**คำสั่ง**  จงดูตัวอย่าง Code Matlab ในไฟล์ main\_ekf\_find\_mistake.m ที่ให้สำหรับงาน EKF localization ใน โปรแกรมดังกล่าวมีการสร้าง landmark หลายจุด โดยการสุ่ม โดยเราสมมุติว่าเราทราบตำแหน่งของจุดเหล่านี้ ทั้งหมด ตำแหน่งจริงคือตัวแปร X (สีน้ำเงิน) และตำแหน่งที่ทำการ Predict คือตัวแปร mu (สีแดง) ในโปรแกรม หุ่นยนต์เริ่มต้นที่ตำแหน่งที่สุ่มขึ้นจุดหนึ่ง (x, y, th) และค่าเริ่มต้นของการทำนาย mu ก็สุ่มขึ้นมาเช่นเดียวกัน เมื่อหุ่นยนต์เคลื่อนที่ไปเรื่อยๆ ก็จะทำการทำนายตำแหน่งใหม่ตาม landmark ที่ตรวจพบ โดยมีวัตถุประสงค์ เพื่อให้ตำแหน่งที่ทำนายมีค่าใกล้เคียงกับตำแหน่งจริงมากที่สุด จงดำเนินการดังนี้

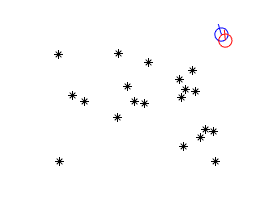
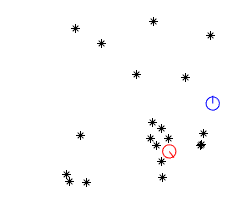
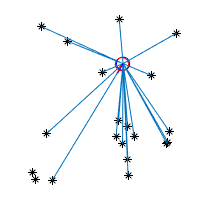
**คำตอบ**

**1. ทดลอง รันโปรแกรม หลายๆ รอบ รายงานผลที่สังเกตุเห็น**

จากการทดลองรันโปรแกรมหลายๆ รอบ พบว่าเมื่อเริ่มต้นโปรแกรม หุ่นยนต์จริงสีน้ำเงิน และหุ่นยนต์ที่ได้จากการประมาณค่าความน่าจะเป็นสีแดงอยู่ห่างกันตามที่กำหนดไว้ในโปรแกรม (ดังแสดงในรูปที่ 1 ก.) เมื่อระยะเวลาผ่านไปตัวสีแดงจะขยับเข้าใกล้ตัวสีน้ำเงิน (ดังแสดงในรูปที่ 1 ข.) และในกรณีที่หุ่นยนต์ทั้งสองเคลื่อนที่ห่างออกจากจุด Landmark พบว่าหุ่นยนต์ทั้ง 2 เคลื่อนที่ห่างออกจากกันเอง (ดังแสดงในรูปที่ 1 ค.) จนกระทั่งเคลื่อนที่เข้าใกล้ Landmark หุ่นยนต์ก็เข้าใกล้กันอีกครั้ง ซึ่งโดยหลักการแล้ว การทำงานของโปรแกรมถูกต้อง แต่ในบางครั้งสังเกตุเห็นได้ว่า บางจุดขณะเกิดการเลี้ยวตำแหน่งของหุ่นยนต์ตัวที่ทำนายค่าหลุดออกมาจากหุ่นยนต์ตัวจริง (ดังแสดงในรูปที่ 2 ข.) ทั้งที่มีการลู่เข้าตำแหน่งจริงแล้ว (ดังแสดงในรูปที่ 2 ก.) ตามหลักการไม่น่าเป็นไปได้

จุดสังเกตุที่สำคัญ

- โดยมากแล้วหุ่นยนต์กระโดดออกขณะเลี้ยว ดังนั้นต้องพิจารณาส่วนที่เกี่ยวกับการการเลี้ยวเป็นกรณีพิเศษ



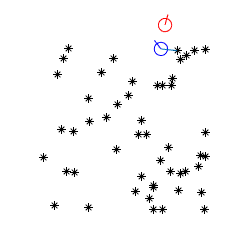
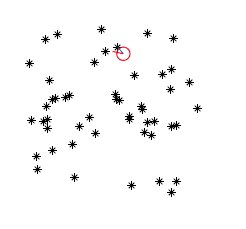
ค

ข

ก

รูปที่ 1 แสดงการรันโปรแกรมและสังเกตุผลลัพธ์ที่เกิดขึ้น

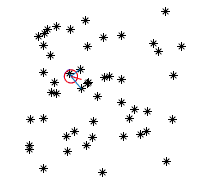
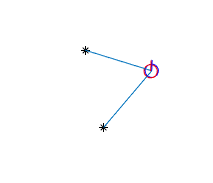
ข



ก

รูปที่ 2 แสดงการลู่เข้าค่าจริง หลังจากนั้นลู่ออก

**2. ทดลองปรับค่า Sensor range ต่างๆ จาก 1, 2, 5, 10 แต่ละครั้งรันหลายๆ รอบ รายงานผลที่สังเกตเห็น**  
 จากการทดลองพบว่า ที่ Sensor range น้อยๆ จะสามารถทำงานได้ดีในกรณีที่ landmark อยู่ติดกันและอยู่ใกล้หุ่นยนต์เท่านั้น (ดังแสดงในรูปที่ 3 ก) กรณีนอกเหนือจากนั้นมีการทำนายผิดพลาดมากขึ้น และเมื่อเพิ่มค่า Sensor range สามารถทำงานได้ดีถึงแม้ว่า landmark จะมีน้อยก็ตาม (ดังแสดงในรูปที่ 3 ข.) แต่ในสถาณการจริงนั้นถ้า Sensor range มีค่ามากราคาของเซนเซอร์จะค่อนข้างสูง ดังนั้นการเพิ่มจำนวน landmark เป็นเรื่องที่น่าสนใจ



ข.

ก.

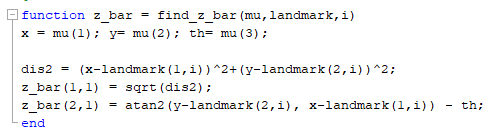
รูปที่ 3 แสดงการปรับ Sensor range และ Landmark

**3. ทดลองปรับค่า Parameter อื่นๆ รายงานผลที่สังเกตเห็น**

จากโปรแกรมข้างต้นนั้น Parameter ที่สำคัญแลมีผลกระทบอย่างมากกับการทำนายค่าคือ Sensor range และ landmark จากการทดลองปรับจำนวนของ landmark พบว่ามีความสอดคล้องกับ Sensor range เนื่องจากถ้า landmark เยอะสามารถลด Sensor range ได้ หรือถ้าต้องการลด landmark ก็สามารถทำโดยการเพิ่ม Sensor range ได้ในทำนองเดียวกัน

**4. ในโปรแกรมนั้นยังมีข้อผิดพลาดอยู่บางจุด ถ้าสังเกตจากการปรับค่าพารามิเตอร์ตามรายการด้านบนและทำการ ทดลองซ้ำๆ จะเห็นว่าบางครั้งตำแหน่งที่ทำนายได้ของหุ่นยนต์มีการกระโดดไปมาหลังจากที่ค่าทำนายลู่เข้าไปแล้ว ซึ่งไม่ควรจะเกิดขึ้น จงศึกษาโปรแกรมให้ละเอียดและปรับแก้โปรแกรมเพื่อแก้ปัญหาดังกล่าว หากสะดวกกว่า นักศึกษาสามารถเขียนโปรแกรมของตัวเองขึ้นมาใหม่ได้ทั้งหมด**

จากการศึกษาพบว่า ปัญหาที่เกิด อยู่ในขั้นตอนการหา error\_z เพื่อนำไปอัพเดทในการทำนาย (ดังแสดงในรูปที่ 4 ก.) ซึ่งต้องทราบค่า z\_bar ที่หาได้จาก function (ดังแสดงในรูปที่ 4 ข.) แต่เนื่องจากใน atan2 ใน Matlab สามารถใช้งานได้ในช่วง -pi ถึง pi เท่านั้น เมื่อค่าที่ส่งเข้ามาไม่อยู่ในช่วงที่กำหนด ทำให้การคำนวณค่าผิดพลาด จึงเกิดปัญหาดังที่กล่าวมาข้างต้น

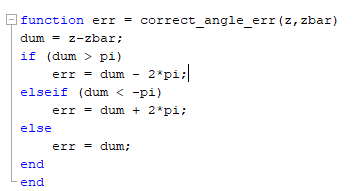


ก.

ข.

รูปที่ 4 function หา error\_z

ดังนั้นการแก้ปัญหาควรเขียน function เพื่อมาป้างกันในกรณีที่ err\_z เกิน pi และน้อยกว่า -pi (ดังแสดงในรูปที่ 5 ก.) และแทนการคำนวณหาค่า err\_z (ดังแสดงในรูปที่ 4 ก.) ด้วย function ที่สร้างขึ้นมาแก้ปัญหานี้ (ดังแสดงในรูปที่ 5 ข.) โปรแกรมที่ได้รับการแก้ปัญหา แสดงดัง ไฟล์แนบ (main\_ekf\_find\_mistake.m)



ก.

ข.

รูปที่ 5 function แก้ err\_z ที่เกิน pi และน้อยกว่า –pi