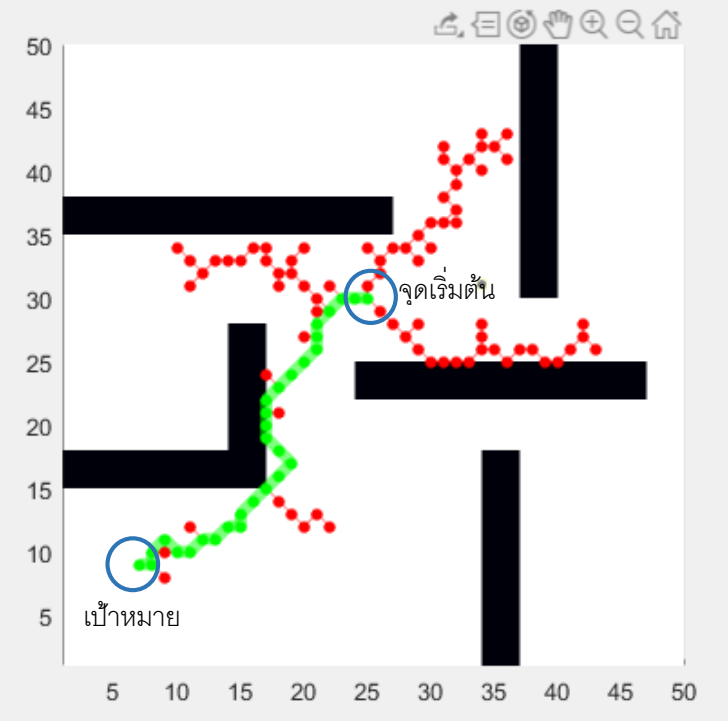
Path Planning – RRT & RRT*

จงดูตัวอย่าง code Matlab ในไฟล์ main_rrt_no_obstacle.m ที่ให้และดำเนินการดังต่อไปนี้

1. ปรับโปรแกรม RRT ให้สามารถรองรับการวางแผนเส้นทางที่หลบเลี่ยงสิ่งกีดขวางแบบ Static

การสร้าง Path planning คือการวางแผนการเคลื่อนที่ของหุ่นยนต์จากจุดเริ่มต้นไปที่เป้าหมายอย่างถูกต้อง โดยที่ไม่ชนสิ่งกีดขวางแบบ Static จากโจทย์ใช้หลักการ RRT คือ ทดลองสุ่มจุดรอบๆที่อยู่ในแผนที่ที่รู้พิกัด x,y ทั้งหมด แล้วเชื่อมโยงจุดที่อยู่ใกล้กันจนกระทั่งถึงเป้าหมาย จากนั้นผู้เขียนได้ทดลองเขียนโปรแกรมขึ้นมาใหม่ โดยอาศัยหลักการพื้นฐานตามที่ให้มาดังไฟล์แนบ “main_rrt_no_obstacle.m” โดยทำการเพิ่มส่วนที่ช่วยป้องกันชนสิ่งกีดขวาง ดังแสดงในรูปที่ 1 สามารถแสดงการทำงานของโปรแกรมได้ดังวีดีโอแนบ“Path_planning_rrt.mov” และดังไฟล์แนบ “Path_planning_rrt.m”



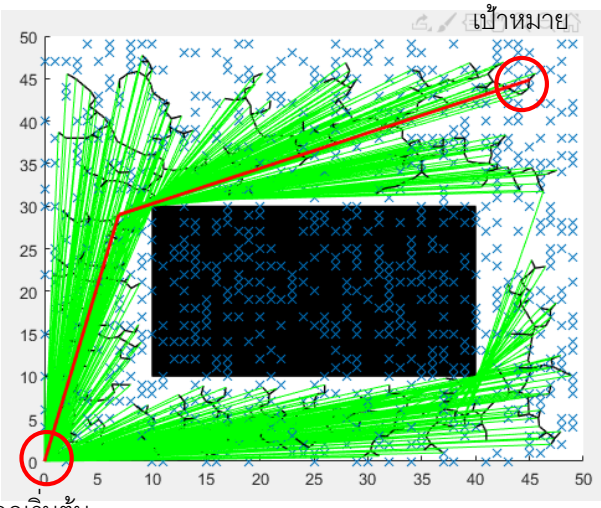
รูปที่ 1 แสดงการทำงานของ path planning โดยใช้ rrt

สรุปผลการทดลองเขียนโปรแกรมพบว่าการใช้ RRT เกิดปัญหา คือต้องใช้เวลามากในการประมวณผล แล้วเส้นทางการเคลื่อนที่โค้งไปมา เมื่อนำไปใช้งานจริงจะเกิดปัญหาในการควบคุมหุ่นยนต์ได้ ดังแสดงในรูปที่ 1 เส้นสีเขียว

Path Planning – RRT & RRT*

2. เขียนโปรแกรม RRT* เพื่อการวางแผนเส้นทางที่สามารถรองรับการหลบเลี่ยงสิ่งกีดขวางแบบ static ส่งผลการรันโปรแกรม (capture หน้าจอ + video) ทั้งกรณีไม่มีและมีสิ่งกีดขวาง รวมทั้งไฟล์โปรแกรมทั้งหมด

จากโจทย์ RRT* เป็นการพัฒนามาจาก RRT โดยมาแก้ปัญหาเรื่องการเคลื่อนที่ของ RRT ในเรื่องของการโค้งไปมาเยอะเกินความจำเป็น โดยใช้หลักการ ตัวที่สุ่มค่ามาทุกตัวต้องมีการเทียบกับจุดเริ่มต้น เพื่อเมื่อมีการสุ่มได้ทางเลือกที่ดีกว่า ใกล้กว่าก็สามารถตัดสินใจเปลี่ยนทิศทางได้ จึงทำให้การเคลื่อนที่เข้าใกล้เส้นตรงมากกว่าแบบ RRT ปกติ ดังแสดงในรูปที่ 2



รูปที่ 2 แสดงการทำงานของ path planning โดยใช้ rrt*

สรุปผลการทดลองเขียนโปรแกรมพบว่าการใช้ RRT* สามารถหาทิศทางที่ใกล้ที่สุดได้ เส้นทางการเคลื่อนที่ไม่มีความซับซ้อนเท่า RRT สามารถแสดงการทำงานของโปรแกรมได้ดังวีดีโอแนบ“Path_planning_rrt_s.mov” และดังไฟล์แนบ “Path_planning_rrt_s.m”