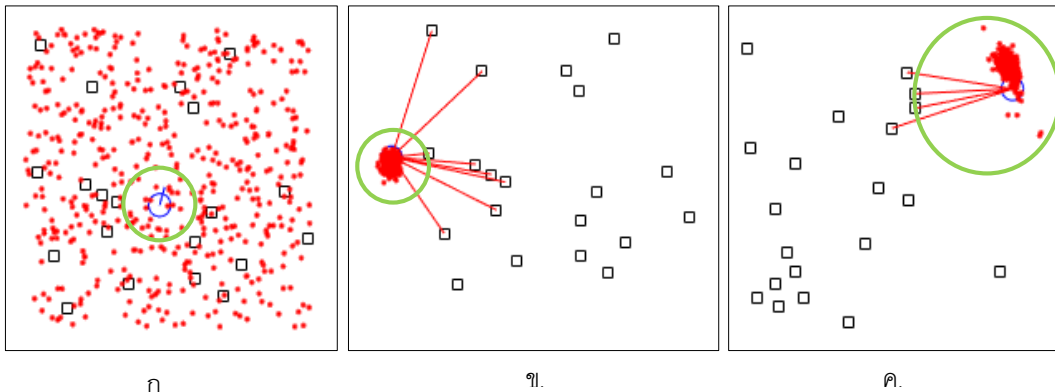


## Monte Carlo (Particle Filter) Localization

**คำสั่ง** จงดูตัวอย่าง Code Matlab ในไฟล์ main\_mcl.m ที่ให้สำหรับงาน Monte Carlo localization ในโปรแกรมดังกล่าวมีการสร้าง landmark หลายจุด โดยการสุ่ม โดยเราสมมติว่าเราทราบตำแหน่งของจุดเหล่านี้ทั้งหมด ตำแหน่งจริงคือตัวแปร  $X$  (สีน้ำเงิน) และตำแหน่งที่ทำการ Predict สร้างเป็นกลุ่ม Particle ที่ตำแหน่งแทนด้วย ตัวแปร  $belX$  ที่มี Weight กำกับตำแหน่งของ Particle ที่มีค่า Weight สูงสุดคือตำแหน่งที่เราทำนายในโปรแกรมหุ่นยนต์เริ่มต้นที่ตำแหน่งที่สุ่มขึ้นจุดหนึ่ง ( $x, y, \theta$ ) และค่าเริ่มต้นของการทำนาย  $belX$  ก็สุ่มขึ้นมาเช่นเดียวกัน เมื่อหุ่นยนต์เคลื่อนที่ไปเรื่อยๆ ก็จะมีการทำนายตำแหน่งใหม่ โดยมีวัตถุประสงค์เพื่อให้ตำแหน่งที่ทำนายมีค่าใกล้เคียงกับตำแหน่งจริงมากที่สุด จึงดำเนินการดังนี้

### 1. ทดลองรันโปรแกรม หลายๆ รอบรายงานผลที่สังเกตเห็น

จากโจทย์พบว่า เมื่อเริ่มต้น โปรแกรมจะทำการสุ่ม Landmark Particle และ ตำแหน่งเริ่มต้นของหุ่นยนต์ (ดังแสดงในรูปที่ 1 ก.) จากนั้น Particle จะเริ่มรวมตัวกันที่ตำแหน่งที่มีความน่าจะเป็นสูงสุดที่หุ่นยนต์จริงจะอยู่ (ดังแสดงในรูปที่ 1 ข.) แต่เมื่อหุ่นยนต์เคลื่อนที่ไกลจาก Landmark หรือเกินกว่าระยะ Sensor Range พบว่า Particle กระจายตัวออกจากตำแหน่งจริงของหุ่นยนต์ และทำให้การทำนายค่าผิดพลาด (ดังแสดงในรูปที่ 1 ค.)



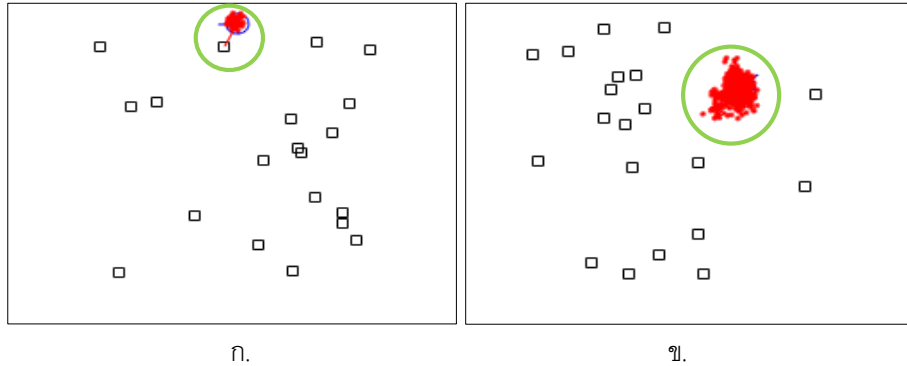
รูปที่ 1 แสดงการทดลองรันโปรแกรม

### 2. ทดลองปรับค่า Sensor Range ต่างๆ จาก 1, 2, 5, 10 แต่ละครั้งรันหลายๆ รอบ รายงานผลที่สังเกตเห็น

จากการทดลองปรับค่า Sensor Range พบว่า ค่าดังกล่าวส่งผลกระทบโดยตรงและโดยอ้อมกับการทำนายค่าของหุ่นยนต์ ในกรณีที่ Sensor Range น้อย พบว่าถ้าอยู่ใกล้กับ Landmark สามารถทำนายได้ดี (ดังแสดงในรูปที่ 2 ก.) แต่ในกรณีที่หุ่นยนต์อยู่ห่างจาก Landmark เกินระยะ Sensor Range จะส่งผลให้ Particle แยกตัวออกจากกันและทำนายผิด (ดังแสดงในรูปที่ 2 ข.) ดังนั้นการทดลองนี้แสดงให้เห็นได้ว่า การปรับ Sensor Range มี

## Monte Carlo (Particle Filter) Localization

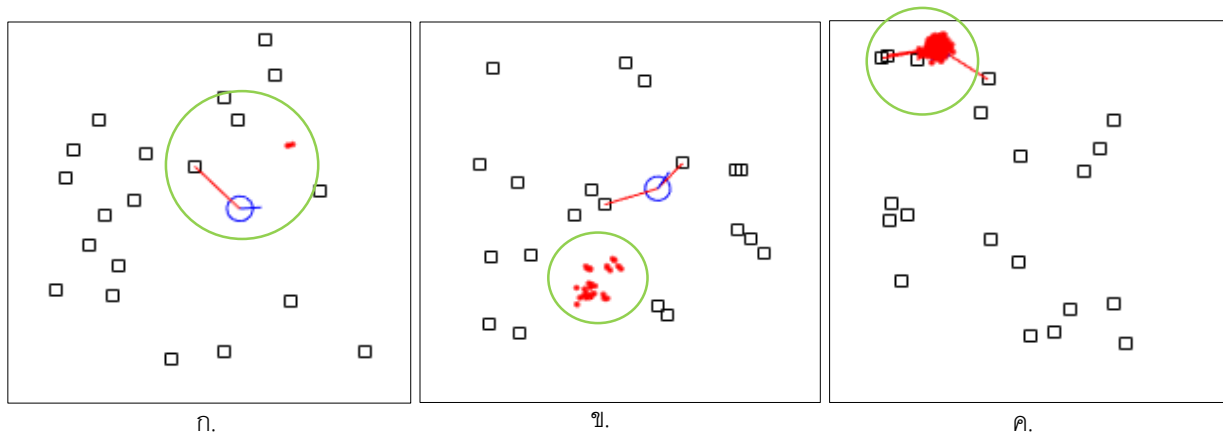
ความเกี่ยวเนื่องกับจำนวนของ Landmark และมีความสำคัญต่อการทำนายค่าตำแหน่งของหุ่นยนต์จริงให้ถูกต้องได้ ดังนั้นควรเลือกปรับ Sensor Range โดยพิจารณาจากจำนวนของ Landmark



รูปที่ 2 แสดงผลกระทบจากการปรับ Sensor range ต่อการทำนายค่าของหุ่นยนต์

### 3. ทดลองปรับค่าจำนวน Particle (น้อย ปานกลาง มาก) รายงานผลที่สังเกตเห็น

จากการทดลองถ้ากำหนดให้มี Sensor Range เป็น 2 กำหนด Particle เป็น 2 จะไม่สามารถทำนายค่าได้เลย (ดังแสดงในรูปที่ 3 ก.) จากนั้นถ้าเพิ่ม Particle จาก 2 เป็น 50 พบว่ายังไม่สามารถทำนายได้ (ดังแสดงในรูปที่ 3 ข.) ต่อมาทดลองเพิ่ม Particle จาก 50 เป็น 500 พบว่าสามารถทำนายได้ (ดังแสดงในรูปที่ 3 ค.)

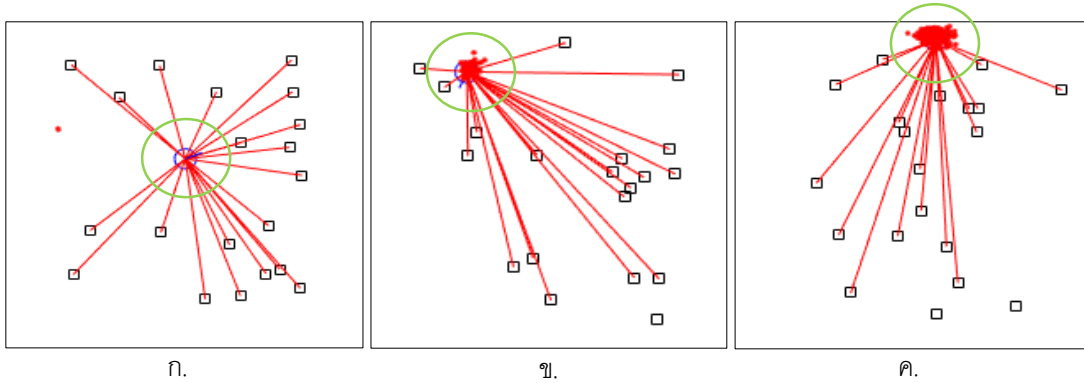


รูปที่ 3 แสดงผลกระทบจากการปรับ Particle ต่อการทำนายค่าของหุ่นยนต์

จากข้างต้นทำให้เกิดข้อสงสัยว่าการปรับ Particle จะมีความเกี่ยวเนื่องกับการปรับ Sensor Range หรือไม่ การทดลองพบว่าที่ Sensor Range สูงเท่าไรก็ตามถ้า Particle น้อยมากๆ (การทดลองนี้ Particle เป็น 2) จะไม่สามารถทำนายได้ (ดังแสดงในรูปที่ 4 ก) แต่ถ้ามี Particle มากขึ้นพบว่า Sensor Range จะมีผลกระทบมากขึ้น

## Monte Carlo (Particle Filter) Localization

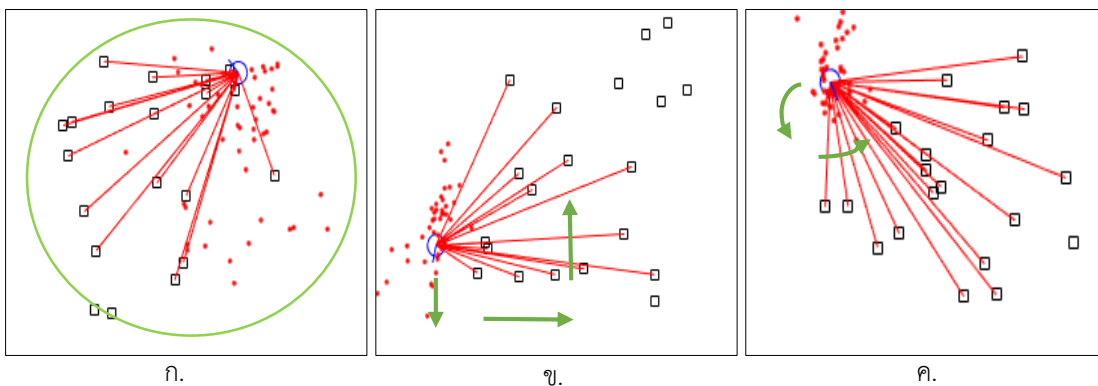
คือ สามารถทำนายได้ถูกต้อง (ดังแสดงในรูปที่ 4 ข. และ ค.) จาก (รูปที่ 3 ข.) ใช้จำนวน Particle เท่ากันกับ (รูปที่ 4 ข.) แต่เนื่องจาก (รูปที่ 3 ข.) นั้น Sensor Range น้อยกว่าจึงไม่สามารถทำนายได้ถูกต้อง



รูปที่ 4 แสดงผลกระทบจากการปรับ Particle และ sensor range ต่อการทำนายค่าของหุ่นยนต์

#### 4. ทดลองปรับค่า Covariance ของ Sensor ( $\text{sigSensor}$ ) และของตำแหน่งหุ่นยนต์ $\text{sigXY}$ $\text{sigTH}$ รายงานผลที่สังเกตเห็น

จากการทดลองเพิ่มค่า  $\text{sigSensor}$  พบว่าส่งผลให้ Particle เกิดปั่นป่วน (ดังแสดงในรูปที่ 5 ก.) และไม่สามารถทำนายได้ เนื่องจาก  $\text{sigSensor}$  คือการใส่ค่า สัญญาณรบกวนให้กับ Sensor ต่อมาได้ทำการเพิ่มค่าตัวแปร  $\text{sigXY}$  และ  $\text{sigTH}$  พบว่าเมื่อทำการเพิ่ม  $\text{sigXY}$  ทำให้การเคลื่อนที่ในแนว XY มีสัญญาณรบกวน ทำให้เกิดการสั่นไถล (ดังแสดงในรูปที่ 5 ข.) และ  $\text{sigTH}$  สร้างสัญญาณรบกวนในการเลี้ยวเช่น เกิดการสั่นไถลขณะเลี้ยว (ดังแสดงในรูปที่ 5 ค.) จากการทดลองทำให้เข้าใจได้ว่าตัวแปรดังกล่าวมาข้างต้นนี้ ใช้ทดสอบว่าขณะทำงานจริง (หุ่นยนต์ที่ใช้ทำงานจริงในธรรมชาติ) ถ้าเกิดเหตุการณ์ดังกล่าวขึ้นโปรแกรมจะสามารถทำงานได้หรือไม่

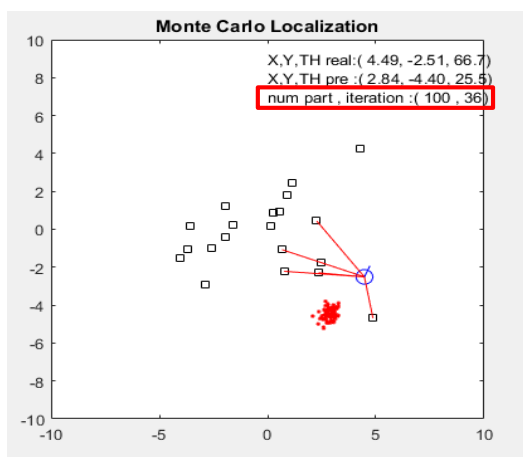


รูปที่ 5 แสดงผลกระทบจากการเพิ่มสัญญาณรบกวนให้ระบบ (การสั่นไถล)

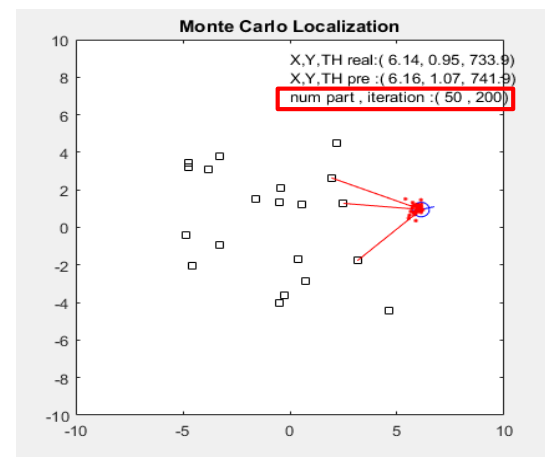
## Monte Carlo (Particle Filter) Localization

5. จงศึกษาโปรแกรมให้ละเอียดและปรับแก้โปรแกรมให้จำนวน Particle มีค่าลดลง เมื่อรันจำนวน Step ครบ 100 ครั้ง โดยมีวัตถุประสงค์เพื่อลดการคำนวณแต่ยังคงการทำนายตำแหน่งที่แม่นยำไว้ได้ หากสะดวกกว่านักศึกษาสามารถเขียนโปรแกรมของตัวเองขึ้นมาใหม่ได้ทั้งหมด

จากโจทย์เมื่อโปรแกรมเริ่มทำงานได้มีการกำหนดจำนวน Particle เป็น 100 ตัว (ดังแสดงในรูปที่ 6 ก.) จากนั้นเมื่อการทำนายมีความถูกต้อง โปรแกรมจะลดจำนวน Particle ให้เหลือ 50 ตัว (ดังแสดงในรูปที่ 6 ข.) โดยมีวัตถุประสงค์เพื่อลดการคำนวณแต่ยังคงการทำนายตำแหน่งที่แม่นยำไว้ได้ สามารถแสดงการทำงานได้ดังไฟล์แนบ “main\_mcl\_rev.m”



ก.



ข.

รูปที่ 6 แสดงการปรับจำนวน Particle มีค่าลดลง เมื่อรันจำนวน Step ครบ 100 ครั้ง