลองและวัดผล เราจึงกำหนดค่าความแปรปรวนขึ้นมาค่าหนึ่ง ซึ่งเป็นความแปรปรวนจากการวัดด้วย กล้อง และเข็มทิศ และให้เท่ากันสำหรับหุ่นยนต์ทุกตัว

##### 6.1.1 การทดลองสำหรับขั้นตอนวิธีการช่วยกันระบุตำแหน่งแบบที่ 1

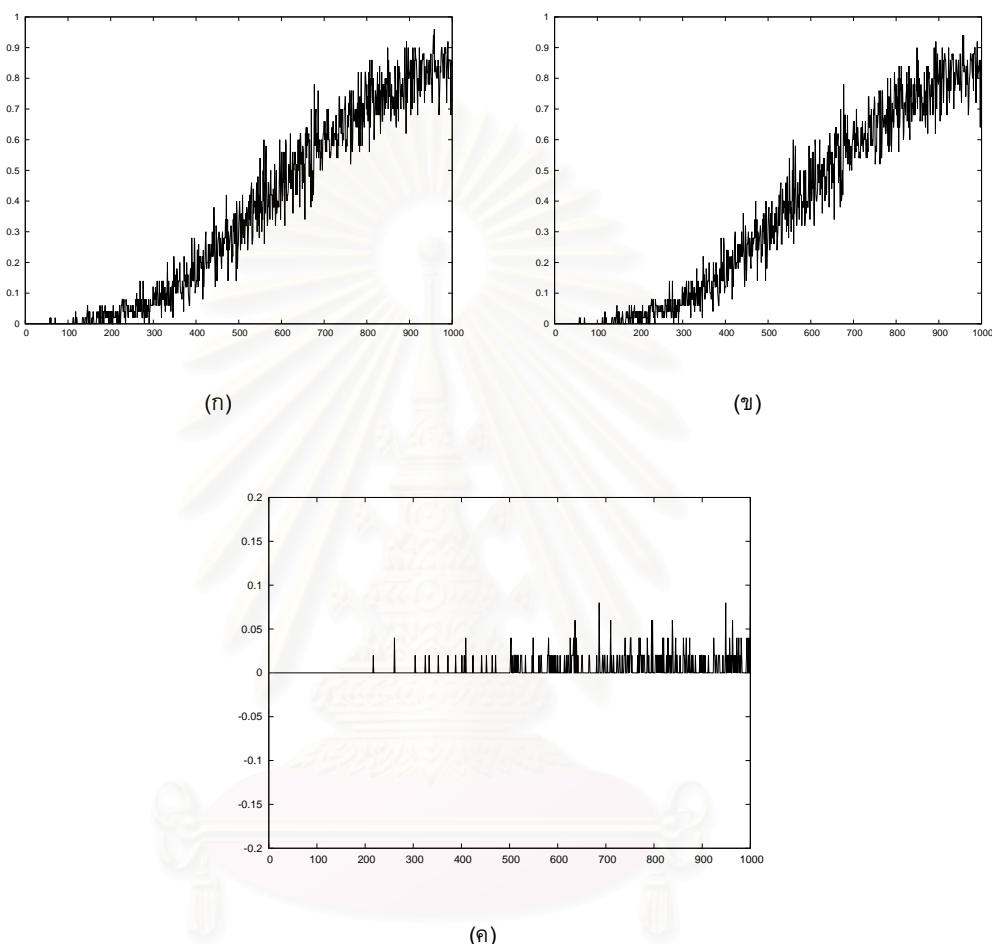
จากข้อจำกัดของขั้นตอนวิธีที่ต้องอาศัยข้อมูลเชิงมุมที่ไม่มีความคลาดเคลื่อน เราจึงกำหนดค่าความแปรปรวนของทั้งระบบเป็นศูนย์สิ่งที่เราจะทำการทดลองสำหรับขั้นตอนวิธีนี้ก็คือ อัตราความถูกต้องในการระบุตำแหน่งเมื่อจำนวนของหุ่นยนต์เปลี่ยนไป จากการที่ขั้นตอนวิธีนี้ไม่สมบูรณ์นั่นคือคำตอบที่ได้อาจจะไม่ถูกต้องเสมอ

#### ขั้นตอนการทำงานทดลอง

1. สุ่มตำแหน่งของหุ่นยนต์แต่ละตัวทั้งหมด  $N$  ตัว โดยมีลักษณะการกระจายตัวแบบสม่ำเสมอทั่วพื้นที่ทำการทดลอง
2. หุ่นยนต์แต่ละตัวจะทำการวัดข้อมูลเชิงมุมเทียบกับแกน  $x$  (เราจะข้ามขั้นตอนการปรับข้อมูลเชิงมุมด้วยข้อมูลจากเข็มทิศไป เนื่องจากการวัดข้อมูลเทียบกับแกน  $x$  จะมีค่าเท่ากับการวัดตำแหน่งเชิงมุมของหุ่นยนต์ตัวอื่นๆแล้วปรับด้วยข้อมูลจากเข็มทิศ)
3. นำข้อมูลทั้งหมดมาใช้งานตามขั้นตอนวิธี
4. วัดผลการทดลอง

5. ทำซ้ำขั้นตอนตั้งแต่ 1-4 จนครบ 50 ครั้งและหาค่าความน่าจะเป็นที่คำตอบไม่ถูกต้อง

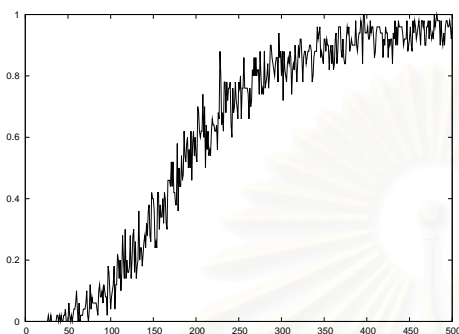
จำนวนหุ่นยนต์ที่ใช้ในการทดลองเท่ากับ  $10, \dots, 1000$  ตัว ตำแหน่งของหุ่นยนต์แต่ละตัวได้มาจากการสุ่มตำแหน่งหุ่นยนต์โดยใช้การกระจายตัวแบบสม่ำเสมอ กราฟ [6.1] แสดงผลจากการทดลอง จะเห็น



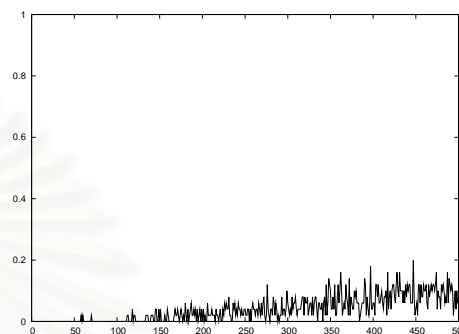
รูปที่ 6.1: กราฟแสดงค่าความผิดพลาดของขั้นตอนวิธี แกน x แสดงจำนวนหุ่นยนต์ แกน y แสดงค่าความน่าจะเป็นที่จะผิดพลาดรูป ก. พื้นที่ทดลองมีขนาด  $640 \times 480$  รูป ข. พื้นที่ทดลองมีขนาด  $320 \times 240$  รูป ค. แสดงถึงผลต่างระหว่างรูป ก. และ รูป ข.

ว่าในช่วง 0-100 ตัวนั้นแทบจะไม่มีค่าผิดพลาดเลยแต่เมื่อจำนวนหุ่นยนต์เกินกว่า 600 ตัวเป็นต้นไป ความผิดพลาดจะสูงเกินกว่า 50% ต่อไปเราจะทำการทดลองเปลี่ยนความหนาแน่นของหุ่นยนต์ต่อพื้นที่เพื่อทดสอบว่าขั้นตอนวิธีนี้ได้รับผลกระทบจากความหนาแน่นที่เปลี่ยนไปหรือไม่โดยการลดขนาดพื้นที่ทำการทดลองลงเหลือ  $320 \times 240$  ตารางหน่วยและจำนวนหุ่นยนต์ที่นำมาทำการทดลองเท่ากับ  $10, \dots, 1000$  ตัว และทำการทดลอง 50 ครั้งต่อจำนวนหุ่นยนต์หนึ่งจำนวนแล้วหาค่าความน่าจะเป็นที่คำตอบผิด กราฟ[6.1ข.] แสดงผลจากการทดลอง จะเห็นว่าค่าความน่าจะเป็นที่คำตอบผิดนั้นมีความแตกต่างน้อยมากเมื่อความหนาแน่นของหุ่นยนต์เปลี่ยนไป กราฟ [6.1ค.] แสดงค่าความแตกต่างระหว่างความน่าจะเป็นของกราฟ [6.1ก.] และกราฟ [6.1ข.]

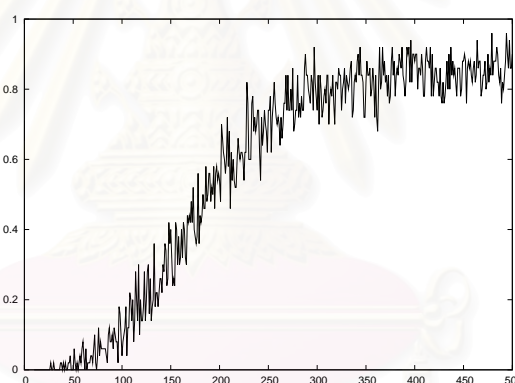
การทดลองถัดมาจะทำการทดสอบว่าผลของขั้นตอนวิธีนี้จะขึ้นอยู่กับรูปร่างของสภาพแวดล้อมที่ใช้ในการทดลองหรือไม่ โดยทำการทดลองกับพื้นที่ที่มีอัตราส่วนระหว่างความกว้างและความยาวเปลี่ยนไปจากเดิมเป็น  $1000 \times 10$  และเทียบกับ  $116 \times 87$  ซึ่งทั้งสองสภาพแวดล้อมนั้นมีพื้นที่ประมาณเท่ากัน แต่อัตราส่วนระหว่างความกว้างและความยาวไม่เท่ากัน จำนวนหุ่นยนต์ที่ใช้ทดลองเท่ากับ  $10, \dots, 500$  กราฟ [6.2ก.] แสดงผลจากพื้นที่  $1000 \times 10$  กราฟ [6.2ข.] แสดงผลจากพื้นที่  $116 \times 87$  และ กราฟ



(ก)



(ข)



(ค)

รูปที่ 6.2: กราฟแสดงค่าความผิดพลาดของขั้นตอนวิธี แกน x แสดงจำนวนหุ่นยนต์ แกน y แสดงค่าความน่าจะเป็นที่จะผิดพลาดในรูปก. พื้นที่ทดลองมีขนาด  $1000 \times 10$  ในรูปข. พื้นที่ทดลองมีขนาด  $116 \times 87$  และรูปค. แสดงถึงผลต่างระหว่างรูป ก. และ รูป ข.

[6.2ค.] แสดงค่าผลต่างความน่าจะเป็นระหว่างสองกราฟก่อนหน้านี้ จะสามารถเห็นได้อย่างชัดเจนว่ารูปร่างของสภาพแวดล้อมนั้นจะส่งผลต่อค่าความน่าจะเป็นที่คำตอบจะผิดอย่างชัดเจน

เราทำการทดลองขั้นตอนวิธีนี้เฉพาะในสภาพแวดล้อมจำลองเท่านั้นเนื่องจาก ในสภาพแวดล้อมจริงค่าความคลาดเคลื่อนในการวัดจะเกิดขึ้นเสมอ



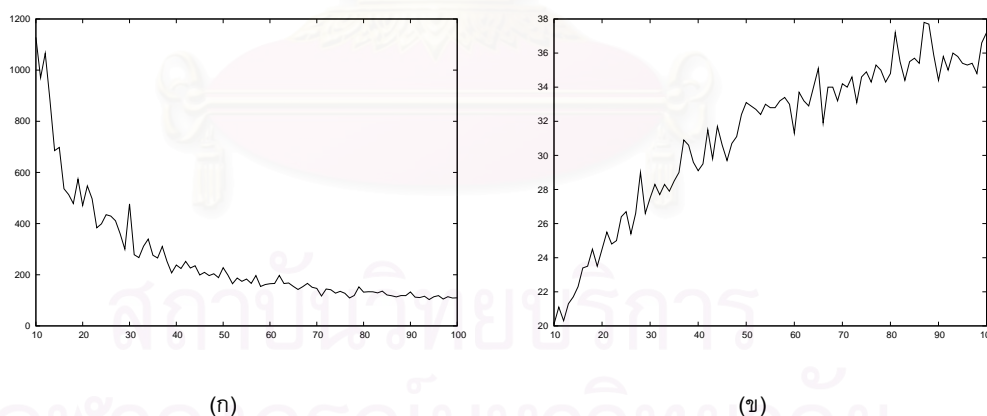
### 6.1.2 การทดลองสำหรับขั้นตอนวิธีการช่วยกันระบุตำแหน่งแบบที่ 2

ปัญหาการร่วมกันระบุตำแหน่งโดยใช้เพียงข้อมูลเชิงมุมนั้นสามารถจะเขียนให้อยู่ในรูปปัญหาการหาค่าเหมาะที่สุดเชิงคอนเวกซ์ได้ดังแสดงไว้ในบทที่ 4 และวิธีการแก้ปัญหาสำหรับปัญหาประเภทนี้มีหลายวิธี ซึ่งในวิทยานิพนธ์ฉบับนี้เสนอไว้สองวิธีนั่นคือวิธี ไล่ทางลาด และ วิธี interior point สำหรับวิธีแก้ปัญหาแบบไล่ทางลาดนั้นเราทำการทดลองโดยการเขียนโปรแกรมด้วยภาษา C# และสำหรับวิธี interior point นั้นเราใช้ cvx ซึ่งเป็นส่วนต่อขยายของโปรแกรม MATLAB เป็นตัวแก้ปัญหา โดยเป็นตัวแก้ปัญหาที่ใช้สำหรับการแก้ปัญหาการหาค่าเหมาะที่สุดเชิงคอนเวกซ์โดยเฉพาะ

เราจะทำการนิยามค่าความคลาดเคลื่อนที่อยู่ในคำตอบของปัญหาดังนี้กำหนดให้ทิศทางเชิงมุมจากหุ่นยนต์ตัว  $i$  ไปยังหุ่นยนต์ตัว  $j$  ที่ได้จากรูปแบบตำแหน่งคือ  $d'_{ij} = \frac{c_j - c_i}{|c_j - c_i|}$ ,  $i = 1, \dots, N, j = 1, \dots, N, i \neq j$  ความคลาดเคลื่อนของการหาตำแหน่งเราจะคำนวณจากผลรวมของค่าสัมบูรณ์ของผลต่างเชิงมุมระหว่างค่าจริงและคำตอบหรือจะเขียนได้ว่า  $E = \sum_{i=1}^N \sum_{j=1, j \neq i}^N |\angle(d_{ij}, d'_{ij})|$  และค่าเฉลี่ยความคลาดเคลื่อนของการหาตำแหน่งคือ  $\bar{E} = \frac{E}{N}$  และค่าเฉลี่ยความคลาดเคลื่อนของมุมคือ  $\bar{e} = \frac{\bar{E}}{N-1}$

การทดลองแรกเราจะทดลองการเปลี่ยนแปลงจำนวนของหุ่นยนต์เทียบกับระยะเวลาที่ใช้ในการหาคำตอบสำหรับกรณีที่ไม่มีความคลาดเคลื่อนในการวัดจะทำการสุ่มตำแหน่งใหม่ทั้งหมด 10 ครั้งสำหรับจำนวนหุ่นยนต์จำนวนหนึ่งแล้วหาค่าเฉลี่ยระยะเวลา จำนวนหุ่นยนต์ที่ใช้ในการทดลองคือ 10, ..., 100 ตัว

- วิธีการแก้ปัญหาแบบไล่ทางลาด สำหรับวิธีการแก้ปัญหาแบบทำซ้ำระยะเวลาที่ใช้ในการหาคำตอบนั้น



รูปที่ 6.3: แสดงจำนวนขั้นตอนที่ใช้ในการแก้ปัญหา แกน x แสดงจำนวนหุ่นยนต์ แกน y แสดงจำนวนขั้นตอน กราฟ ก. สำหรับวิธีการแก้ปัญหาแบบไล่ทางลาดกราฟ ข. สำหรับวิธีการแก้ปัญหาโดยวิธี interior point

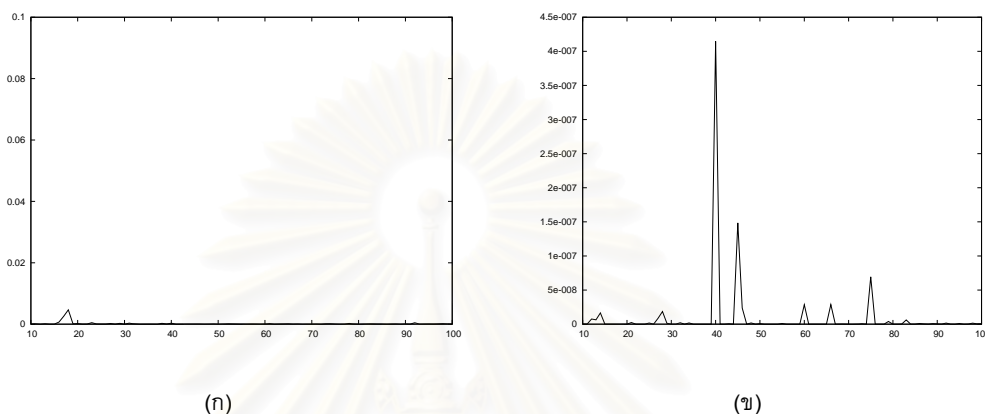
สามารถวัดได้โดยวัดจำนวนรอบที่ทำซ้ำสำหรับหุ่นยนต์แต่ละตัว กราฟ [6.3ก.] แสดงผลที่ได้จากการทดลอง จะสังเกตได้ว่าเมื่อจำนวนของหุ่นยนต์มากขึ้นระยะเวลาที่ใช้ในการทำงานก็จะน้อยลง

- วิธีการแก้ปัญหาโดยวิธี interior point

สำหรับวิธีการวัดของตัว cvx ใน MATLAB ทำโดยการวัดจำนวนรอบทำซ้ำของตัวหาค่าเหมาะที่สุด กราฟ [6.3ข.] แสดงผลที่ได้จากการทดลอง

การทดลองถัดไปเราจะทดลองความคลาดเคลื่อนที่เกิดขึ้นเทียบกับจำนวนหุ่นยนต์ ในกรณีที่ไม่มี ความคลาดเคลื่อนในการวัดจะทำการสุ่มตำแหน่งใหม่ทั้งหมด 10 ครั้งสำหรับจำนวนหุ่นยนต์จำนวนหนึ่ง แล้วหาค่าเฉลี่ยความคลาดเคลื่อนของมุม จำนวนหุ่นยนต์ที่ใช้ในการทดลองคือ  $10, \dots, 100$  ตัว

- วิธีการแก้ปัญหาแบบไต่ทางลาด กราฟ [6.4ก.] แสดงผลที่ได้จากการทดลอง จะสังเกตเห็นว่าค่าเฉลี่ย



รูปที่ 6.4: แสดงค่าเฉลี่ยความคลาดเคลื่อนของมุม แกน x แสดงจำนวนหุ่นยนต์ แกน y แสดงค่าความผิดพลาดหน่วยเป็นเรเดียน กราฟ ก. สำหรับวิธีการแก้ปัญหาแบบไต่ทางลาดกราฟ ข. สำหรับวิธีการแก้ปัญหาโดยวิธี interior point

ความคลาดเคลื่อนของมุมนั้นมีค่าน้อยมาก ซึ่งก็มีความเป็นไปได้ว่าค่าความคลาดเคลื่อนตรงนี้อาจจะเกิดการผลของการคำนวณของตัวเลข

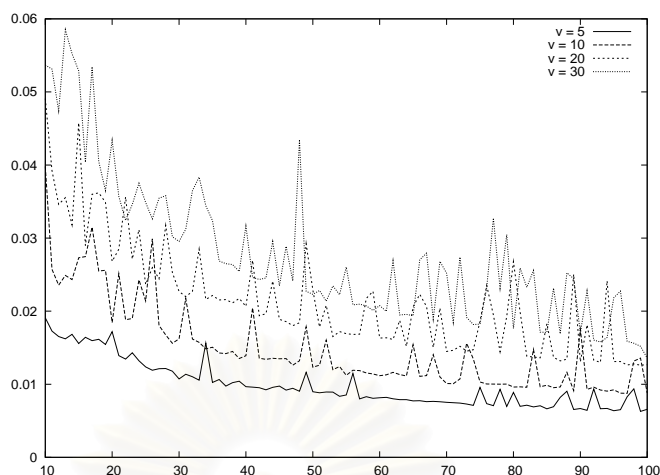
- วิธีการแก้ปัญหาโดยวิธี interior point

กราฟ [6.4ข.] แสดงผลที่ได้จากการทดลอง ค่าเฉลี่ยความคลาดเคลื่อนของมุมนั้นมีค่าน้อยมากเข้าใกล้ศูนย์เสมอไม่ว่าจะเปลี่ยนจำนวนหุ่นยนต์เป็นเท่าไรก็ตามแสดงให้เห็นว่าขั้นตอนวิธีที่เราแนะนำนั้นถูกต้อง

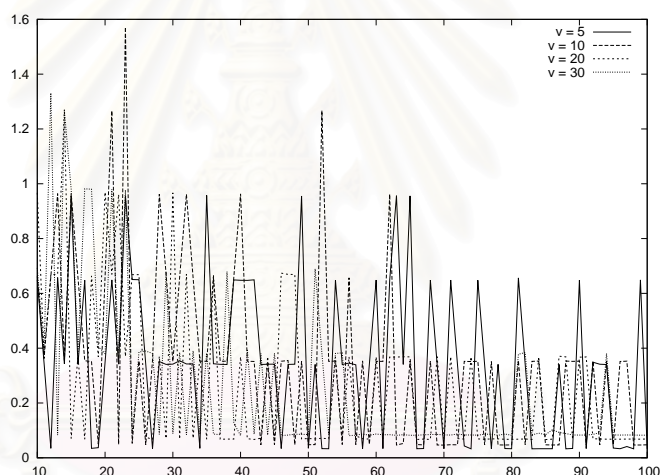
การทดลองถัดไปเราจะทดลองความคลาดเคลื่อนที่เกิดขึ้นเทียบกับจำนวนหุ่นยนต์ ในกรณีที่ค่าความแปรปรวนของวัดมีค่าต่างๆจะทำการสุ่มตำแหน่งใหม่ทั้งหมด 10 ครั้งสำหรับจำนวนหุ่นยนต์จำนวนหนึ่งและค่าความแปรปรวนค่าหนึ่ง แล้วหาค่าเฉลี่ยความคลาดเคลื่อนของมุมจำนวนหุ่นยนต์ที่ใช้ในการทดลองคือ  $10, \dots, 100$  ตัว ค่าความแปรปรวนที่นำมาใช้คือ 5, 10, 20 และ 30

- วิธีการแก้ปัญหาแบบไต่ทางลาด กราฟ [6.5ก.] แสดงผลที่ได้จากการทดลอง จะสังเกตเห็นว่าค่าเฉลี่ยความคลาดเคลื่อนของมุมนั้นมีค่าน้อยลงเมื่อจำนวนหุ่นยนต์มากขึ้นเรื่อยๆ
- วิธีการแก้ปัญหาโดยวิธี interior point

กราฟ [6.5ข.] แสดงผลที่ได้จากการทดลอง ค่าเฉลี่ยความคลาดเคลื่อนของมุมนั้นมีค่าน้อยแต่ไม่ขึ้นอยู่กับจำนวนตัวของหุ่นยนต์ อาจจะเป็นผลมาจากความคลาดเคลื่อนในการคำนวณทศนิยม



(ก)



(ข)

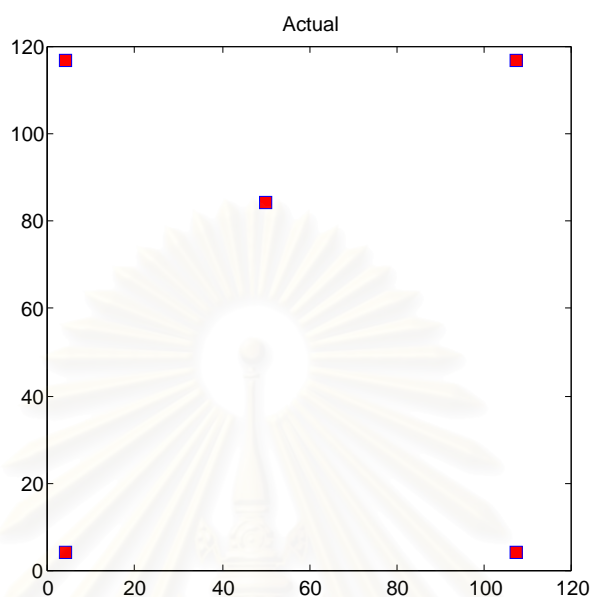
รูปที่ 6.5: แสดงค่าเฉลี่ยความคลาดเคลื่อนของมุม แกน  $x$  แสดงจำนวนหุ่นยนต์ แกน  $y$  แสดงค่าความผิดพลาดหน่วยเป็นเรเดียน กราฟ ก. สำหรับวิธีการแก้ปัญหาแบบได้ทางลาดกราฟ ข. สำหรับวิธีการแก้ปัญหาโดยวิธี interior point

ผลการทดลองสำหรับสภาพแวดล้อมจำลองแสดงให้เห็นว่าขั้นตอนวิธีที่เสนอไปนั้น สามารถนำมาใช้ในปัญหาการร่วมกันระบุตำแหน่งได้ ถัดไปจะทำการเสนอขั้นตอนการทดลองสำหรับสภาพแวดล้อมจริงเพื่อทดสอบขั้นตอนวิธีที่ได้นำเสนอกับสภาพจริง

## 6.2 สภาพแวดล้อมจริง

เนื่องจากข้อจำกัดทางด้านจำนวนหุ่นยนต์และกล้องที่จะใช้ในการทดลอง เราจึงใช้เพียงอุปกรณ์ติดแถบสีแล้ววางอยู่กับที่แทนการใช้หุ่นยนต์จริง อุปกรณ์ที่ใช้ในการวัด ข้อมูลเชิงมุมคือกล้องมองเห็นรอบ

ทิศทางระบบประกอบด้วยกล้องออบนิหนึ่งตัวและแถบสีทั้งหมด 4 แถบสี(นั่นคือการจำลองหุ่นยนต์ทั้งหมด 5 ตัว) ลำดับแรกเราจะทำการกำหนดตำแหน่งของหุ่นยนต์(แถบสี)ที่จะทำการทดลอง โดยตำแหน่งของหุ่นยนต์แต่ละตัวเป็นดังรูป[6.6] เราจะทำการบันทึกภาพสำหรับหุ่นยนต์แต่ละตัวด้วยการแทนที่แถบสี

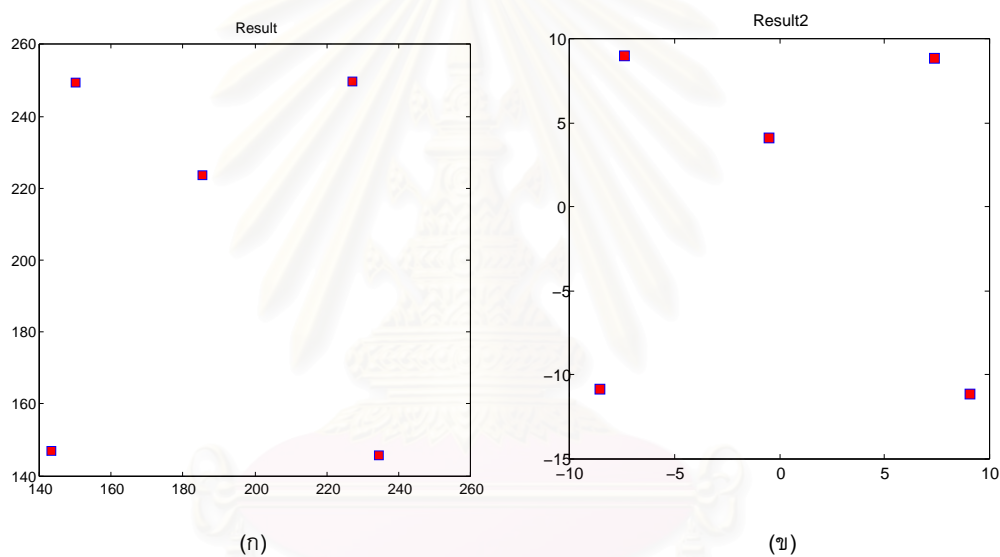


รูปที่ 6.6: แสดงตำแหน่งของหุ่นยนต์แต่ละตัว

ด้วยกล้องออบนิภาพที่ได้จากหุ่นยนต์แต่ละตัวจะนำมาคำนวณหาตำแหน่งเชิงมุมของหุ่นยนต์ตัวอื่นๆ การประมวลผลภาพจะทำการใช้โปรแกรมที่เขียนโดยใช้ภาษา C++ ร่วมกับ OpenCV [42] โดยเมื่อได้ตำแหน่งเชิงมุมสำหรับหุ่นยนต์แต่ละตัวครบแล้วจะนำข้อมูลทั้งหมดไปใส่ให้กับโปรแกรมที่ใช้แก้ปัญหาโดยใช้วิธีการไล่ทางลาดและนำไปใส่ให้กับตัวแก้ปัญหาการหาค่าเหมาะสมที่สุดเชิงคอนเวกซ์(cvx บน MATLAB) ผลการหาตำแหน่งสำหรับขั้นตอนวิธีแบบไล่ทางลาดเป็นดังรูป [6.7ก.] และผลการหาตำแหน่งสำหรับตัวแก้ปัญหาการหาค่าเหมาะสมที่สุดเชิงคอนเวกซ์เป็นดังรูป [6.7ข.]

ในบทถัดไปเราจะสรุปผลการทดลองและสรุปงานวิทยานิพนธ์ รวมถึงเสนอแนวทางในการทำวิจัย  
ขั้นถัดไป

สถาบันวิทยบริการ  
จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ 6.7: แสดงตำแหน่งของหุ่นยนต์ที่เป็นคำตอบ รูป ก. สำหรับวิธีการแก้ปัญหาแบบได้ทางลาดรูป ข. สำหรับวิธีการแก้ปัญหาโดยวิธี interior point

สถาบันวิทยบริการ  
จุฬาลงกรณ์มหาวิทยาลัย



# บทที่ 7

## สรุป

### 7.1 สรุปผลการทดลอง

จากในบทที่ 6 ได้ทำการทดลองทั้งขั้นตอนวิธีที่ 1 และ 2 โดยขั้นตอนวิธีที่ 1 นั้นทำการทดลองเฉพาะในสภาพแวดล้อมจำลอง ส่วนขั้นตอนที่ 2 ทำการทดลองทั้งในสภาพแวดล้อมจริงและสภาพแวดล้อมจำลอง สามารถสรุปผลการทดลองได้ดังนี้

1. ผลการทดลองของขั้นตอนวิธีแบบที่ 1 แสดงให้เห็นว่าขั้นตอนวิธีนี้จะมีโอกาสให้ผลลัพธ์ที่ถูกต้องสูงเมื่อจำนวนหุ่นยนต์น้อยๆ (< 100 ตัว) และ โอกาสที่ผลลัพธ์จะถูกต้องนั้นขึ้นอยู่กับลักษณะของสภาพแวดล้อมด้วย
2. ผลการทดลอง ของ ขั้นตอนวิธีแบบที่ 2 ในกรณีที่ไม่มีค่าความคลาดเคลื่อนในการวัด ผลการทดลองสอดคล้องกับสมการที่ได้นำเสนอไป นั่นคือรูปแบบตำแหน่งของกลุ่มหุ่นยนต์ที่เป็นคำตอบ จะมีลักษณะข้อมูลเชิงมุมตรงกับลักษณะข้อมูลเชิงมุมที่แท้จริงของหุ่นยนต์แต่ละตัว และในกรณีที่มีความคลาดเคลื่อนในการวัด ค่าความคลาดเคลื่อนเฉลี่ยสำหรับแต่ละมุมก็จะมีค่าน้อยลงเรื่อยๆ เมื่อจำนวนหุ่นยนต์มากขึ้น วิธีการแก้ปัญหาแบบไต่ทางลาดนั้นให้ผลที่ดีกว่าทั้งทางด้านระยะเวลาที่ทำงานและค่าความคลาดเคลื่อน เนื่องจากไม่มีปัญหาเกี่ยวกับความละเอียดของทศนิยม

### 7.2 สรุปงานวิจัย

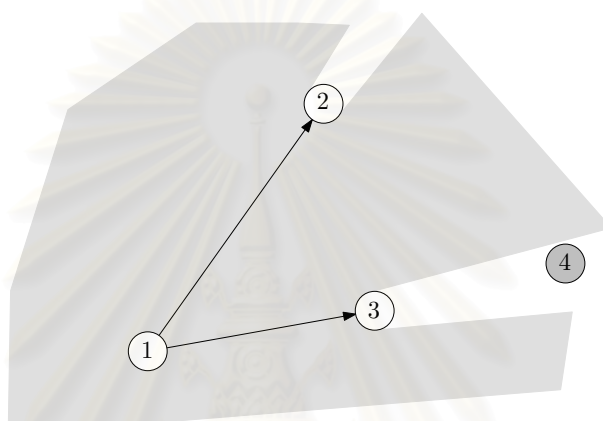
งานวิจัยชิ้นนี้นำเสนอขั้นตอนวิธีสองแบบที่ใช้ในการแก้ปัญหาการร่วมกันระบุตำแหน่งของกลุ่มหุ่นยนต์โดยใช้เพียงข้อมูลเชิงมุม ข้อมูลเชิงมุมนี้คือตำแหน่งเชิงมุมของหุ่นยนต์ตัวอื่นๆเทียบกับตัวเอง สภาพแวดล้อมของปัญหานี้คือ ระนาบ 2 มิติ และหุ่นยนต์มีลักษณะเป็นจุดไม่มีมิติทำให้ไม่เกิดการบังกัน

ขั้นตอนวิธีแบบที่ 1 อาศัยการนำข้อมูลทั้งหมดจากหุ่นยนต์สามตัวมาตัด แล้วจุดที่เป็นตำแหน่งที่แท้จริงของหุ่นยนต์คือจุดที่เกิดจากการตัดกันของข้อมูลจากหุ่นยนต์ทั้งสามตัว ขั้นตอนวิธีนี้ใช้ความซับซ้อนเชิงเวลาที่น้อยเท่ากับ  $O(N^2 \log(N))$  แต่ความถูกต้องของขั้นตอนวิธีจะขึ้นอยู่กับจำนวนหุ่นยนต์และลักษณะรูปร่างของสภาพแวดล้อม แต่ขั้นตอนวิธีนี้ไม่สามารถนำไปใช้งานในสภาพแวดล้อมจริงได้เนื่องจากไม่สามารถทนต่อความคลาดเคลื่อนในการวัดได้

ส่วนขั้นตอนวิธีแบบที่ 2 ได้เปลี่ยนรูปแบบของปัญหาให้อยู่ในรูปปัญหาการหาค่าเหมาะที่สุดเชิงคอนเวกซ์ จากรูปปัญหาที่ได้จัดรูปใหม่แล้วจะได้ว่าในกรณีที่ไม่มีค่าความคลาดเคลื่อนในการวัด คำตอบของปัญหานี้จะเป็นคำตอบที่ถูกต้องของปัญหาการร่วมกันระบุตำแหน่งโดยใช้ข้อมูลเชิงมุม และได้ทำการพิสูจน์โดยการทดลอง และได้ผลลัพธ์ที่ตรงตามที่กล่าวไว้ ปัญหานี้สามารถที่จะหาขั้นตอนวิธีเพื่อแก้ปัญหาก็ได้โดยมีความซับซ้อนเชิงเวลาในรูปพหุนาม

### 7.3 แนวทางการวิจัยในขั้นต่อไป

จากที่กล่าวไปในบทที่ 4 ว่าในวิทยานิพนธ์ฉบับนี้จะนำเสนอขั้นตอนวิธีในกรณีที่หุ่นยนต์ทุกตัวสามารถตรวจพบได้หมด ก็คือไม่มีการบังกันและไม่มีความบกพร่องในการตรวจหา ในการทดลองเบื้องต้นที่ได้ทำเพิ่มเติมโดยการปรับปรุงขั้นตอนวิธีการแก้ปัญหาแบบที่ 2 บางส่วน ได้ให้ผลลัพธ์ที่น่าสนใจคือขั้นตอนวิธีนี้มีโอกาสที่จะใช้งานกับกรณีที่มีการบังกันเกิดขึ้นได้ การปรับปรุงที่ได้ทำเพิ่มเติมนั่นคือมีการใส่รูปแบบการบังกันให้กับขั้นตอนการวัดข้อมูลเชิงมุม โดยมีการกำหนดให้หุ่นยนต์แต่ละตัวมีมิติ(ในสภาพแวดล้อม 2 มิติก็คือการมีพื้นที่) ในที่นี้จะกำหนดให้หุ่นยนต์มีลักษณะเป็นแผ่นวงกลม และหุ่นยนต์จะมองเห็นกันได้ก็ต่อเมื่อไม่มีหุ่นยนต์ตัวอื่นอยู่ในแนวการมองเห็น ดังรูป[7.1] หลังจากได้ทำการปรับปรุง



รูปที่ 7.1: แสดงรูปแบบการบังกันของหุ่นยนต์ จะเห็นว่าหุ่นยนต์หมายเลข 1 จะสามารถมองเห็นหุ่นยนต์หมายเลข 2 และ 3 ได้เท่านั้น

ขั้นตอนวิธีแล้วจึงทำการทดลองกับสภาพแวดล้อมจำลอง โดยในสภาพแวดล้อมจำลองนั้นก็ใส่รูปแบบการบังกันเช่นนี้เข้าไปด้วย ผลลัพธ์ที่ได้จากการทดลองคือมีโอกาสที่หุ่นยนต์แต่ละตัวจะสามารถเข้าสู่ตำแหน่งที่ต้องการได้ แต่ก็ยังมีบางกรณีที่ไม่สามารถเข้าสู่ตำแหน่งที่ต้องการได้ ฉะนั้นแนวทางในการวิจัยขั้นต่อไปก็คือการปรับหรือหาขั้นตอนวิธีที่ใช้งานได้แม้จะมีการบังกันเกิดขึ้น

ถัดมาในวิทยานิพนธ์นี้ได้นำเสนอขั้นตอนวิธีการแก้ปัญหาเฉพาะในกรณีที่มิติเท่ากับ 2 เท่านั้น แต่ขั้นตอนวิธีทั้งสองอย่างที่ได้นำเสนอไปนั้นสามารถที่จะทำงานกับสภาพแวดล้อมที่มี  $D$  มิติได้ เมื่อ  $D > 2$  สำหรับขั้นตอนการแก้ปัญหาแบบที่ 1 นั้นก็ทำได้โดยการหาจุดและปริภูมิย่อยที่เป็นเปลือกหุ้มเชิงคอนเวกซ์ของกลุ่มหุ่นยนต์ ซึ่งจะได้ว่าปริภูมิย่อยที่เป็นเปลือกหุ้มเชิงคอนเวกซ์นั้นจะมีทิศทางตั้งฉากที่ไม่ซ้ำกันยกเว้นในกรณีที่อยู่ในระนาบเดียวกัน หลังจากนั้นขั้นตอนการแก้ปัญหาก็จะมีลักษณะเช่นเดียวกับที่ได้เสนอไป ส่วนขั้นตอนการแก้ปัญหาแบบที่ 2 นั้น ก็จะสามารถปรับปรุงได้โดยไม่ต้องมีการปรับเปลี่ยนมากนัก เนื่องจากลักษณะของสมการที่แทนด้วยเวกเตอร์อยู่แล้ว สิ่งที่ต้องทำมีเพียงเพิ่มมิติของเวกเตอร์เข้าไปเท่านั้น

## รายการอ้างอิง

- [1] Sebastian Thrun, Maren Bennewitz, Wolfram Burgard, Armin B. Cremers, Frank Dellaert, and Dieter Fox. Minerva: a second-generation museum tour-guide robot. In IEEE International Conference on Robotics and Automation, 1999, 1999.
- [2] J. Evans, B. Krishnamurthy, B. Barrows, T. Skewis, and V. Lumelsky. Handling real-world motion planning: a hospital transport robot. IEEE Control Systems Magazine, 12(1992):15--19.
- [3] <http://www.irobot.com/>.
- [4] J.R. Spletzer and C.J. Taylor. A bounded uncertainty approach to multi-robot localization. In Intelligent Robots and Systems, 2003. (IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on, 2003.
- [5] Lieven Vandenberghe Stephen Boyd. Convex Optimization. Cambridge University Press, 2004.
- [6] R. Kurazume, S. Nagata, and S. Hirose. Cooperative positioning with multiple robots. In Robotics and Automation, 1994. Proceedings., 1994 IEEE International Conference on, 1994.
- [7] R. Kurazume, S. Hirose, S. Nagata, and N. Sashida. Study on cooperative positioning system (basic principle and measurement experiment). In Robotics and Automation, 1996. Proceedings., 1996 IEEE International Conference on, 1996.
- [8] R. Kurazume and S. Hirose. Study on cooperative positioning system: optimum moving strategies for cps-iii. In Robotics and Automation, 1998. Proceedings. 1998 IEEE International Conference on, 1998.
- [9] Dieter Fox, Wolfram Burgard, Hannes Kruppa, and Sebastian Thrun. Collaborative multi-robot localization. In in Proc. 23rd Annu. German Conf. Artif. Intell., 1999.
- [10] Dieter Fox, Wolfram Burgard, Hannes Kruppa, and Sebastian Thrun. A probabilistic approach to collaborative multi-robot localization. Autonomous Robot, 8(2000):325--344.
- [11] I.M. Rekleitis, G. Dudek, and E.E. Milios. Multi-robot collaboration for robust exploration. In Robotics and Automation, 2000. Proceedings. ICRA '00. IEEE International Conference on, 2000.
- [12] S.I. Roumeliotis and G.A. Bekey. Collective localization: a distributed kalman filter approach to localization of groups of mobile robots. In Robotics and Automation, 2000. Proceedings. ICRA '00. IEEE International Conference on, 2000.
- [13] Sebastian Thrun, Dieter Fox, and Wolfram Burgard. Monte carlo localization with mixture proposal distribution. In AAAI National. Conf. Artificial Intelligent, 2000.
- [14] J. Spletzer, A.K. Das, R. Fierro, C.J. Taylor, V. Kumar, and J.P. Ostrowski. Cooperative localization and control for multi-robot manipulation. In Intelligent Robots and Systems, 2001. Proceedings. 2001 IEEE/RSJ International Conference on Volume 2, 2001.

- [15] Renato Tinos, Luis E. Navarro-Serment, and Christiaan J.J. Paredis. Fault tolerant localization for teams of distributed robots. In Intelligent Robots and Systems, 2001. Proceedings. 2001 IEEE/RSJ International Conference on, 2001.
- [16] A. Howard, M.J. Matark, and G.S. Sukhatme. Localization for mobile robot teams using maximum likelihood estimation. In Intelligent Robots and System, 2002. IEEE/RSJ International Conference on, 2002.
- [17] I.M. Rekleitis, G. Dudek, and E.E. Milios. Multi-robot cooperative localization: a study of trade-offs between efficiency and accuracy. In Intelligent Robots and System, 2002. IEEE/RSJ International Conference on, 2002.
- [18] I. Roumeliotis Stergios and A. Bekey George. Distributed multirobot localization. IEEE TRANSACTIONS ON ROBOTICS AND AUTOMATION, 18(2002):781--795.
- [19] A. Howard, M.J. Mataric, and Sukhatme G.S. Putting the 'i' in 'team': an ego-centric approach to cooperative localization. In Robotics and Automation, 2003. Proceedings. ICRA '03. IEEE International Conference on, 2003.
- [20] W. Rynn, W.A. Malik, and Sooyong Lee. Sensor based localization for multiple mobile robots using virtual link. In Intelligent Robots and Systems, 2003. (IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on, 2003.
- [21] Jinsuck Kim and Nancy M. Amato. Complexity analysis and approximate solutions for two multiple-robot localization problems. In Robotics and Automation, 2004. Proceedings. ICRA '04. 2004 IEEE International Conference on, 2004.
- [22] Vincenzo Caglioti, Augusto Citterio, and Andrea Fossati. Cooperative, distributed localization in multi-robot systems: a minimum-entropy approach. In Distributed Intelligent Systems: Collective Intelligence and Its Applications, 2006. DIS 2006. IEEE Workshop on, 2006.
- [23] Yi Feng, Zhigang Zhu, and Jizhong Xiao. Heterogeneous multi-robot localization in unknown 3d space. In Intelligent Robots and Systems, 2006. (IROS 2006). Proceedings. 2006 IEEE/RSJ International Conference on, 2006.
- [24] Dimitrios Koutsounikolas, Saumitra M. Das, Y. Charlie Hu, Yung-Hsiang Lu, and C.S. George Lee. Cocoa: Coordinated cooperative localization for mobile multi-robot ad hoc networks. In Distributed Computing Systems Workshops, 2006. ICDCS Workshops 2006. 26th IEEE International Conference on, 2006.
- [25] A.I. Mourikis and S.I. Roumeliotis. Performance analysis of multirobot cooperative localization. Robotics, IEEE Transactions on [see also Robotics and Automation, IEEE Transactions on], 22(2006):666 -- 681.
- [26] Qining Wang, Lianghuan Liu, Guangming Xie, and Long Wang. Learning from human cognition: Collaborative localization for vision-based autonomous robots. In Intelligent Robots and

- Systems, 2006. (IROS 2006). Proceedings. 2006 IEEE/RSJ International Conference on, 2006.
- [27] Nirupama Bulusu, John Heidemann, and Deborah Estrin. Gps-less low cost outdoor localization for very small devices. Technical report, IEEE Personal Communications Magazine, 2000.
- [28] Nirupama Bulusu, Deborah Estrin, Lewis Girod, and John Heidemann. Density adaptive algorithms for beacon placement in wireless sensor networks. In Proceedings of the Sixth International Symposium on Communication Theory and Applications (ISCTA '01), 2001.
- [29] Lance Doherty, Kristofer Pister, and Laurent El Ghaoui. Convex position estimation in wireless sensor networks. In IEEE InfoCom 2001, 2001.
- [30] Andreas Savvides, ChihChieh Han, and Mani B. Strivastava. Dynamic finegrained localization in adhoc networks of sensors. In ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM), 2001.
- [31] Tian He, Chengdu Huang, Brian M. Blum, John A. Stankovic, and Tarek Abdelzaher. Range-free localization schemes for large scale sensor networks. In MobiCom, 2003.
- [32] Radhika Nagpal, Howard Shrobe, , and Jonathan Bachrach. Organizing a global coordinate system from local information on an ad hoc sensor network. In International Workshop on Information Processing in Sensor Networks (IPSN), 2003.
- [33] Dragos Niculescu and Badri Nath. Dv based positioning in ad hoc networks. Journal of Telecommunication Systems, 22:267--280, 2003.
- [34] Yi Shang, Wheeler Ruml, Ying Zhang, and Markus Fromherz. Localization from mere connectivity. In MobiHoc 2003, 2003.
- [35] Nayef A. Alsindi, Kaveh Pahlavan, Bardia Alavi, and Xinrong Li. A novel cooperative localization algorithm for indoor sensor networks. In Personal, Indoor and Mobile Radio Communications, 2006 IEEE 17th International Symposium on, 2006.
- [36] Nayef A. Alsindi, Kaveh Pahlavan, and Bardia Alavi. An error propagation aware algorithm for precise cooperative indoor localization. In Military Communications Conference, 2006. MILCOM 2006, 2006.
- [37] Kalle Astrom and Magnus Oskarsson. Solutions and ambiguities of the structure and motion problem for 1d retinal vision. Journal of Mathematical Imaging and Vision, 12(2000):121--135.
- [38] Kalle Astrom and Fredrik Kahl. Ambiguous configurations for the 1D structure and motion problem. In Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on, 2001.
- [39] H.W.L. Naus and C.V. van Wijk. Simultaneous localisation of multiple emitters. In Radar, Sonar and Navigation, IEE Proceedings -, 2004.



[40] <http://mathworld.wolfram.com/GradientDescent.html>.

[41] <http://mathworld.wolfram.com/SylvestersCriterion.html>.

[42] [sourceforge.net/projects/opencvlibrary](http://sourceforge.net/projects/opencvlibrary).



สถาบันวิทยบริการ  
จุฬาลงกรณ์มหาวิทยาลัย

## ประวัติผู้เขียนวิทยานิพนธ์

นายสรวิชัย ชนพานิชกุล เกิดเมื่อวันที่ 11 มกราคม 2527 ที่จังหวัดกรุงเทพมหานคร สำเร็จการศึกษาในระดับปริญญาวิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมคอมพิวเตอร์ จากคณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัยในปีการศึกษา 2547

มีความสนใจในงานทางด้าน การเขียนโปรแกรมคอมพิวเตอร์และงานประดิษฐ์ที่เกี่ยวข้องกับระบบอัตโนมัติ ผลงานที่เคยประดิษฐ์มาได้แก่ หุ่นยนต์บินได้ บ้านอัจฉริยะ หุ่นยนต์เตะฟุตบอล และผลงานทางด้านโปรแกรมได้แก่ ระบบการจัดการฐานข้อมูลคลังสินค้า ระบบคอมพิวเตอร์วิทัศน์สำหรับหุ่นยนต์เตะฟุตบอล จอพื้นผิวสัมผัสโดยใช้กล้อง ตัวควบคุมเครื่องนับเหรียญอัตโนมัติ และ ตัวควบคุมหุ่นยนต์เจียร์พลอย



สถาบันวิทยบริการ  
จุฬาลงกรณ์มหาวิทยาลัย