

รายงานฉบับสมบูรณ์

โครงการวิจัยร่วมภาครัฐและเอกชน
ระบบผลิตอัตโนมัติเพื่อการพัฒนาอุตสาหกรรมการผลิตขั้นสูง

เรื่อง

Cobots: Collaborative Robots

โดย

ดร. วิทยา วัฒนสุโกประสิทธิ์

ภาควิชาวิศวกรรมเครื่องกล
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย

กรกฎาคม 2548

Inertial Measurement Unit for Mobile Robot

บทคัดย่อ

Inertial Measurement Unit เป็นสิ่งจำเป็นในการวัดสัญญาณการควบคุมระดับสูง สำหรับหุ่นยนต์เคลื่อนที่ งานวิจัยนี้เป็นการออกแบบและพัฒนา Inertial Measurement Unit เพื่อต้องการหา

มุมและความเร็วในการเคลื่อนที่ของหุ่นยนต์เคลื่อนที่ ที่มีสามองศาอิสระ หลักการโดยย่อคือการนำ Rate Gyroscope และ Accelerometer สำหรับวัดอัตราการหมุน และความเร่งในทั้งสามแกน และกรอปกับการใช้สมการ Forward Jacobian และ Inverse Jacobian ของหุ่นยนต์ดังกล่าวเพื่อทำการวัดความเร็วของหุ่นยนต์โดยอาศัยการวัดจากล้อ (odometry) และนำข้อมูลทั้งหมดเข้าสู่ Kalman Filter ทำการประมาณหา Bias ของ Gyro และ Accelerometer เพื่อหาความเร็วและมุมของหุ่นยนต์ที่ต้องการ

จากการทดลอง พบว่าการใช้ Gyroscope และ Accelerometer ร่วมกับ Kalman Filter สามารถที่จะหาความเร็วและมุมของหุ่นยนต์ โดยมีผิดพลาดน้อยกว่าการหาความเร็วและมุมจากการหมุนของล้อแต่เพียงอย่างเดียว เมื่อเปรียบเทียบข้อมูลทั้งสองกับข้อมูลที่ได้จากระบบ Machine Vision ที่วัดตำแหน่งของการเคลื่อนที่ของหุ่นยนต์ โดยการทดลองครั้งนี้ได้ทดสอบบนหุ่นยนต์เคลื่อนที่ ที่ใช้ล้อชนิด omnidirectional ซึ่งมีการ slip สูงทำให้การวัดข้อมูลจากล้อเพียงอย่างเดียวเกิดความผิดพลาดได้ง่าย

สารบัญ

บทที่ 1 : Introduction	1
1.1 ที่มาและความสำคัญ	1
1.2 วัตถุประสงค์	1
1.3 ขอบเขต	2
1.4 ประโยชน์ที่คาดว่าจะได้รับ	2
1.5 ขั้นตอนการดำเนินงาน	2
บทที่ 2 : Related Theory and Reviews	3
2.1 Measurement for Controlling Mobile Robot	3
2.2 Fundamental of Mobile Robot Measurement	3
2.3 Inertial Navigation System	6
2.4 Kalman Filter	8
บทที่ 3 : Inertial Measurement Unit	10
3.1 หน่วยงานที่จะทำการศึกษาและพัฒนา	10
3.2 การคำนวณการเคลื่อนที่ของล้อ Omni-Direction	11
3.3 การทดสอบการเคลื่อนที่แบบ Open-Loop	12
3.4 การผนวก Inertial Measurement Unit	15
3.5 การพัฒนา Inertial Measurement Unit	16
3.6 การเลือก Gyro และ Accelerometer	16
3.7 การออกแบบตัวกรองสัญญาณความถี่ Low Pass Filter	20
3.8 การทดลองวัดค่า Gyroscope และ Accelerometer ขณะหยุดนิ่ง	25
3.9 การออกแบบและใช้งาน Kalman Filter	26
บทที่ 4 ข้อสรุปและข้อเสนอแนะ	34
Reference	35

บทที่ 1

บทนำ

1.1 ที่มาและความสำคัญ

ในปัจจุบันการควบคุมแบบอัตโนมัติ ได้เข้ามามีบทบาทเป็นอย่างยิ่งในสังคม โดยเฉพาะอย่างยิ่งในทางด้านของเทคโนโลยีของหุ่นยนต์และระบบอัตโนมัติ อย่างไรก็ตาม ในการทำงานที่จะสามารถควบคุมระบบที่สนใจนั้น จำเป็นที่จะต้องมีการตรวจวัดสถานะต่างๆ ของระบบ เพื่อที่จะสามารถนำมาชดเชยความผิดพลาดที่เกิดขึ้นได้ ซึ่งการตรวจวัดนี้บางครั้งอาจทำได้ง่าย เช่น การวัดอุณหภูมิด้วยเทอร์มิสเตอร์ การวัดความดันด้วยเซ็นเซอร์ตรวจวัดความดัน เป็นต้น แต่ในบางกรณีนั้นอาจทำได้ยาก เช่น การวัดท่าทางในการบินของเครื่องบิน การวัดการเคลื่อนที่ของหุ่นยนต์เคลื่อนที่ได้บางรูปแบบ เป็นต้น ทั้งนี้เนื่องจาก เป็นการวัดที่ไม่สามารถเปรียบเทียบกับสิ่งอ้างอิงภายนอกได้ง่ายนัก เช่น ในกรณีของเครื่องบินที่กำลังบินอยู่นั้น เราไม่สามารถนำตัวตรวจวัดมุมด้านหนึ่งติดกับตัวเครื่องบินอีกด้านเทียบกับผิวโลกเพื่อหามุมได้ จึงได้มีการคิดค้นตัวตรวจวัดที่จะทำการวัดได้ โดยอ้างอิงจากแกนอ้างอิงเฉื่อย (Inertia Frame) ซึ่งในโครงงานนี้จะได้ทำการทดลองใช้งานตัวตรวจวัดประเภทนี้ ซึ่งก็คือ Rate Gyroscope และ Accelerometer นำมาประยุกต์เข้ากับหุ่นยนต์เคลื่อนที่ได้ของทีม Plasma-Z จากจุฬาลงกรณ์มหาวิทยาลัย ซึ่งใช้ในการแข่งขันฟุตบอลหุ่นยนต์ในระดับชาติและนานาชาติ เนื่องจากตัวหุ่นยนต์มีความคล่องตัวตลอดจนสมรรถนะสูงมาก เนื่องจากใช้ล้อที่มีองศาอิสระ (Degree of freedoms :DOFs) เท่ากับ 3 หรือที่เรียกว่า Omni-Directional Wheel จนทำให้การควบคุมโดยปกติเป็นไปได้ยาก การนำตัวตรวจวัดมาประยุกต์ใช้จึงมีผลให้หุ่นยนต์สามารถทราบถึงการเคลื่อนที่ของหุ่นยนต์ในเวลา นั้นๆ ซึ่งจะเป็นการเพิ่มความสามารถในการควบคุมให้ดียิ่งขึ้น โดยในขั้นตอนนี้จะมุ่งเน้นที่การหาความเร็วและมุมของตัวหุ่นยนต์กับแกนอ้างอิงให้ได้ถูกต้องมากที่สุด และจึงค่อยนำข้อมูลที่ได้นี้มาใช้ในการควบคุมต่อไป

1.2 วัตถุประสงค์

1. ศึกษาการวัดตำแหน่งและความเร็วของหุ่นยนต์เคลื่อนที่ได้ด้วยด้วยวิธีการต่างๆ เช่น Dead-Reckoning , Rate Gyros, Accelerometers, Machine Visions เป็นต้น
2. เพื่อศึกษาถึงทฤษฎีพื้นฐานของหุ่นยนต์เบื้องต้น (Fundamental of Robotics) และ การควบคุมระบบอัตโนมัติ (Automatics Controls)
3. ศึกษาและพัฒนา Inertial Measurement Unit และเทคนิคในการการชดเชยความผิดพลาดและการผสมผสานข้อมูลจากการวัดแบบต่างๆ

1.3 ขอบเขตของโครงการ

1. เพื่อให้ได้ถึง Inertial Navigation System อันประกอบด้วย Inertial Measurement Unit ที่ติดตั้งอยู่บนหุ่นยนต์เคลื่อนที่ได้ ตลอดจนโปรแกรมคอมพิวเตอร์ที่จะสามารถวัดการเคลื่อนที่ของหุ่นยนต์ที่สนใจได้อย่างถูกต้อง

1.4 ขั้นตอนการดำเนินการ

1. ศึกษาถึงหลักการการทำงานของ Inertial Navigation System ตลอดจนงานวิจัยที่มีมา
2. ศึกษาถึงการทำงานของ Kalman Filter และการประยุกต์ใช้
3. ออกแบบ Inertial Measurement Unit และทดลองติดตั้ง
4. ทดลองเก็บค่าจากการวัด และนำผลที่ได้นำมาวิเคราะห์บนคอมพิวเตอร์
5. พัฒนาเทคนิคในการคำนวณและประยุกต์ใช้ Kalman Filter กับการวัดในข้อ 4
6. พัฒนาโปรแกรมบนหุ่นยนต์ โดยใช้เทคนิคและกระบวนการที่ได้จากข้อ 5 ลงบนตัวหุ่นยนต์
7. ทดสอบการใช้งานจริงและเก็บผลการคำนวณของหุ่นยนต์
8. วิเคราะห์และประเมินผล

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1. ได้ Inertial Navigation System ที่จะให้ข้อมูลในการเคลื่อนที่ของหุ่นยนต์ เพื่อใช้ในการควบคุมหุ่นยนต์เคลื่อนที่ได้ ให้มีความถูกต้องและแม่นยำในการเคลื่อนที่ที่มากขึ้น
2. ได้เทคนิคในการประยุกต์ใช้ Kalman Filter เพื่อที่จะสามารถนำไปประยุกต์ใช้กับระบบที่มีองศาอิสระมากขึ้น เช่น อากาศยานไร้คนขับ เป็นต้น

บทที่ 2

ทฤษฎีที่เกี่ยวข้องกับโครงการ

2.1 การวัดในการควบคุมหุ่นยนต์เคลื่อนที่ได้

ในการควบคุมหุ่นยนต์เคลื่อนที่ได้นั้น ก่อนที่จะสามารถควบคุมได้นั้น จำเป็นที่จะต้องมีการวัดความเป็นไปของระบบที่ควบคุม เพื่อนำมาชดเชยความผิดพลาดที่เกิดขึ้น ซึ่งในกรณีของหุ่นยนต์เคลื่อนที่ได้นั้น การวัดในที่นี้คือ การวัดตำแหน่ง ความเร็ว มุมของหุ่นยนต์ เป็นต้น เพื่อใช้ในการควบคุมหุ่นยนต์ให้เคลื่อนที่ตามเส้นทางที่ต้องการได้อย่างถูกต้องแม่นยำ

2.2 พื้นฐานการวัดการเคลื่อนที่ของหุ่นยนต์เคลื่อนที่ได้

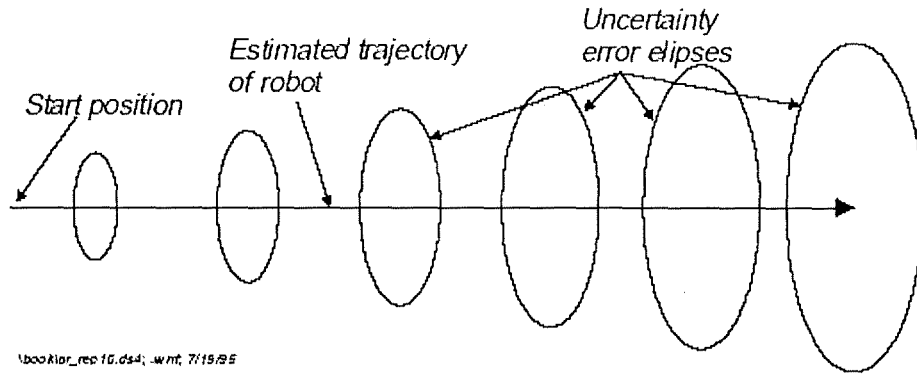
การวัดการเคลื่อนที่ของหุ่นยนต์เคลื่อนที่ได้ในปัจจุบันนั้น มีวิธีการวัดที่ได้คิดค้นขึ้นมาเป็นจำนวนมาก ซึ่งในบางวิธีจะได้นำมาประยุกต์ใช้ในโครงการนี้ด้วย วิธีเหล่านี้ต่างมีข้อดีและข้อเสียต่างกันไป โดยการวัดในที่นี้ได้แก่

2.2.1 Dead Reckoning Method

วิธีนี้เป็นวิธีที่ใช้กันมากในปัจจุบัน โดยการใช้การวัดการหมุนหรือการบิดของล้อ ทำให้รู้ความเร็วและทิศทาง เพื่อนำมาคำนวณถึงการเคลื่อนที่ของหุ่นยนต์เทียบกับการวัดที่ผ่าน มา ทำให้ได้ตำแหน่งโดยประมาณของหุ่นยนต์ ความผิดพลาดในการวัดแบบนี้อาจเกิดจากความผิดพลาดต่างๆ เช่น

- Systematic Error
 - ขนาดของล้อที่ไม่สม่ำเสมอ
 - มุมของล้อที่ไม่ตรงกับที่คำนวณไว้
 - ระยะห่างของล้อ
 - ความถี่ในการวัด (Sampling Rate) ที่ไม่เพียงพอ
- Non Systematic Error
 - พื้นที่หุ่นยนต์วิ่งไม่ราบเสมอกัน
 - เกิดการไถลขึ้น (Sliping)
 - หุ่นยนต์มีความเร่งหรือความหน่วงมากเกินไป (Over acceleration / Over deceleration)
 - มีแรงกระทำจากภายนอกกระทำกับตัวหุ่น เช่น การชน กระแทก

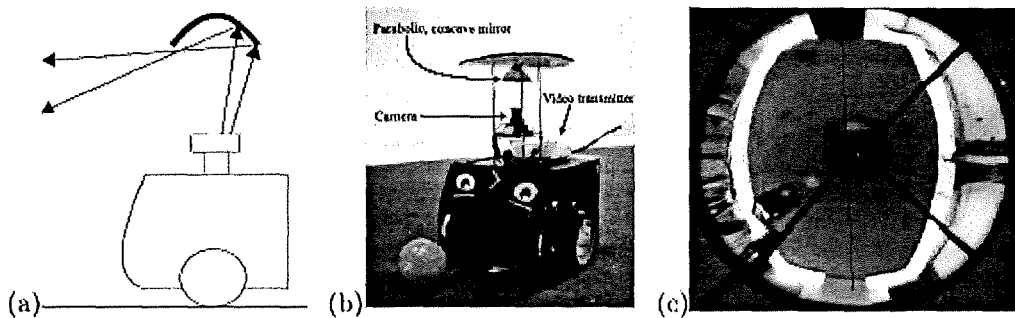
อย่างไรก็ตาม Systematic Error นั้น สามารถแก้ไขได้ด้วยการตรวจวัดและแก้ไขให้ถูกต้องได้ ในขณะที่ Non Systematic Error นั้น ไม่สามารถที่จะคาดการณ์ได้ว่าเกิดขึ้นมากเพียงใด ดังนั้น ในการเคลื่อนที่ของหุ่นยนต์ จึงเกิดวงรีของความไม่แน่นอนที่หุ่นยนต์จะมีตำแหน่งอยู่ภายในวงรีนั้น ที่ใหญ่ขึ้นเรื่อยๆ ดังรูป



รูปที่ 1 วงรีของความไม่แน่นอนในการเคลื่อนที่ของหุ่นยนต์

2.2.2 Machine Vision

วิธีนี้อาศัยการใช้กล้องวิดีโอหรือกล้องถ่ายภาพ ทำการถ่ายภาพพื้นที่รอบตัวหุ่นที่สนใจ จากนั้นจึงใช้คอมพิวเตอร์ประมวลผลภาพที่ได้ เพื่อให้ได้ถึงตำแหน่งของหุ่นยนต์ที่สนใจ การคำนวณจากภาพนี้ อาจใช้ภาพจากกล้องสองตัว (Stereo Vision) เพื่อให้ได้ความลึกของภาพเช่นเดียวกับการมองเห็นของมนุษย์ หรืออาจใช้ภาพสะท้อนจากกระจกชนิดพิเศษ เพื่อให้ได้ภาพการมองเห็นรอบตัว (Omni-Directional Vision)

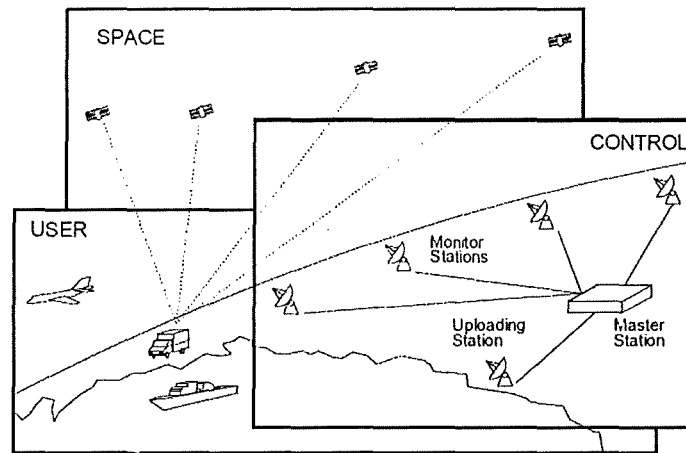


รูปที่ 2 Omni Directional Vision

การหาตำแหน่งจากการมองเห็นนี้ จำเป็นที่จะต้องมีการกำหนดจุดสังเกตหรือ Landmark ให้กับตัวหุ่นยนต์ และสภาพแสง รูปร่างตลอดจนสีของจุดสังเกตนี้จะต้องมีความถูกต้องใกล้เคียงกับที่ได้ป้อนให้คอมพิวเตอร์ไว้ ดังนั้น วิธีนี้จึงเหมาะกับสภาพแวดล้อมที่มีการจัดวางแผน เตรียมข้อมูลไว้แล้ว

2.2.3 Global Positioning System

วิธีนี้อาศัยการใช้ดาวเทียม 24 ดวง โคจรรอบอยู่ 19,000 ไมล์ทะเลเหนือพื้นผิวโลก โดยการหาตำแหน่งบนผิวโลก ในสามมิติจะอาศัยการวัดเวลาที่ต่างกันของสัญญาณที่รับได้จากดาวเทียมดวงต่างๆ นำมาคำนวณให้ได้ ระยะเสมือน (Pseudo Range) จากตัวรับถึงดาวเทียมดวงต่างๆ แล้วจึงใช้วิธี Triangulation โดยใช้หลักทางเรขาคณิต เพื่อให้ได้ตำแหน่งที่ต้องการ



รูปที่ 3 ระบบโดยรวมของระบบ GPS

ระบบ GPS นี้ สามารถให้ความแม่นยำได้สูงสุดถึงในระดับเซนติเมตร แต่ในการใช้งานทั่วไปนั้น เครื่องรับของพลเรือนจะให้ความแม่นยำอยู่ในระยะ 3-5 เมตร แต่อย่างไรก็ตาม อัตราเร็วของระบบนี้จะให้ตำแหน่งได้อยู่ที่ 1-10 Hz ซึ่งอาจที่จะไม่เพียงพอในการควบคุมหุ่นยนต์ที่มีความเร็วสูง อีกทั้งระบบจะมีปัญหาเมื่อตัวรับอยู่ในบริเวณจุดอับสัญญาณ เช่นภายในตึก เป็นต้น

2.2.4 Inertial Navigation System

วิธีนี้ใช้การวัด โดยอาศัยตัวตรวจวัดที่ทำกรวัดสถานะของระบบโดยเทียบกับกรอบเฉื่อย (Inertia Frame) ซึ่งในที่นี้ได้แก่ Rate Gyroscope ซึ่งวัดอัตราการหมุนของระบบ และ Accelerometer ซึ่งวัดความเร่งของระบบในแนวแกนของ Sensor เมื่อได้ค่าจากตัวตรวจวัดเหล่านี้แล้ว จึงนำไปเข้าสู่เทคนิค ในการที่จะหาถึงมุมและความเร็ว ตลอดจนตำแหน่งของระบบ แต่อย่างไรก็ตามความคลาดเคลื่อนที่ได้จากการวัดด้วยวิธีนี้อาจเกิดจาก

- ความคลาดเคลื่อนจากอุณหภูมิ (Temperature Drift)
- การเกิดความโน้มเอียง (Bias) ในการวัด
- สัญญาณรบกวนอันเกิดจากการทำงานภายในตัวเซ็นเซอร์เอง
- สัญญาณรบกวนจากภายนอก

2.3 Inertial Navigation System

2.3.1 Rate Gyroscope Overview

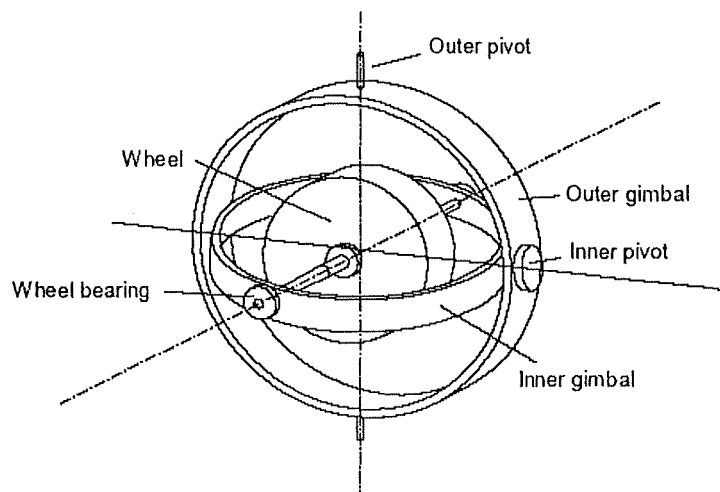
Rate Gyroscope เป็นตัวตรวจวัดชนิดหนึ่งซึ่งสามารถที่จะตรวจวัดอัตราการหมุนรอบแกนที่สนใจได้ โดยชนิดของ Gyroscope นี้อาจแบ่งได้เป็น

- Mechanical Gyroscope
- Optical Gyroscope
- MEMs Gyroscope

Mechanical Gyroscope

Mechanical Gyroscope ได้รับการพัฒนาเป็นประเภทแรกในช่วงทศวรรษที่ 1800 โดย ในปี 1852 นักฟิสิกส์ชาวฝรั่งเศส Leon Foucault ได้แสดงให้เห็นว่า Gyroscope นั้นสามารถที่จะวัดการหมุนของโลกได้

หลักการของ Mechanical Gyroscope นั้น ใช้หลักการของ gyroscopic precession ซึ่งสามารถแสดงได้จากการทดลองถือวงล้อจักรยานที่หมุนอยู่ แล้วพยายามบิดวงล้อนี้ในทิศที่ไม่ใช้การหมุน ผู้ทดลองจะพบกับแรงต้านเกิดขึ้น ซึ่งปรากฏการณ์นี้สามารถอธิบายได้ด้วยการอนุรักษ์โมเมนตัมเชิงมุมของการหมุนของวงล้อ จากปรากฏการณ์นี้เอง จึงมีการประยุกต์นำมาใช้ในการหาทิศทางของการเคลื่อนที่ โดยการสร้าง Stable Platform ดังรูป



รูปที่ 4 Mechanical Gyroscope Configuration

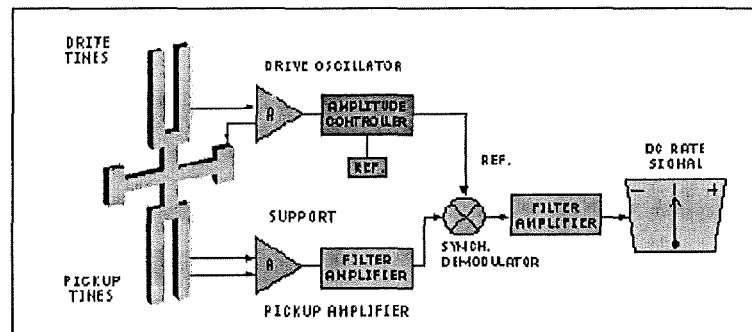
หลักการคือเมื่อมีการเคลื่อนที่ที่เกิดขึ้น วงล้อที่อยู่ภายในจะพยายามรักษาเสถียรภาพของตัวเองไว้ เราจึงสามารถวัดมุมปัจจุบันเทียบกับวงล้อที่บริเวณ Wheel bearing ทั้งสองได้ ซึ่งวิธีการนี้ถูกนำไปใช้ในการนำทางของเรือเดินสมุทรและเครื่องบินตลอดจนยานอวกาศในยุคแรกๆ

Optical Gyroscope

Gyroscope ใช้หลักของแสงในการหาอัตราการหมุนที่ต้องการวัด โดยอาศัยการเปลี่ยนเฟสของแสงเลเซอร์เมื่อเคลื่อนที่ในทิศต่างๆ ที่กระทบกับการหมุน แล้วนำมาวิเคราะห์ถึงอัตราการหมุนต่อไป เนื่องจาก Gyroscope ประเภทนี้มีความซับซ้อนและละเอียดอ่อน จึงทำให้มีราคาที่สูงกว่าแพงตั้งแต่หลักแสนไปจนถึงหลักล้านบาท แต่ยังมีข้อได้เปรียบแบบกลไกเนื่องจากเล็กและมีน้ำหนักที่เบาตลอดจนใช้พลังงานที่น้อยกว่า

MEMs Gyroscope

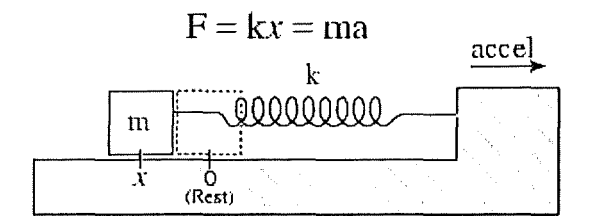
MEMs ย่อมาจาก Micro Electro-mechanical system เป็นการใช้เทคโนโลยีทางด้านการผลิตชิ้นส่วนอิเล็กทรอนิกส์ สร้างกลไกทางไฟฟ้า-เครื่องกลขนาดเล็กขึ้นมาบนชิปซิลิกอน ซึ่งทำให้สามารถผลิตได้เป็นจำนวนมาก และราคาถูก การทำงานของ MEMs Gyroscope นั้น ใช้การสั่นสะเทือน จากวัสดุ Piezoelctric สร้างให้เกิดการสั่นสะเทือน และเมื่อมีการหมุนเกิดขึ้น จะเกิดแรง Coriolis ขึ้นกับชิ้นส่วน ทำให้ลักษณะการหมุนเปลี่ยนไป จึงนำมาเปลี่ยนเป็นสัญญาณทางไฟฟ้าได้



รูปที่ 5 Block diagram ของ Gyroscope

2.3.2 Accelerometer Overview

เป็นตัวตรวจวัดที่สามารถวัดความเร่งของการเคลื่อนที่ ให้เป็นสัญญาณทางไฟฟ้า โดยใช้หลักการของมวล-สปริง ขนาดเล็กบนชิปซิลิกอน ซึ่งการเปลี่ยนแปลงของความยาวของสปริงซึ่งก็คือตำแหน่งของมวลที่เปลี่ยนไปจะสัมพันธ์กับความเร่งที่กระทำในทิศที่สนใจ ดังนั้นจึงทำการแปลงตำแหน่งของมวลที่เปลี่ยนไปออกมาเป็นสัญญาณทางไฟฟ้าต่อไป



รูปที่ 6 หลักการทำงานของ Accelerometer

2.4 Kalman Filter

Kalman filter เป็นตัวกรองที่ใช้สำหรับปัญหาทั่วไปในการประมาณภาวะ $x \in \mathbb{R}^n$ ของกระบวนการควบคุมที่ไม่ต่อเนื่องที่สามารถครอบงำได้ด้วย Stochastic difference equation

$$x_k = Ax_{k-1} + Bu_k + w_{k-1}$$

สมการที่ 1 Stochastic difference equation

ด้วยการวัด $z \in \mathbb{R}^m$ นั่นคือ

$$z_k = Hx_k + v_k$$

สมการที่ 2 Measurement equation

ตัวแปรสุ่ม w_k และ v_k แสดงถึง noise ของกระบวนการและการวัดตามลำดับ โดยมีสมมติฐานว่าไม่ขึ้นอยู่กันและกัน เป็น white noise และมีการกระจายแบบปกติ

ในทางปฏิบัติเมตริกซ์ process noise covariance Q และ measurement noise covariance R จะเปลี่ยนไปในแต่ละขั้นเวลา อย่างไรก็ตามเราตั้งสมมติฐานว่าคงที่

เมตริกซ์ขนาด $n \times n$ A ใน สมการที่ 1 Stochastic difference equation นั่นคือความสัมพันธ์ระหว่าง state ของ state ในช่วงเวลาที่ $k-1$ และ state ในเวลาปัจจุบัน k ในทางปฏิบัติ A อาจจะไปตามเวลาแต่ในที่นี้เราสมมติให้คงที่ เมตริกซ์ $n \times l$ B แสดงถึงความสัมพันธ์ของตัวควบคุมเลือกภายนอก $u \in \mathbb{R}^l$ กับ state x เมตริกซ์ $m \times n$ H ในสมการที่ 2 Measurement equation แสดงถึงความสัมพันธ์ระหว่าง state กับ measurement z_k ในทางปฏิบัติ H อาจจะไปในแต่ละเวลาแต่เราสมมติให้คงที่

ในการจะเขียนสมการสำหรับ Kalman filter เราเริ่มจากตั้งจุดมุ่งหมายที่จะหาสมการที่จะมาคำนวณการประมาณ state ภายหลัง \hat{x}_k^- และถ่วงน้ำหนักระหว่างผลลบของค่าการวัดจริง z_k กับค่าที่ได้จากการคาดเดา $H\hat{x}_k^-$ ดังแสดงด้านล่าง

$$\hat{x}_k = \hat{x}_k^- + K(z_k - H\hat{x}_k^-)$$

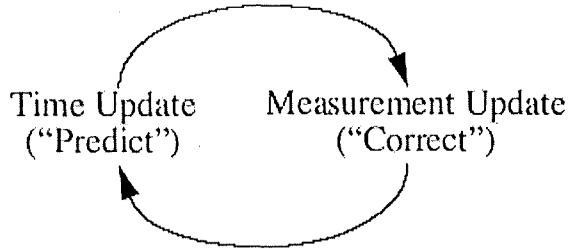
สมการที่ 3 measurement innovation

เมตริกซ์ขนาด $n \times m$ K ในสมการที่ 3 measurement innovation นั่นคือค่าตัวคูณที่จะลดค่าความผิดพลาดของ posteriori error covariance รูปแบบหนึ่งของผลลัพธ์ของการหาค่า K นั่นคือ

$$K_k = P_k^- H^T (HP_k^- H^T + R)^{-1}$$

kalman filter ประมาณกระบวนการโดยการให้รูปแบบหนึ่งของการควบคุมแบบป้อนกลับ: ฟิลเตอร์จะทำการประมาณ state ของกระบวนการที่เวลาหนึ่งแล้วทำการป้อนกลับด้วยการวัดที่มี noise และดังนั้นสมการของ kalman filter จึงแบ่งได้เป็นสองกลุ่มนั่นคือกลุ่มสมการปรับปรุงเวลาและกลุ่มสมการปรับปรุงการวัด กลุ่มสมการปรับปรุงเวลาจะรับผิดชอบในส่วนของการคาดเดา state ไปด้านหน้าและการประมาณ error covariance เพื่อที่จะให้ได้การประมาณแบบนำหน้าสำหรับขั้น

ช่วงเวลาถัดไป สมการการปรับปรุงการวัดจะรับผิดชอบในส่วนของการป้อนกลับ สมการปรับปรุงเวลานั้นอาจมองว่าเป็นสมการการคาดเดาในขณะที่สมการการปรับปรุงการวัดอาจมองได้ว่าเป็นสมการการแก้ไข ในขั้นตอนสุดท้ายของการประมาณคือการรวมทั้งการคาดเดาและการแก้ไขเข้าด้วยกันเพื่อแก้ปัญหา



สมการการปรับปรุงเวลา

$$\hat{x}_k^- = A\hat{x}_{k-1} + Bu_k$$

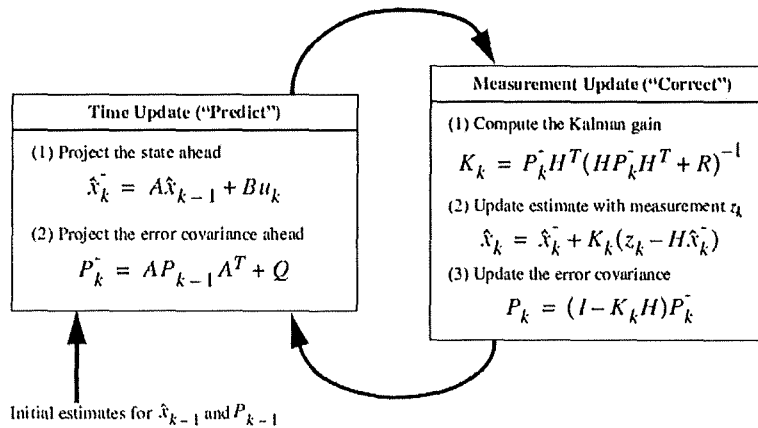
$$P_k^- = AP_{k-1}A^T + Q$$

สมการการปรับปรุงการวัด

$$K_k = P_k^- H^T (HP_k^- H^T + R)^{-1}$$

$$\hat{x}_k = \hat{x}_k^- + K(z_k - H\hat{x}_k^-)$$

$$P_k = (I - K_k H)P_k^-$$



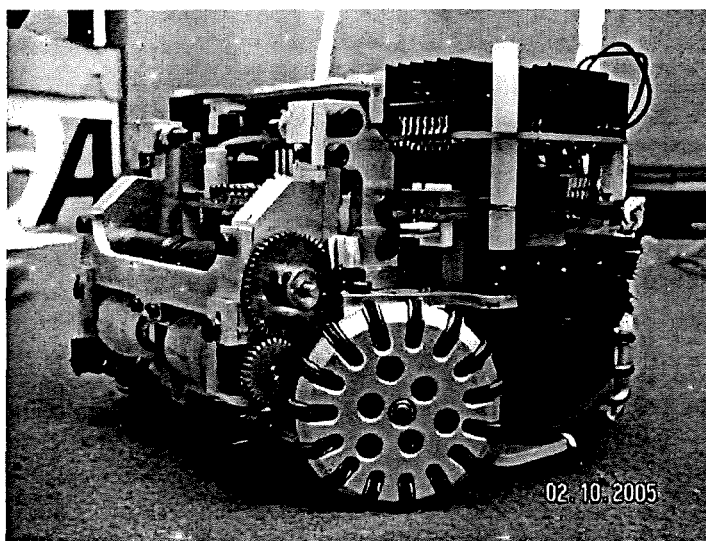
รูปที่ 7 สรุปลำดับการทำงานของ Discrete kalman filter

บทที่ 3

การออกแบบและพัฒนา Inertial Measurement Unit

3.1 หุ่นยนต์ที่จะทำการศึกษาและพัฒนา

หุ่นยนต์ที่จะทำการศึกษา ใช้หุ่นยนต์เคลื่อนที่ได้ของทีม Plasma-Z ใช้ล้อแบบ Omni-Directional ซึ่งมีองศาอิสระเท่ากับ 3 และมีความเร็วสูงสุดประมาณ 3 m/s ที่ความเร่งมากกว่า 4 m/s²



รูปที่ 8 หุ่นยนต์เคลื่อนที่ได้ของทีม Plasma-Z

รายละเอียดเชิงเทคนิคของหุ่นยนต์มีโดยดังนี้

- Microcontroller C8051F120; Operate at 100 MHz
- มอเตอร์ขับเคลื่อน Fahulber 2224SR-006 with Planetary Gearhead 14:1
- ล้อ 4 x Single-in-line Omni-Directional Wheel, Aluminum
- วงจรขับ L298N dual channel. Maximum Drive Current = 2A
- Motion Controller LM629: Precision Motion Controller.
- ความเร็วสูงสุด 3 m/s ในทุกทิศทาง
- ความเร่งสูงสุด 4 m/s²
- ความเร็วเชิงมุมสูงสุด 10 rad/s

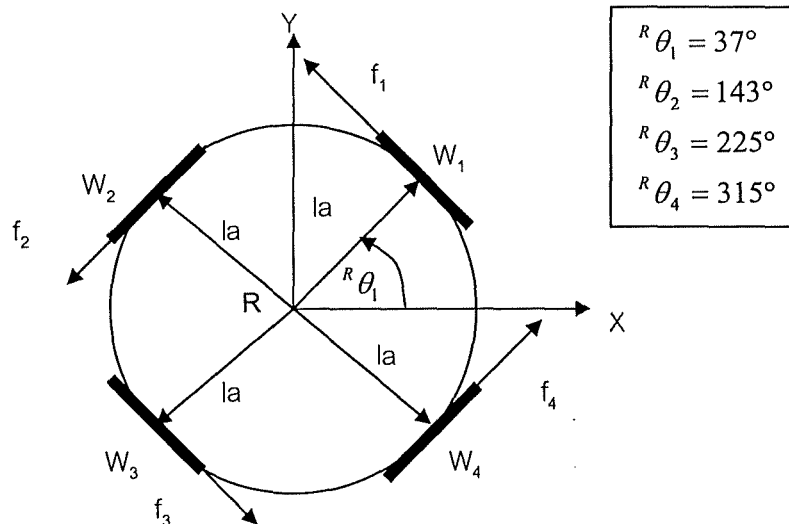
3.2 การคำนวณการเคลื่อนที่ของล้อ Omni-Directional

3.2.1 สมการการเคลื่อนที่ไปข้างหน้า และ สมการการเคลื่อนที่ผกผัน (Forward and Inverse Kinematics Solution)

ในการที่จะให้หุ่นยนต์เคลื่อนที่ในความเร็วและทิศทางที่ต้องการ จึงมีความจำเป็นที่จะต้องมีการหาสมการคณิตศาสตร์เพื่อทำการแปลงความเร็วของตัวหุ่นยนต์ที่ต้องการให้อยู่ในรูปความเร็วเชิงมุมของล้อแต่ละล้อ โดยในการหารูปแบบสมการจะใช้วิธีการของ Muir

3.2.2 Coordinate System Assignment

ในการหา Kinematics Model เลือกใช้ระบบแกนแบบ Sheth-Uicker แทนระบบแกนแบบ Denavit-Hartenberg ซึ่งระบบแกนแบบ Sheth-Uicker นี้ จะวางแกนอ้างอิง 2 แกนซ้อนกัน ณ จุดสัมผัสโดยแกนหนึ่งจะอยู่บนกลไกหนึ่ง และแกนที่เหลืออยู่บนกลไกที่ต่อกันอยู่ ซึ่งในกรณีหุ่นยนต์เคลื่อนที่ได้นี้ ล้อหนึ่งซึ่งสัมผัสกับพื้น แกนหนึ่งจะอยู่ที่ล้อ ณ จุดสัมผัส และอีกแกนจะอยู่ที่พื้น ณ จุดสัมผัส ทำให้สามารถหาแบบจำลองของการเคลื่อนที่บนพื้นได้



รูปที่ 9 Coordinate System Assignment

3.3.3 Actuated Inverse Velocity Solution

สมการที่ใช้ในการคำนวณความเร็วในการหมุนของแต่ละล้อ เมื่อกำหนดความเร็วลัพธ์ของหุ่นยนต์ หรือเรียกว่า Inverse Kinematics ซึ่งมีรูปแบบดังนี้

$$\begin{bmatrix} \omega_{w_1,x} \\ \omega_{w_2,x} \\ \omega_{w_3,x} \\ \omega_{w_4,x} \end{bmatrix} = \frac{1}{R} \begin{bmatrix} f_{x_1} & f_{y_1} & l_a \\ f_{x_2} & f_{y_2} & l_a \\ f_{x_3} & f_{y_3} & l_a \\ f_{x_4} & f_{y_4} & l_a \end{bmatrix} \begin{bmatrix} \bar{B} V_{Bx} \\ \bar{B} V_{By} \\ \bar{B} V_{Bz} \end{bmatrix}$$

สมการที่ 4 Actuated Inverse Velocity Solution

เมื่อ

$\omega_{W_n x}$:	ความเร็วเชิงมุมของล้อที่ n
f_{x_n}	:	Unit Vector ของทิศทางของล้อที่ n ในแนวแกน $^R X$
f_{y_n}	:	Unit Vector ของทิศทางของล้อที่ n ในแนวแกน $^R Y$
R	:	รัศมีของล้อ เท่ากับ 27mm
la	:	ระยะจากจุดศูนย์กลางหุ่นยนต์ ถึงศูนย์กลางล้อ เท่ากับ 82.261mm
${}^B v_{Bx}$:	ความเร็วของศูนย์กลางหุ่นยนต์ เทียบกับจุดที่ซ้อนทับกับศูนย์กลางของหุ่นยนต์ในขณะนั้น (Instantaneously coincident point) ในแนว X
${}^B v_{By}$:	ความเร็วของศูนย์กลางหุ่นยนต์ เทียบกับจุดที่ซ้อนทับกับศูนย์กลางของหุ่นยนต์ในขณะนั้น (Instantaneously coincident point) ในแนว Y
${}^B \omega_{Bz}$:	ความเร็วเชิงมุมของหุ่นยนต์ เทียบกับจุดที่ซ้อนทับกับศูนย์กลางของหุ่นยนต์ในขณะนั้น (Instantaneously coincident point) รอบแกน Z

3.3.4 Sensed Forward Velocity Solution

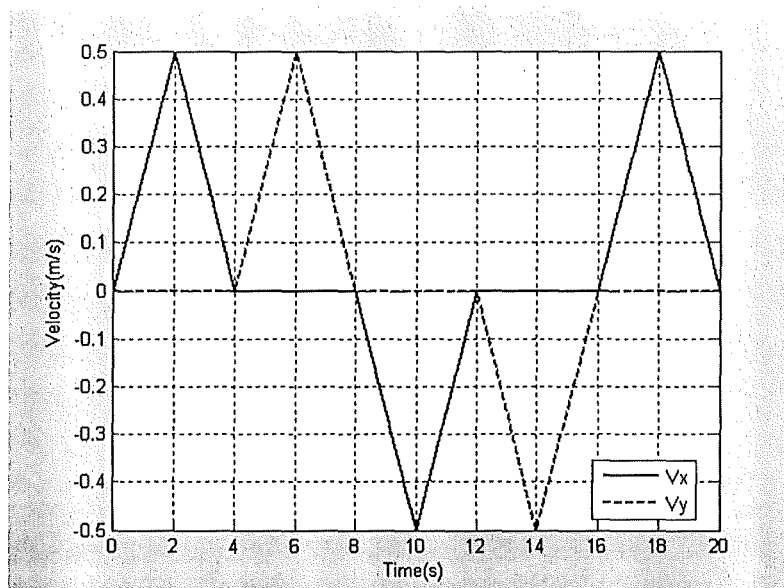
สมการที่ใช้ในการคำนวณความเร็วล้อของหุ่นยนต์ เมื่อกำหนดความเร็วในการหมุนของแต่ละล้อ หรือเรียกว่า Forward Kinematics ซึ่งเมื่อแทนค่าตัวแปรต่างๆ จะได้สมการดังนี้

$$\begin{bmatrix} {}^B v_{Bx} \\ {}^B v_{By} \\ {}^B v_{Bz} \end{bmatrix} = \begin{bmatrix} -0.0103138 & -0.0103138 & 0.0103138 & 0.0103138 \\ 0.0094756 & -0.0094756 & -0.0083896 & 0.0083896 \\ 0.0821134 & 0.0821134 & 0.0698862 & 0.0698862 \end{bmatrix} \begin{bmatrix} \omega_{w_1 x} \\ \omega_{w_2 x} \\ \omega_{w_3 x} \\ \omega_{w_4 x} \end{bmatrix}$$

สมการที่ 5 Sensed Forward Velocity Solution

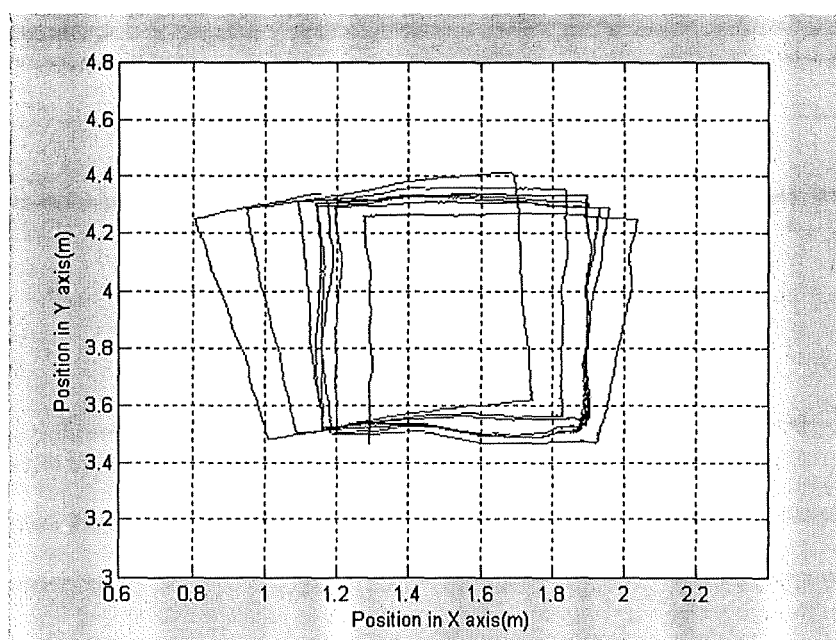
3.2 การทดสอบการเคลื่อนที่แบบ Open-Loop

กระทำโดยการป้อนรูปแบบของ Profile ความเร็วของหุ่นยนต์ที่ต้องการ โดยกำหนดให้หุ่นยนต์เคลื่อนที่เป็นรูปสี่เหลี่ยมจัตุรัส ขนาด 1 x 1 m เพื่อให้เหมาะสมกับพื้นที่ทดสอบ ซึ่งมีขนาดจำกัด โดยมีลักษณะ Profile ของความเร็วของหุ่นยนต์ดังนี้



รูปที่ 10 คำสั่งความเร็วในแกนต่างๆ ที่ป้อนให้กับหุ่นยนต์

ในขณะเดียวกันก็จับตำแหน่งการเคลื่อนที่ไว้ โดยการใช้ระบบ Machine Vision ของทีม Plasma-Z แล้วนำผลมาวิเคราะห์บนคอมพิวเตอร์ โดยพบว่า

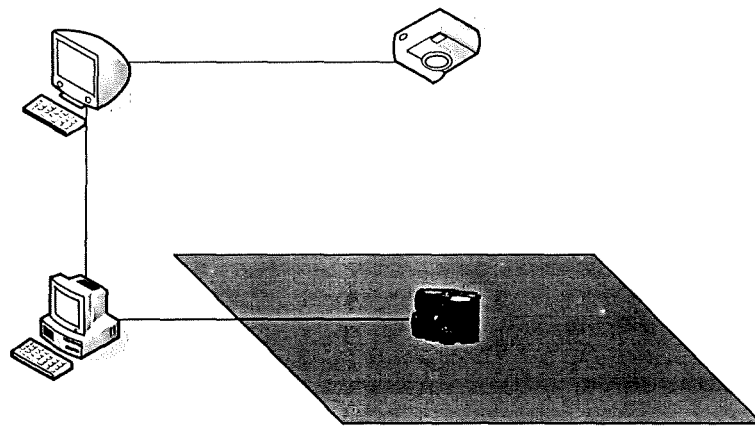


รูปที่ 11 ตำแหน่งของหุ่นยนต์เมื่อเคลื่อนที่แบบ Open-Loop

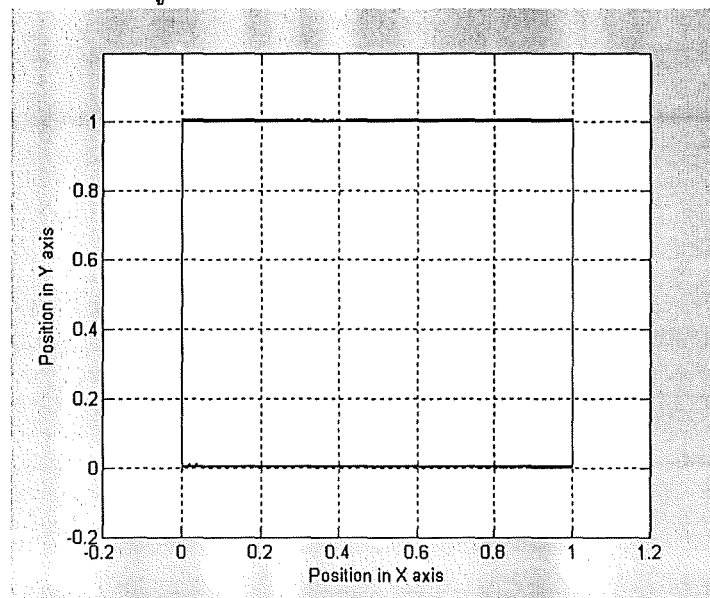
จะพบว่า หุ่นยนต์เคลื่อนที่เป็นรูปสี่เหลี่ยมที่บิดเบี้ยว อันเนื่องมาจากความผิดพลาดที่เกิดจากการเคลื่อนที่ของหุ่นยนต์ จึงมีความจำเป็นที่จะต้องมีการควบคุมความเร็วของหุ่นยนต์ในรูปของความเร็วลัพท์และมุมของหุ่นยนต์เกิดขึ้น

และจากการทดลอง จึงเป็นที่น่าสนใจว่าการวัดความเร็วของหุ่นยนต์โดยการดูจากการหมุนของล้อ โดยการใช้สมการ Sensed Forward Velocity Solution ซึ่งเป็นวิธีพื้นฐานที่ใช้ในหุ่นยนต์เคลื่อนที่ได้โดยทั่วไป จะสามารถตรวจพบความผิดพลาดในการเคลื่อนที่ของหุ่นยนต์ได้หรือไม่ ถ้าได้เราอาจใช้การวัดนี้ นำมาป้อนกลับเข้าสู่ชุดควบคุม เพื่อลดความผิดพลาดที่เกิดขึ้นได้

การที่จะหาคำตอบได้นั้น ทำโดยการป้อนสมการ Sensed Forward Velocity Solution ให้หุ่นยนต์ แล้วจึงเก็บข้อมูลความเร็วที่หุ่นยนต์คำนวณออกมา แล้วนำมาหาเป็นตำแหน่งที่วัดได้จากล้อแล้วนำมาเปรียบเทียบกับตำแหน่งที่วัดได้จากกล้อง



รูปที่ 12 ระบบโดยรวมของการทดสอบ



รูปที่ 13 ตำแหน่งของหุ่นยนต์เมื่อวัดจากล้อ

จะพบว่า การวัดจากล้อ ไม่สามารถบอกถึงความผิดพลาดจากการวิ่งได้ เนื่องจากตำแหน่งที่ได้เป็นตำแหน่งที่ตรงกับคำสั่งที่สั่งเข้าไป แสดงว่าตัวล้อหมุนตามคำสั่งที่สั่งเข้าไป แต่มีการสลิปขึ้นระหว่างตัวล้อกับพื้น จึงทำให้เกิดความผิดพลาดในการวิ่งมากขึ้นเรื่อยๆ

Serial Cable

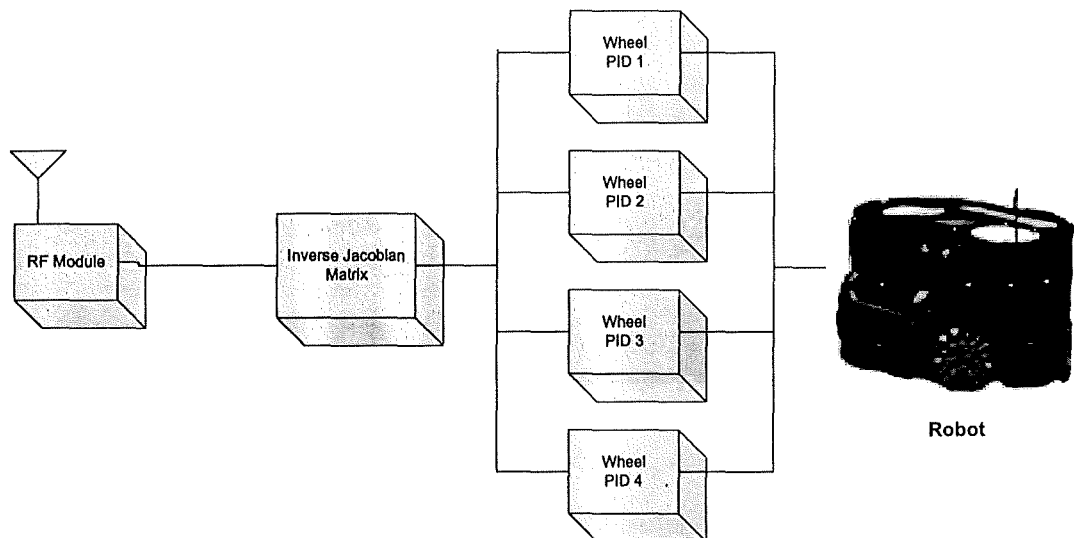
AI Computer

ดังนั้นการที่จะทำการควบคุมให้หุ่นยนต์เคลื่อนที่ได้ถูกต้องนั้น จึงต้องการวัดการเคลื่อนที่อื่น นำมาใช้แทนการวัดจากล้อ เนื่องจากการวัดจากล้อไม่สามารถบอกถึงความผิดพลาดที่เกิดขึ้นได้

การใช้การวัดโดยใช้ Inertia Measurement น่าจะเป็นทางเลือกหนึ่งที่จะสามารถควบคุมให้หุ่นยนต์มีความผิดพลาดที่น้อยลงได้ ด้วยตัวหุ่นยนต์เอง และไม่ต้องมีการป้อนข้อมูลช่วยจากภายนอก

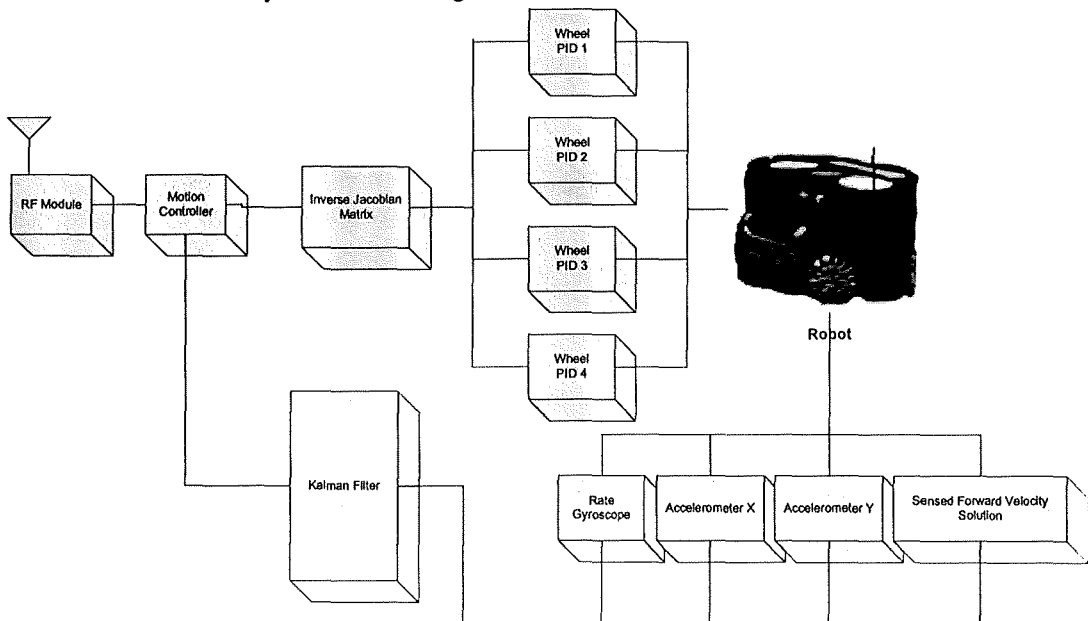
3.3 การนำ Inertial Measurement Unit ผนวกเข้ากับระบบ

3.3.1 Plasma-Z Robot System.



รูปที่ 14 ระบบบนหุ่นยนต์ของทีม Plasma-Z ก่อนการปรับปรุง

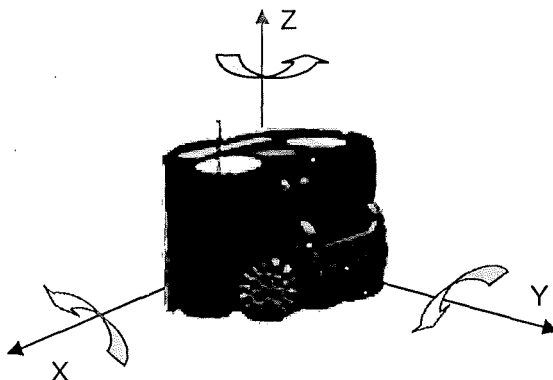
3.3.1 Robot system after Integrate Inertial Measurement Unit



รูปที่ 15 ระบบบนหุ่นยนต์หลังการปรับปรุง

3.4 การพัฒนา Inertial Measurement Unit

เมื่อพิจารณาที่จะมีการติดตั้ง Inertial Measurement Unit เพิ่มเติมเข้ากับระบบแล้ว จึงจำเป็นที่จะต้องมีการพิจารณาว่าควรที่จะมีการวัดในแนวแกนใดของการเคลื่อนที่ของหุ่นบ้าง ซึ่งจากการวิเคราะห์ พบว่าการวัดใน 3 แกนของการเคลื่อนที่: ด้านข้าง(X),ด้านหน้า-หลัง(Y)และการหมุนรอบตัว(Z) ก็เพียงพอแล้วในการวัด เนื่องจากหุ่นยนต์วิ่งอยู่บนระนาบของสนามเท่านั้น ดังนั้นการเคลื่อนที่บนล่าง จึงไม่จำเป็น และเมื่อกำหนดว่า หุ่นยนต์จะไม่มีการพลิกล้ม การวัดการหมุนในแนวแกน X และ Y จึงไม่จำเป็น



รูปที่ 16 แกนที่ทำการวัดและแกนที่ไม่ทำการวัด (สีเทา)

ดังนั้นจึงสรุปได้ว่า จะทำการวัดในการเคลื่อนที่ในแนว X และ Y ของหุ่นยนต์ และวัดการหมุนในแนวแกน Z ของหุ่นยนต์

3.5 การเลือก Gyro และ Accelerometer ที่จะนำมาใช้งาน

เมื่อทราบถึงแกนที่จำเป็นที่จะต้องวัดแล้ว จึงถึงขั้นตอนของการเลือกตัวตรวจวัดที่จะนำมาใช้งานซึ่งในที่นี้คือ Rate Gyroscope สำหรับวัดการหมุน และ Accelerometer สำหรับวัดการเคลื่อนที่

เนื่องจากตัวตรวจวัดทั้งสองมีในท้องตลาดเป็นจำนวนมาก จึงจำเป็นที่จะต้องมีการกำหนดในการเลือก คือ ต้องมีราคาที่ไม่แพงมาก เนื่องจากมีราคาตั้งแต่ไม่กี่พันบาท ไปจนถึงหลายสิบล้านบาท ขึ้นอยู่กับเทคโนโลยีและความแม่นยำในการวัด และข้อกำหนดอีกข้อคือ ต้องมีขนาดเล็กพอที่จะประกอบลงในหุ่นยนต์ที่จะทำการทดสอบได้

ข้อจำกัดในบางประการของการจัดหาตัวตรวจวัดประเภทนี้ที่พบคือ เนื่องจากสามารถนำตัวตรวจวัดเหล่านี้ไปสร้างจรวดขีปนาวุธได้ ดังนั้นในการสั่งซื้อบางครั้งจะต้องมีการรับรองการนำไปใช้งาน จึงทำให้การจัดซื้ออาจจะไม่สะดวกนัก โดยเฉพาะอย่างยิ่งตัวตรวจวัดที่มีความแม่นยำสูงๆ

เมื่อพิจารณาถึงข้อจำกัดต่างๆ แล้ว จึงได้ตัวตรวจวัดที่น่าสนใจและเหมาะสม ดังนี้

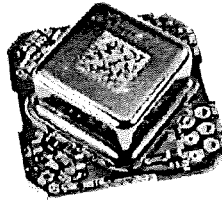
3.5.1 Rate Gyroscope

Silicon Sensing Systems: CRS03-11

CRS03-11 เป็น Rate Gyroscope แบบ Piezoelectrics ที่มีจุดเด่นคือ มีย่านการวัดสูงสุดถึง 10 rad/s ซึ่งมากพอที่จะวัดความเร็วในการหมุนสูงสุดของหุ่นยนต์ได้ และมี Bandwidth = 50 Hz ซึ่งเพียงพอกับการวัด CRS03-11 มีคุณสมบัติทางเทคนิค ดังนี้

Specification(Typical Data)	CRS03-11
Rate Range	$\pm 573^\circ/\text{s}$
Output	Analog voltage (Ratiometric)
Scale Factor	
<ul style="list-style-type: none"> Nominal 	3.49mV/ $^\circ/\text{s}$
<ul style="list-style-type: none"> Variation over temperature range 	$< \pm 5\%$
<ul style="list-style-type: none"> Non linearity 	$< \pm 0.5\%$ of full scale
Bias	
<ul style="list-style-type: none"> Setting tolerance 	$< \pm 30^\circ/\text{s}$
<ul style="list-style-type: none"> Variation over temperature range 	$< \pm 30^\circ/\text{s}$
<ul style="list-style-type: none"> Ratiometric error 	$< \pm 1^\circ/\text{s}$
<ul style="list-style-type: none"> Drift vs. time 	$< \pm 0.55^\circ/\text{s}$ in any 30s period
<ul style="list-style-type: none"> g Sensitivity 	$< \pm 0.1^\circ/\text{s/g}$ on any axis
Bandwidth	55Hz(-3dB)
Quiescent Noise	$< 1\text{mV rms}$
Environment	
<ul style="list-style-type: none"> Temperature 	-20°C to $+60^\circ\text{C}$
<ul style="list-style-type: none"> Linear acceleration 	$< 100\text{g}$
<ul style="list-style-type: none"> Shock 	200g(1ms, $\frac{1}{2}$ sine)
<ul style="list-style-type: none"> Vibration 	2g rms (20Hz to 2kHz random)
<ul style="list-style-type: none"> Cross-axis sensitivity 	$< 5\%$
Mass	$< 10\text{ gram}$
Electrical	
<ul style="list-style-type: none"> Voltage (Supply) 	+4.75V to +5.25V
<ul style="list-style-type: none"> Current (Supply) 	$< 35\text{mA}$ (steady state)
<ul style="list-style-type: none"> Noise and ripple 	$< 15\text{mV rms}$ (DC to 100Hz)
<ul style="list-style-type: none"> Ready time 	$< 0.2\text{s}$

ตารางที่ 1 คุณสมบัติทางเทคนิคของ CRS03-11



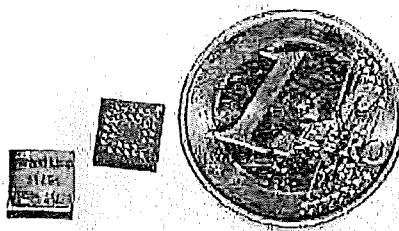
รูปที่ 17 Silicon Sensing Systems: CRS03-11

Analog Devices: ADXRS300

ADXRS300 เป็น Rate Gyroscope แบบ MEMs ซึ่งถูกสร้างขึ้นด้วยซิลิกอน มีจุดเด่นคือมีขนาดเล็กมาก คือ ประมาณ 7x7 mm เท่านั้น จึงนิยมนำมาใช้ในอุปกรณ์ขนาดเล็ก และสัญญาณขาออกยังมีวงจร Signal Conditioning อยู่ภายในอีกด้วย แต่เนื่องจากใช้ Package แบบ Ball Grid Array ซึ่งบัดกรีด้วยมือได้ยาก จึงทำให้การนำไปใช้มีข้อจำกัด

Specification(Typical Data)	ADXRS300ABG
Sensitivity	
• Dynamic Range	$\pm 300^{\circ}/s$
• Initial	5 mV/ $^{\circ}/s$
• Over temperature	5 mV/ $^{\circ}/s$
• Non linearity	0.1 % of Full Scale
Null	
• Initial Null	2.5 V
• Over temperature	2.3 – 2.7 V
• Turn on time	35 ms
• Linear Acceleration effect	0.2 $^{\circ}/s/g$
• Voltage Sensitivity	1 $^{\circ}/s/V$
Noise Performance	
• Rate Noise Density	0.1 $^{\circ}/s/\sqrt{Hz}$
Frequency Response	
• 3dB Bandwidth(User Selectable)	40 Hz
• Sensor Resonant Frequency	14 kHz
Power Supply	
• Operating Voltage Range	4.75 to 5.25 V ,5.0 V Typical
• Quiescent Supply Current	6.0 mA
Temperature Range	
• Specified Performance Grade A	-40 to +85 $^{\circ}C$

ตารางที่ 2 คุณสมบัติทางเทคนิคของ ADXRS300ABG



รูปที่ 18 Analog Devices: ADXRS300

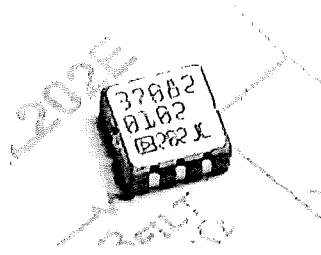
3.5.2 Accelerometers

Analog Devices: ADXL202

เป็น Accelerometer แบบ MEMs ซึ่งมีย่านการวัดอยู่ที่ ± 2 g ซึ่งน่าจะเพียงพอกับการใช้งาน และยังเป็น Accelerometer ที่มีผู้นิยมใช้มากที่สุดตัวหนึ่ง เนื่องจากมีขนาดเล็ก และยังให้สัญญาณขาออกทั้งแบบ Analog และ Digital พร้อมกัน ผู้ใช้จึงสามารถเลือกใช้ได้ตามความเหมาะสม

Specification(Typical Data)	ADXL202
Sensor Input	
• Measurement Range	± 2 g
• Non Linearity	0.2 % of Full Scale
Sensitivity	
• $X_{\text{FILT}}, Y_{\text{FILT}}$	312 mV/g
• Temperature Drift	$\pm 0.5\%$
Zero g Bias level	
• 0 g Voltage $X_{\text{FILT}}, Y_{\text{FILT}}$	2.5V
• 0 g Offset vs. Temperature	2.0 mg/ $^{\circ}$ C
Noise Performance	
• Noise Density	200 μ g/ $\sqrt{\text{Hz}}$ rms
Frequency Response	
• 3 dB Bandwidth	6 kHz
• Sensor Resonant Frequency	10 kHz
Power Supply	
• Operating Voltage Range	3.0 to 5.25 V
• Quiescent Supply Current	0.6 mA
Temperature Range	
• Operating Range	-40 to +85 $^{\circ}$ C

ตารางที่ 3 คุณสมบัติทางเทคนิคของ ADXL202



รูปที่ 19 Analog Devices: ADXL202

3.5.3 ผลการเลือกใช้ตัวตรวจวัด

ในส่วนของ Rate Gyroscope นั้น เลือกใช้ CRS03-11 เนื่องจากมีเสถียรภาพต่อแรงสั่นสะเทือนที่มากกว่าซึ่งเหมาะสมกับหุ่นยนต์ที่มีล้อแบบ Omni-Directional ที่จะเกิดแรงสั่นสะเทือนขณะวิ่งสูง และยังสามารถบัดกรีเข้ากับวงจรได้ง่ายกว่าของ Analog Devices

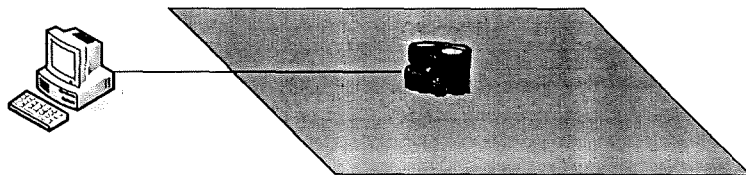
Accelerometer นั้นใช้ ADXL202 เนื่องจากมีราคาถูก สามารถหาซื้อได้ง่าย และมีงานการวัดที่เหมาะสมกับการนำมาใช้งาน

3.6 การออกแบบตัวกรองสัญญาณความถี่ต่ำผ่าน (Low Pass Filter)

ก่อนเราจะนำเอาสัญญาณที่ได้จาก Sensor นำมาใช้งานนั้น จำเป็นที่จะต้องมีการทำ Signal Conditioning เสียก่อน นั่นคือ กำจัดสัญญาณรบกวนที่มาจาก การวัดออก และทำการปรับขนาดของสัญญาณ ให้มีขนาดที่เหมาะสมกับการนำไปใช้เสียก่อน

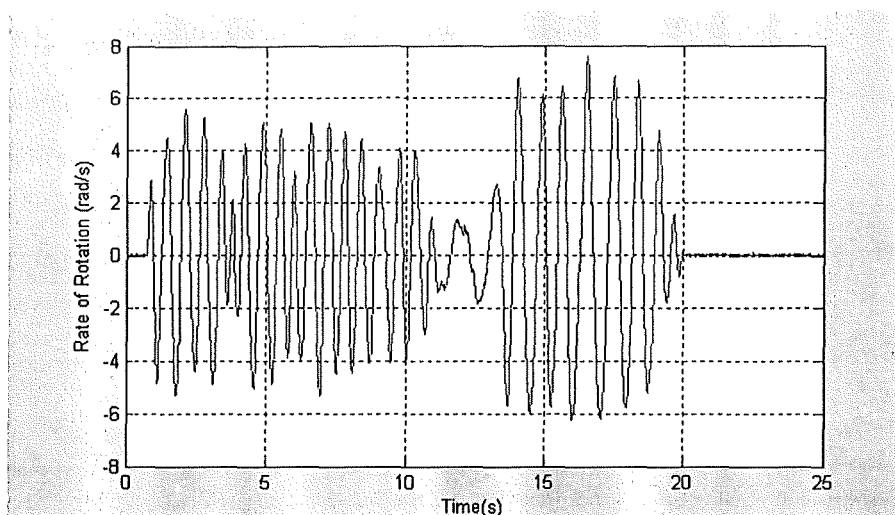
การกำจัดสัญญาณรบกวน ซึ่งในกรณีนี้คือ สัญญาณรบกวนความถี่สูง ซึ่งจะเข้ามากับการวัด ซึ่งสัญญาณนี้มีความถี่อยู่เหนือการเคลื่อนที่ของตัวหุ่นยนต์ การปล่อยให้สัญญาณนี้เข้าไปในระบบ จะเป็นการเพิ่มความผิดพลาดในการวัดขึ้น

ก่อนที่จะออกแบบตัวกรองสัญญาณความถี่ต่ำผ่านนั้น จะต้องทราบถึงช่วงความถี่ของการเคลื่อนที่ ที่เกิดขึ้นจริงเสียก่อน เพื่อที่จะทำการหา Cut off frequency ที่เหมาะสม การหาช่วงความถี่ของการเคลื่อนที่นี้ ทำได้โดยการเก็บค่าจาก Sensor ในขณะที่หุ่นยนต์เคลื่อนที่ตามปกติ แล้วนำค่าที่ได้ นำมาทำ Discrete Fourier Transform เพื่อแปลงสัญญาณที่ได้จากในรูป Time-Domain ให้อยู่ในรูปของ Frequency-Domain



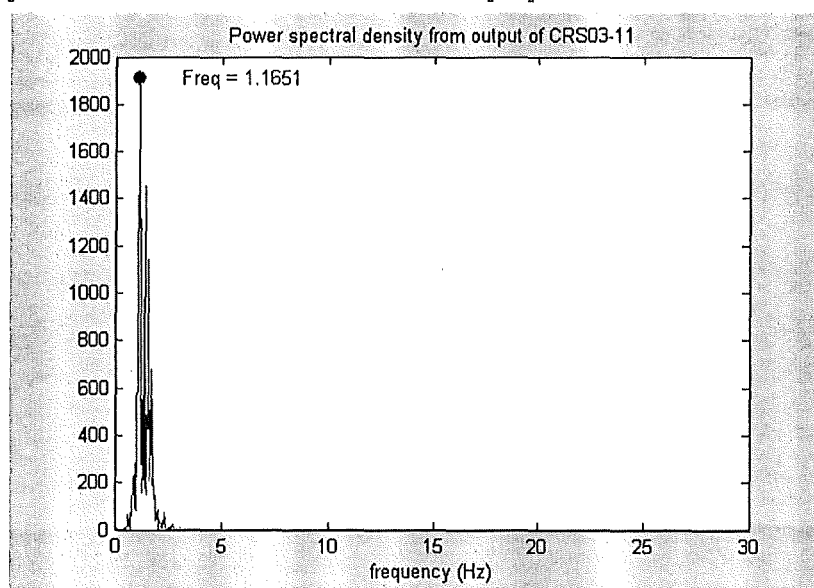
รูปที่ 20 การทดลองหาช่วงความถี่ของการเคลื่อนที่

ผลการวัดในรูป Time - domain



รูปที่ 21 ผลการวัดจาก Gyroscope

ข้อมูลหลังผ่าน Discrete Fourier Transform อยู่ในรูปของ Frequency-Domain



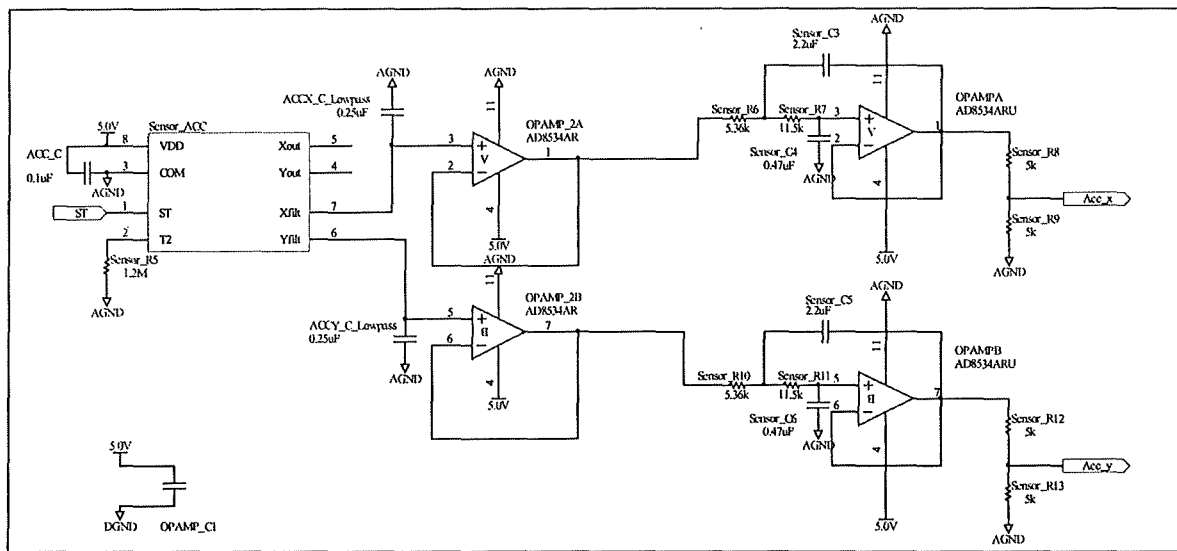
รูปที่ 22 ผลการแปลงข้อมูลการเคลื่อนที่ให้อยู่ในรูป Frequency Domain

จากข้างต้น จะพบว่าช่วงความถี่ของการเคลื่อนที่ของหุ่นยนต์นั้น อยู่ในช่วงเพียง 0 -2 Hz เท่านั้น โดยมีจุดสูงสุดอยู่ที่ 1.1651 Hz ดังนั้นจึงเลือก Cutoff Frequency เพื่ออีก 1 decade หรือ 10 เท่า ทำให้ได้ Cutoff Frequency ของ Low-pass filter เท่ากับที่ 20 Hz

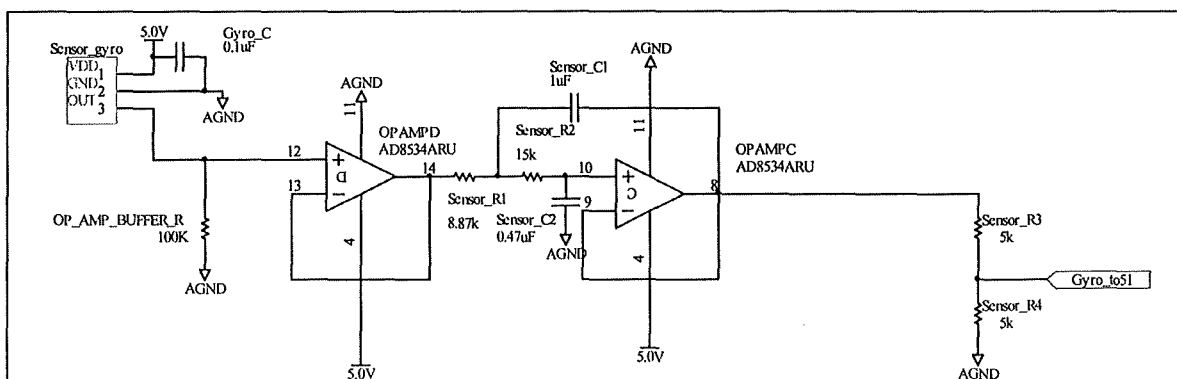
เมื่อได้ Cutoff Frequency แล้ว จึงทำการเลือก ชนิดของ Filter ที่จะนำมาใช้ โดยเลือกใช้ analog Filter แบบ Butterworth เนื่องจาก มีอัตราขยายของ Pass Band ที่เรียบ และเลือกใช้ Order ของ Filter เป็น 3th order Butterworth ในส่วนของ Accelerometer ซึ่งให้ความชันในการกรอง

เท่ากับ 60 db/decade ซึ่งน่าจะเพียงพอ และยังสามารถใช้ First order filter ซึ่งมีอยู่ภายในตัว Accelerometer เป็น pole หนึ่งของ third order filter นี้ได้ แต่ในส่วนของ Gyroscope เลือกใช้ 2nd order Butterworth Filter เนื่องจากน่าจะเพียงพอกับการใช้งานและข้อจำกัดของจำนวนอุปกรณ์

เมื่อได้ Order และ Cutoff Frequency ของ Filter แล้ว จึงนำมาคำนวณหาค่าของตัวเก็บประจุและตัวต้านทานเพื่อให้ได้ Transfer function ที่ถูกต้องเหมาะสม ซึ่งผลลัพธ์ที่ได้คือ วงจรของ Low pass filter ที่สมบูรณ์ ดังรูป



รูปที่ 23 วงจรส่วนของ Accelerometer



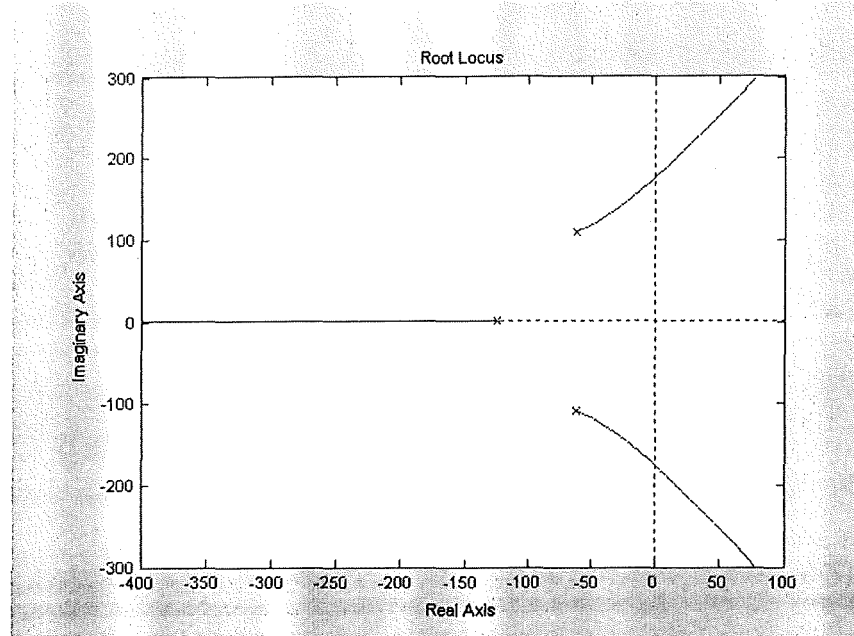
รูปที่ 24 วงจรส่วนของ Gyroscope

ในส่วนวงจรของ Gyroscope นั้น มีการเพิ่มส่วนของ Voltage Follower เพื่อป้องกันวงจรขับขาออกของ Sensor ไม่ให้รับผลของ Impedance ของชุด Filter อันจะส่งผลต่อเสถียรภาพของวงจรถดลด

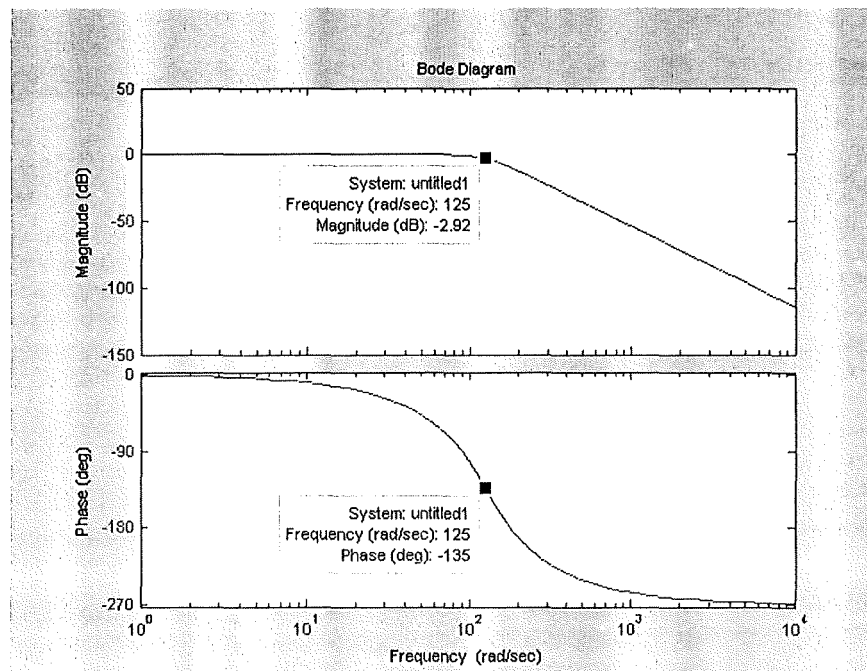
ทางด้านขาออกของ Filter ทั้งสามชุดนั้น จะมี Voltage Divider เพื่อลดแรงดันก่อนที่จะเข้า ADC ภายในไมโครคอนโทรลเลอร์ เนื่องจาก ADC รับค่าได้อยู่ในช่วง 0-2.4V เท่านั้น

เมื่อได้แผนผังวงจรแล้ว จึงทำการตรวจสอบผลโมเดลของ Filter ว่าเป็นไปตามที่คำนวณหรือไม่ ซึ่งได้ผลดังนี้

ส่วนของ 3th order Butterworth low pass filter สามารถนำมา plot บน root-locus และ bode diagram เพื่อวิเคราะห์หึ่ง frequency response ดังนี้

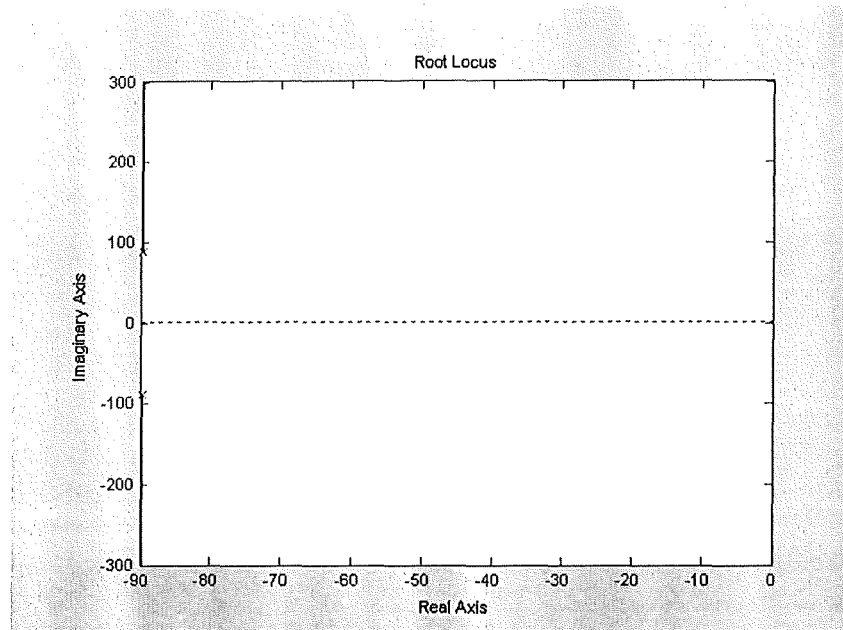


รูปที่ 25 Root-Locus ของ 3th order Butterworth low pass filter ที่ออกแบบ

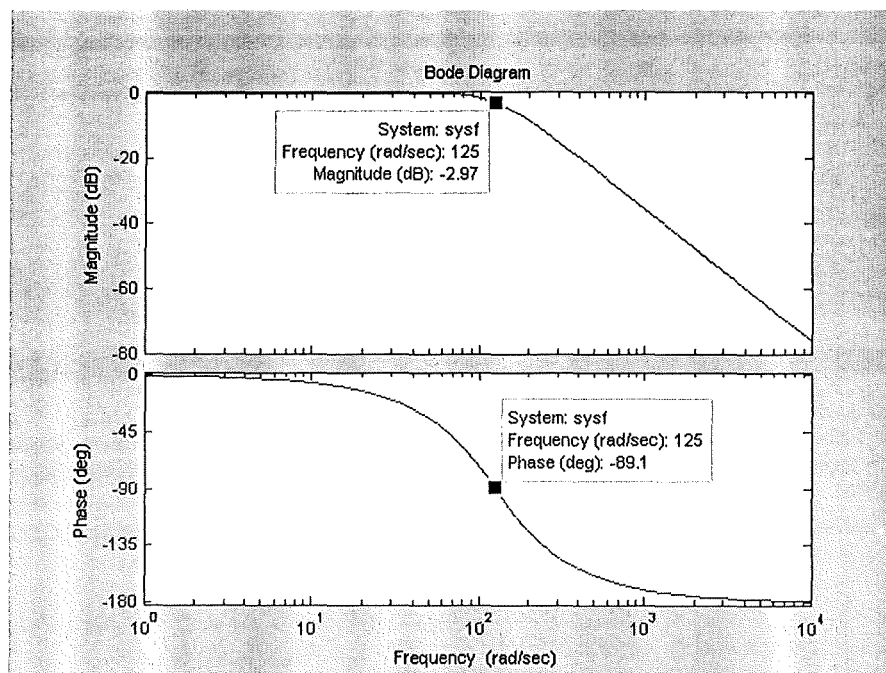


รูปที่ 26 Bode Diagram ของ 3th order Butterworth low pass filter ที่ออกแบบ

2nd order Butterworth low pass filter บน Root-locus และ Bode Diagram



รูปที่ 27 Root-Locus ของ 2nd order Butterworth low pass filter ที่ออกแบบ

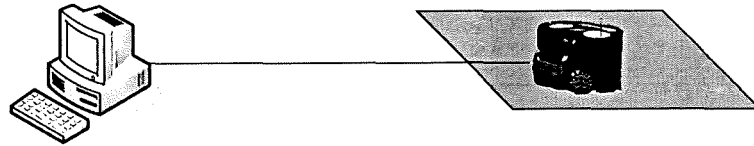


รูปที่ 28 Bode Diagram ของ 2nd order Butterworth low pass filter ที่ออกแบบ

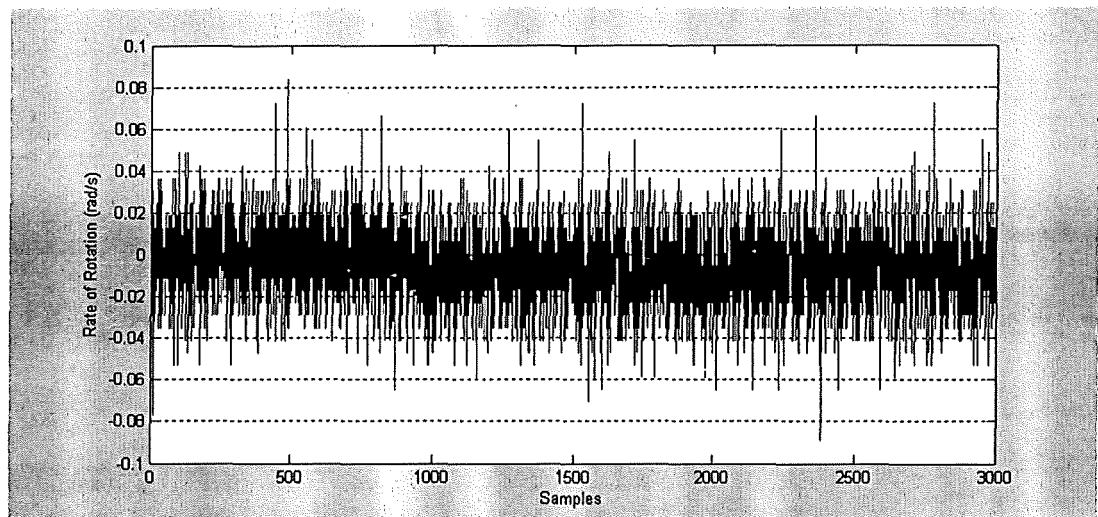
จะเห็นได้ว่า Filter ทั้งสองมี Cutoff Frequency (-3dB) ที่ Frequency เท่ากับ 20 Hz หรือ เท่ากับ $20 \times 2\pi = 125.663$ rad/s เป็นไปตามที่ออกแบบไว้

3.7 การทดลองวัดค่า Gyroscope และ Accelerometer ขณะหยุดนิ่ง

เมื่อได้ออกแบบและสร้างวงจรส่วนของ Sensor และ Filter เป็นที่เรียบร้อยแล้ว จึงทำการทดลองวัดค่าที่ได้จาก Gyroscope และ Accelerometer ขณะหยุดนิ่ง เพื่อหาว่ามีสัญญาณรบกวนที่วัดได้มีมากเท่าใด ซึ่งการวัดนี้ทำโดยตั้งหุ่นยนต์อยู่กับที่ แล้วทำการเก็บค่าการวัดที่ได้ นำมาวิเคราะห์

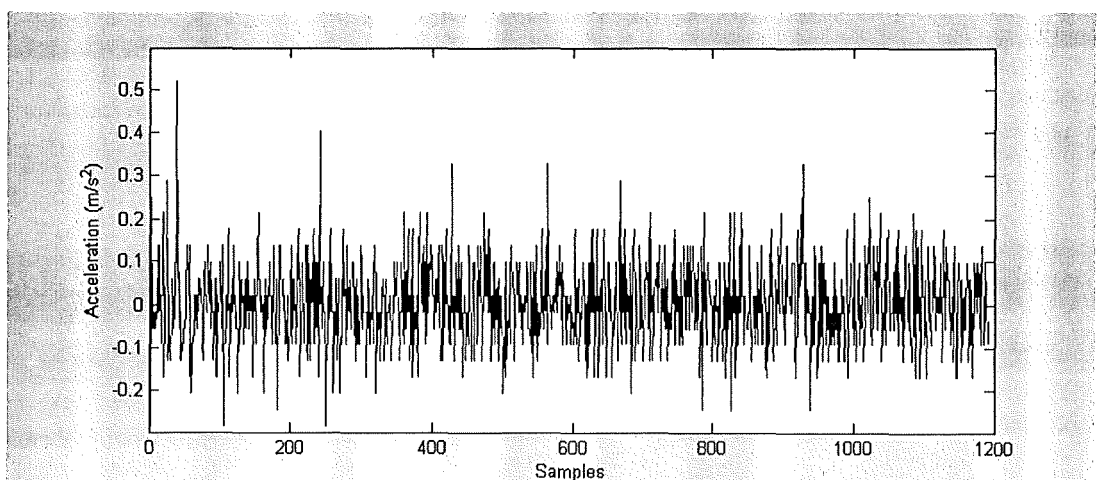


รูปที่ 29 การทดลองวัดค่าขณะหยุดนิ่ง



Serial

รูปที่ 30 ข้อมูลที่วัดได้จาก Gyroscope ขณะหยุดนิ่ง



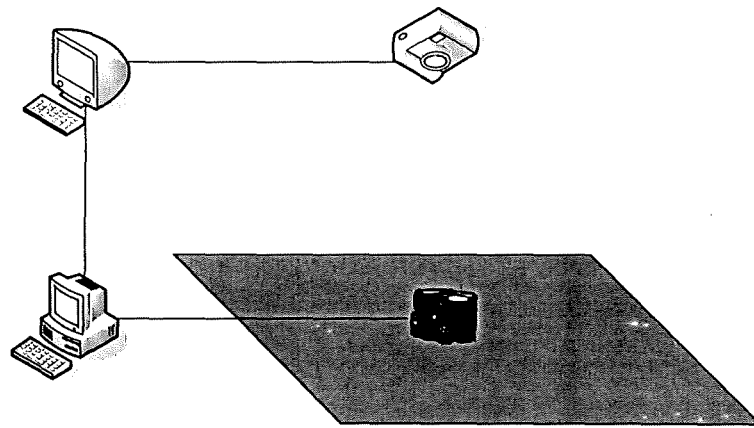
รูปที่ 31 ข้อมูลที่วัดได้จาก Accelerometer ขณะหยุดนิ่ง

จากข้างต้น พบว่าค่าที่ได้จาก Gyroscope มีการแกว่งอยู่ในช่วง ± 0.04 rad/s หรือประมาณ $2.29^\circ/s$ และมี Standard Deviation เท่ากับ 0.02233 rad/s และค่าจาก Accelerometer แกว่งอยู่ในช่วง ± 0.2 m/s² มี Standard Deviation เท่ากับ 0.09359 m/s²

3.8 การออกแบบและใช้งาน Kalman Filter

3.8.1 การทดลองเพื่อหาผลของการใช้งาน Kalman Filter

ทำโดยการป้อนความเร็วที่ต้องการให้หุ่นยนต์ เหมือนกับการทดลองในข้อ 3.2 โดยให้หุ่นยนต์วิ่งเป็นรูปสี่เหลี่ยมจัตุรัส และในขณะเดียวกันก็มีการเก็บข้อมูลตำแหน่งของหุ่นยนต์ที่เคลื่อนที่ด้วยระบบ Vision และยังรับข้อมูลที่ผ่านการคำนวณจากตัวหุ่นยนต์มาทางสาย Serial Cable แล้วนำข้อมูลจากทั้งสองแหล่งมาวิเคราะห์เปรียบเทียบกัน



รูปที่ 32 การทดสอบการใช้งาน Kalman Filter

3.8.2 การใช้งาน Kalman Filter ในแนวแกน Z

เมื่อพิจารณาถึงความสัมพันธ์ของค่าที่วัดได้จาก Gyro จะพบว่าเป็นตามความสัมพันธ์

$$\omega = K \cdot \Omega - Bias$$

เมื่อ ω คือ ความเร็วเชิงมุมที่วัดได้

Vision Computer

Ω คือ ค่าที่วัดได้จาก Gyro

K คือ Scale Factor ในการแปลงค่าที่วัดได้ให้เป็นความเร็วเชิงมุม

$Bias$ คือ Bias หรือค่ากลางของการวัดของ Gyro เป็นจุดที่มีค่าเท่ากับ 0 rad/s

ในทางปฏิบัติจะพบว่า ค่า Bias นี้จะเปลี่ยนแปลงไปตามเวลา โดยอาจขึ้นอยู่กับหลายปัจจัย เช่น อุณหภูมิ แรงดันไฟเลี้ยง เป็นต้น ซึ่งเราจำเป็นต้องทำการหา Bias นี้ให้ได้ มิฉะนั้นจะ

เกิดความผิดพลาดสะสมขึ้น (Drift) เมื่อทำการอินทิเกรตความเร็วเชิงมุมให้เป็นมุม ส่วนค่า Scale Factor นี้เราสามารถที่จะ Calibrate ให้แม่นยำได้

วิธีการหนึ่งที่ใช้ในการที่จะหาถึง Bias ของ Gyro คือการใช้ Kalman Filter ซึ่งเป็น State Estimator/Observer ตัวหนึ่ง มาทำการ estimate หา Bias ที่น่าจะเป็นจากการวัด โดยการป้อนการวัดจากแหล่งอื่นเข้าช่วย ซึ่งในที่นี้คือการวัดจากล้อนั่นเอง

การหาสมการ Time Update

ให้ตัวแปร State ของระบบในแนวแกน Z คือ

$$X = \begin{Bmatrix} \theta \\ Bias \end{Bmatrix} \quad \text{โดย } \theta \text{ มุมของหุ่นยนต์ในแนวแกน Z, Bias คือ Bias ของ Gyro}$$

สมการระบบซึ่งอยู่ในรูป $\dot{X} = Ax + Bu$ คือ

$$\dot{\theta} = \omega = -Bias + K \cdot \Omega \quad \text{จากสมการด้านบน}$$

$$Bias = 0 \quad \text{Gyro Bias มีค่าคงที่}$$

จากสมการข้างต้น ทำให้หา A ซึ่งเป็น Jacobian ของ \dot{X} เทียบกับ State

$$A = \begin{bmatrix} \frac{d\dot{\theta}}{d\theta} & \frac{d\dot{\theta}}{dBias} \\ \frac{dBias}{d\theta} & \frac{dBias}{dBias} \end{bmatrix} = \begin{bmatrix} 0 & -1 \\ 0 & 0 \end{bmatrix}$$

ดังนั้น เราจึงได้สมการ Time update

$$X_{k+1} = A \cdot X_k + Bu_k$$

$$P_k^- = AP_k^- A^T + Q$$

การหาสมการ Measurement Update

เราเลือกใช้การวัดที่จะใช้ใน Measurement Update คือ Bias ของ Gyro เมื่อได้ การวัดจากล้อ (ω_{wheel}) จากสมการ

$$Bias_{wheel} = K \cdot \Omega - \omega_{wheel}$$

เราสามารถหา H ซึ่งซึ่งเป็น Jacobian ของ Measurement เทียบกับ State

$$H = \begin{bmatrix} \frac{dBias_{wheel}}{d\theta} & \frac{dBias_{wheel}}{dBias} \end{bmatrix} = [0 \ 1]$$

ดังนั้น เราจึงสามารถหาสมการ Measurement Update ได้ ดังนี้

$$K_k = P_k^- H^T (HP_k^- H^T + R)^{-1}$$

$$\hat{x}_k = \hat{x}_k^- + K_k (z_k - H\hat{x}_k^-)$$

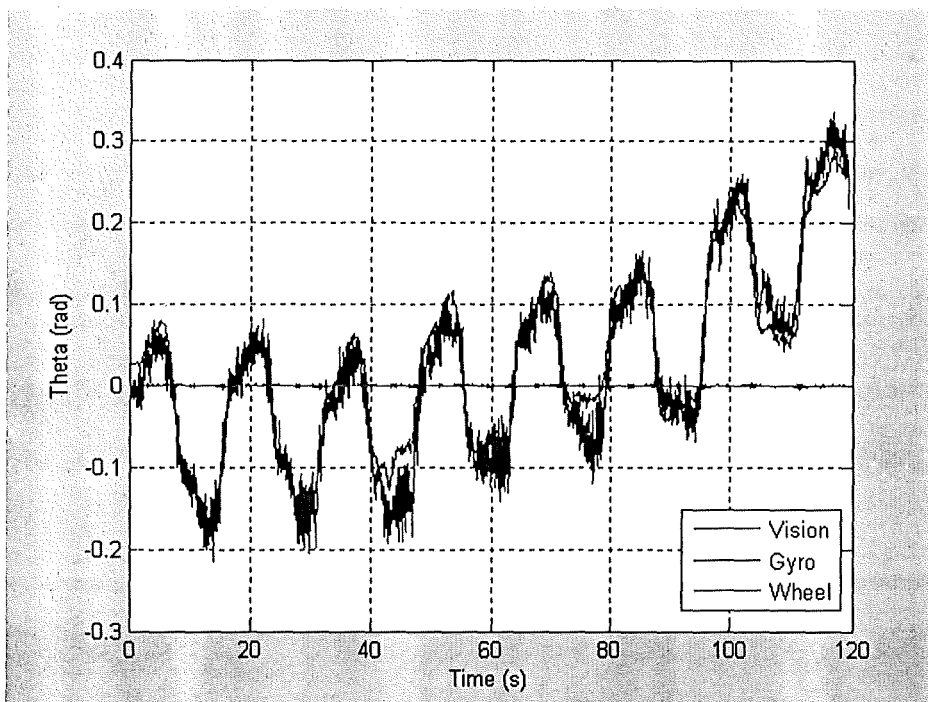
$$P_k = (I - K_k H) P_k^-$$

แต่เนื่องจากล้อของหุ่นยนต์เกิดการสลิปขึ้นในการวิ่ง ดังนั้นจากการทดลองจึงเพิ่ม สมมุติฐานคือ ล้อจะไม่มีการสลิปเกิดขึ้น เมื่อหยุดอยู่กับที่ ดังนั้น ระบบจะมีการ Measurement Update เมื่อ $\omega_{wheel} = 0$ หรือ หุ่นอยู่กับที่นั่นเอง

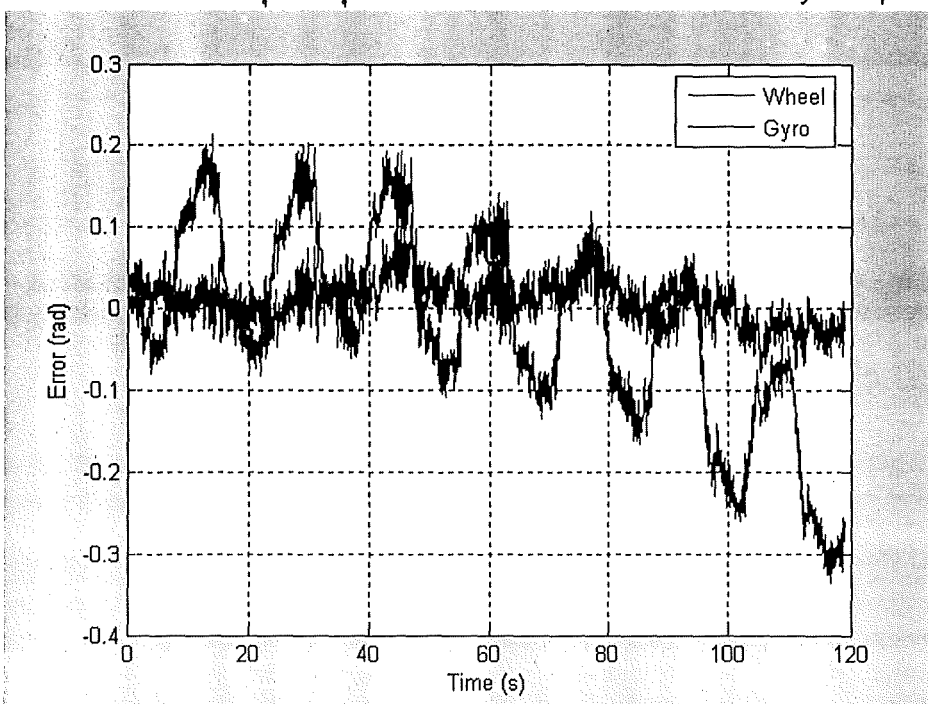
ผลการทดลอง

เลือกใช้ค่าเริ่มต้น

$$P = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, Q = \begin{bmatrix} 0.0001 & 0 \\ 0 & 0 \end{bmatrix}, R = 3.02$$



รูปที่ 33 เปรียบเทียบการวัดมุมของหุ่นยนต์ด้วยระบบ Vision กับการวัดด้วย Gyroscope และล้อ



รูปที่ 34 เปรียบเทียบความผิดพลาดของการวัดจากล้อและGyro เมื่อเทียบกับการวัดจาก Vision

จากรูปข้างต้น จะพบว่าการใช้การวัดข้อมูลจาก Gyroscope ผ่าน Kalman Filter จะสามารถลดความผิดพลาดในการวัด เมื่อเปรียบเทียบกับจากการวัดด้วยล้อแต่เพียงอย่างเดียวได้ โดยจะเห็นว่า เมื่อใช้ Gyroscope ผ่าน Kalman Filter จะมีความคลาดเคลื่อนจากการวัด อยู่ในช่วง ± 0.1 rad และมี Mean Square Error = 6.09 แต่การใช้ล้ออย่างเดียว ให้ความคลาดเคลื่อนเพิ่มขึ้นมากขึ้นตามเวลา, Mean Square Error = 101.56

3.8.2 การใช้งาน Kalman Filter ในแนวแกน X และ Y

ในการวัดด้วย Accelerometer นั้น มีหลักการคล้ายคลึงกันกับการวัดด้วย Gyroscope นั่นคือ

$$a = K \cdot A - Bias$$

เมื่อ a คือ ความเร่งที่วัดได้

A คือ ค่าที่วัดได้จาก Accelerometer

K คือ Scale Factor ในการแปลงค่าที่วัดได้ให้เป็นความเร่ง

Bias คือ Bias หรือค่ากลางของการวัดของ Accelerometer เป็นจุดที่มีค่าเท่ากับ 0 m/s^2

การหาสมการ Time Update

ให้ตัวแปร State ของระบบในแนวแกน X และ Y คือ

$$X = \begin{Bmatrix} V \\ Bias \end{Bmatrix} \quad \text{โดย } V \text{ ความเร็วของหุ่นยนต์ในแนวแกน X, Y, Bias คือ Bias}$$

ของ Accelerometer

สมการระบบซึ่งอยู่ในรูป $\dot{X} = Ax + Bu$ คือ

$$\dot{V} = a = -Bias + K \cdot A \quad \text{จากสมการด้านบน}$$

$$Bias = 0 \quad \text{Accelerometer Bias มีค่าคงที่}$$

จากสมการข้างต้น ทำให้หา A ซึ่งเป็น Jacobian ของ \dot{X} เทียบกับ State

$$A = \begin{bmatrix} \frac{d\dot{V}}{dV} & \frac{d\dot{V}}{dBias} \\ \frac{dBias}{dV} & \frac{dBias}{dBias} \end{bmatrix} = \begin{bmatrix} 0 & -1 \\ 0 & 0 \end{bmatrix}$$

ดังนั้น เราจึงได้สมการ Time update

$$X_{k+1} = A \cdot X_k + Bu_k$$

$$P_k^- = AP_k A^T + Q$$

การหาสมการ Measurement Update

เราเลือกใช้การวัดที่จะใช้ใน Measurement Update คือวัดความเร็วจากล้อ (V_{Wheel}) ซึ่งได้จากสมการ Sensed Forward Velocity Solution

เราสามารถหา H ซึ่งซึ่งเป็น Jacobian ของ Measurement เทียบกับ State

$$H = \begin{bmatrix} \frac{dV_{Wheel}}{dV} & \frac{dV_{Wheel}}{dBias} \end{bmatrix} = [1 \ 0]$$

ดังนั้น เราจึงสามารถหาสมการ Measurement Update ได้ ดังนี้

$$K_k = P_k^- H^T (HP_k^- H^T + R)^{-1}$$

$$\hat{x}_k = \hat{x}_k^- + K_k (z_k - H\hat{x}_k^-)$$

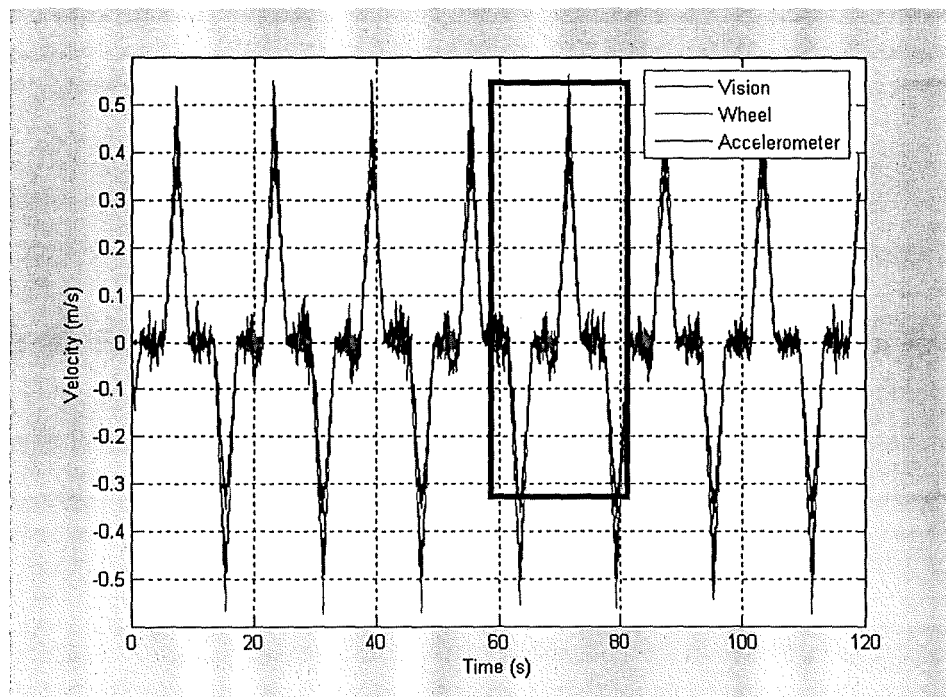
$$P_k = (I - K_k H) P_k^-$$

ผลการทดลอง

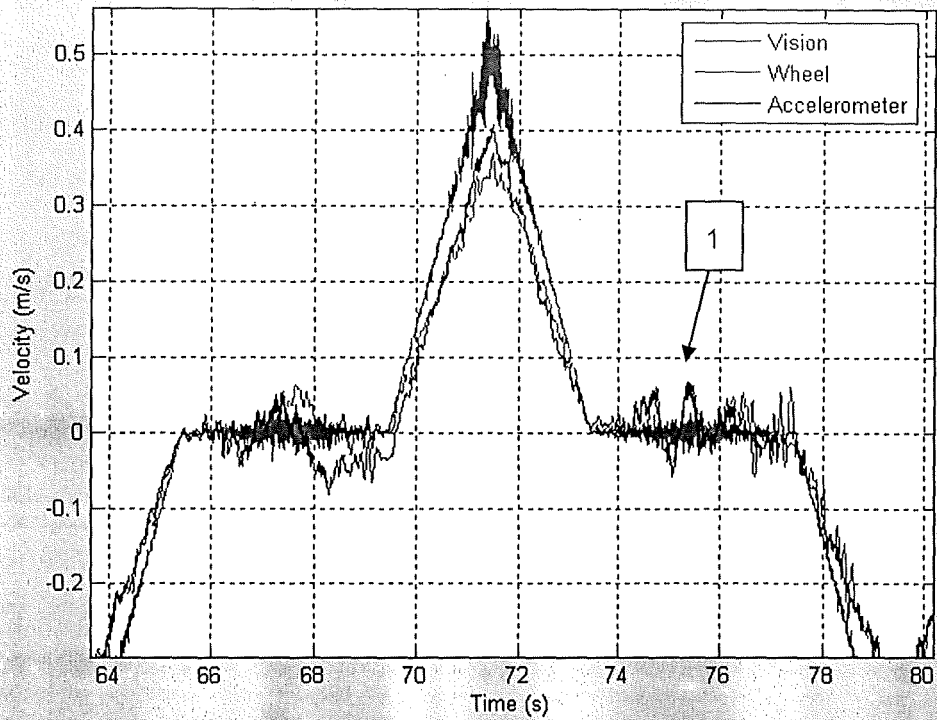
เลือกใช้ค่าเริ่มต้น

$$P = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, Q = \begin{bmatrix} 1 & 0 \\ 0 & 0.0001 \end{bmatrix}, R = 18.6$$

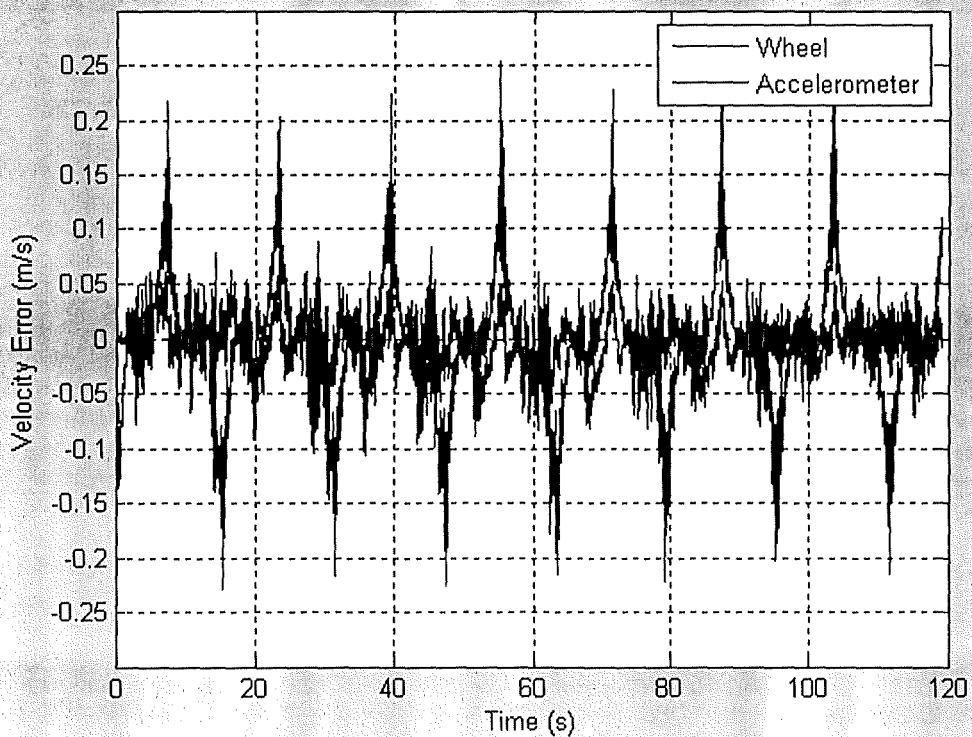
○ แกน X



รูปที่ 35 เปรียบเทียบการวัดความเร็วด้วย Vision กับการวัดด้วย Accelerometer ในแนวแกน X

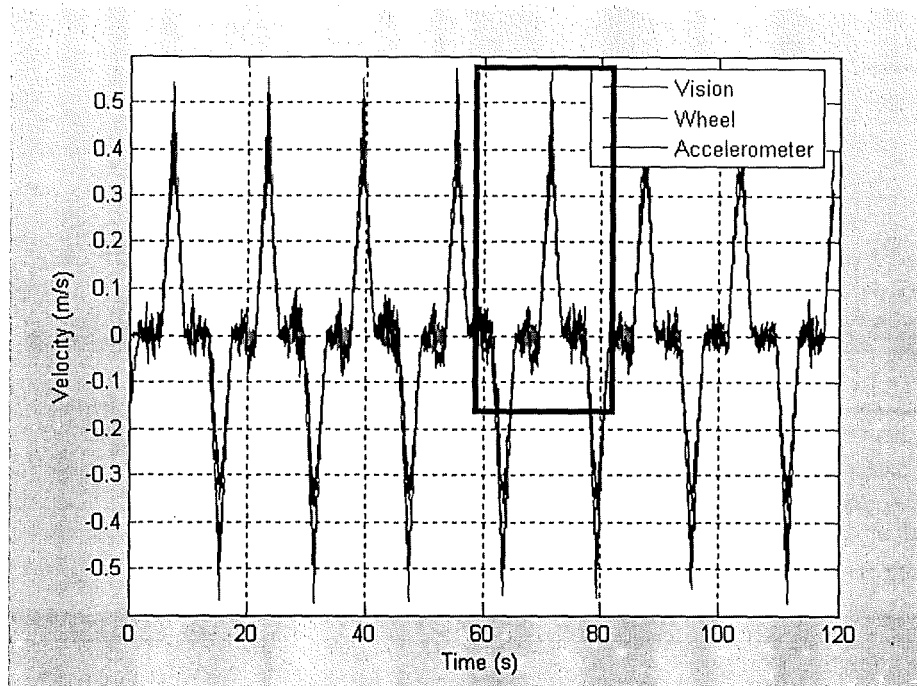


รูปที่ 36 ภาพขยาย (บริเวณสี่เหลี่ยม)

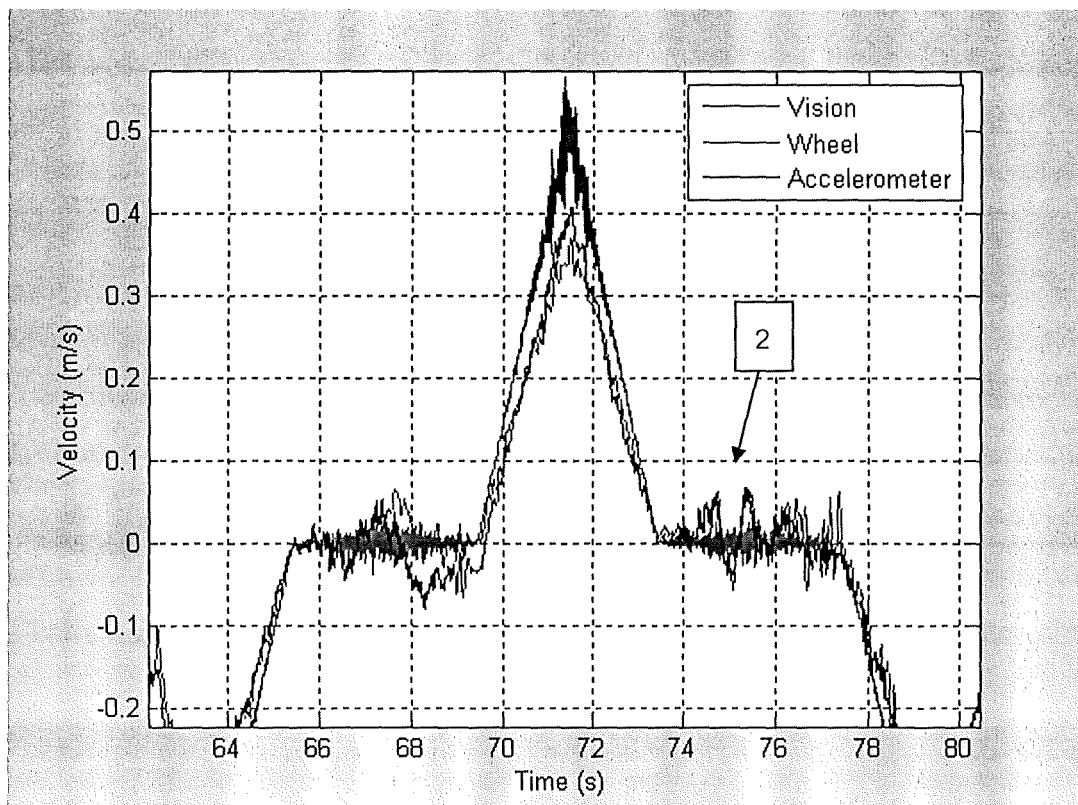


รูปที่ 37 เปรียบเทียบความผิดพลาดจากการวัดด้วยล้อและAccelerometer เมื่อเทียบกับ Vision

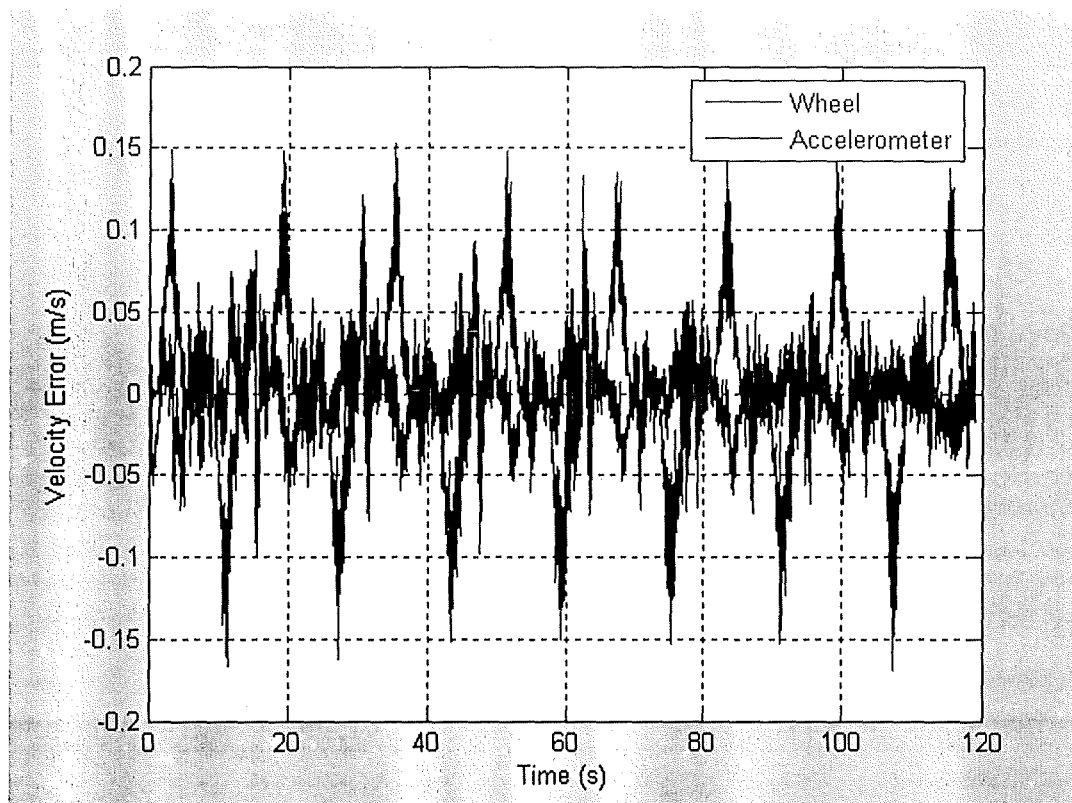
○ แกน Y



รูปที่ 38 เปรียบเทียบการวัดความเร็วด้วย Vision กับการวัดด้วย Accelerometer ในแนวแกน Y



รูปที่ 39 ภาพขยาย (บริเวณสี่เหลี่ยม)



รูปที่ 40 เปรียบเทียบความผิดพลาดจากการวัดด้วยล้อและ Accelerometer เมื่อเทียบกับ Vision

จากข้างต้น ทั้งในแนวแกน X และ Y จะพบว่าความเร็วที่วัดได้จากการใช้ Accelerometer สอดคล้องกับการวัดด้วย Vision ดังจะเห็นได้จากจุด 1 และ 2 ซึ่งแสดงให้เห็นว่า ในจังหวะที่หุ่นเคลื่อนที่ไปด้านใดด้านหนึ่งนั้น หุ่นได้มีการไหลตัวไปในแนวตั้งฉากกับการเคลื่อนที่ด้วย ซึ่งรูปแบบความเร็วที่พบนี้ ไม่สามารถวัดได้จากการวัดจากล้อแต่เพียงอย่างเดียว ดังจะเห็นได้จากรูปที่แสดง

จากรูป ความผิดพลาดในการวัดจากการใช้ Accelerometer ในแนวแกน X จะมีความผิดพลาดอยู่ในช่วง ± 0.05 m/s โดยมี Mean Square Error = 4.64 ในขณะที่ล้อมีความคลาดเคลื่อนอยู่ที่ ± 0.2 m/s โดยมี Mean Square Error = 23.82

เช่นเดียวกับในแนวแกน Y ความผิดพลาดในการวัดจากการใช้ Accelerometer จะมีความผิดพลาดอยู่ในช่วง ± 0.08 m/s โดยมี Mean Square Error = 3.60 ในขณะที่ล้อมีความคลาดเคลื่อนอยู่ที่ ± 0.15 m/s , Mean Square Error = 16.54

บทที่ 4

ข้อสรุปและข้อเสนอแนะ

4.1 ข้อสรุป

จากการศึกษาและทดลอง พบว่าการใช้การวัดข้อมูลจาก Gyroscope ผ่าน Kalman Filter จะมีความคลาดเคลื่อนจากการวัด อยู่ในช่วง ± 0.1 rad และมี Mean Square Error = 6.09 แต่การใช้ล้ออย่างเดียว ให้ความคลาดเคลื่อนเพิ่มขึ้นมากขึ้นตามเวลา, Mean Square Error = 101.56

ในแนวแกน X ความผิดพลาดในการวัดจากการใช้ Accelerometer จะมีความผิดพลาดอยู่ในช่วง ± 0.05 m/s โดยมี Mean Square Error = 4.64 ในขณะที่ล้อมีความคลาดเคลื่อนอยู่ที่ ± 0.2 m/s , Mean Square Error = 23.82

ในแนวแกน Y ความผิดพลาดในการวัดจากการใช้ Accelerometer จะมีความผิดพลาดอยู่ในช่วง ± 0.08 m/s โดยมี Mean Square Error = 3.60 ในขณะที่ล้อมีความคลาดเคลื่อนอยู่ที่ ± 0.15 m/s , Mean Square Error = 16.54

ดังนั้น จะเห็นได้ว่าการนำ Gyroscope และ Accelerometer นำมาใช้งานเป็น Inertial Measurement Unit นั้น สามารถที่จะให้การวัดความเร็วและมุมในการเคลื่อนที่ของหุ่นยนต์ที่มีความคลาดเคลื่อนน้อยกว่าการวัดโดยใช้ความเร็วจากล้อแต่เพียงอย่างเดียว

4.2 ข้อเสนอแนะ

ควรที่จะมีการทดลองเพิ่มเติม โดยทดลองที่ความเร็วและความเร่งของหุ่นยนต์ตลอดจนรูปแบบการวิ่งต่างๆ เพื่อที่จะได้สามารถวิเคราะห์ถึงประสิทธิภาพของวิธีการที่ได้พัฒนาขึ้นมาได้ครอบคลุมขึ้น

ในการวัดข้อมูลการเคลื่อนที่ของหุ่นยนต์นั้น ข้อมูลจากการมองเห็นน่าจะเป็นข้อมูลที่นำเชื่อถือมากกว่าการวัดบนล้อ ดังนั้น การพัฒนาเพิ่มเติมในส่วนของการส่งข้อมูลการมองเห็นกลับให้ตัวหุ่นยนต์ เพื่อป้อนให้กับ Kalman Filter อีกทางหนึ่งน่าจะช่วยลดความผิดพลาดได้มากขึ้น แต่อย่างไรก็ตามจำนวนข้อมูลที่จะสามารถส่งให้หุ่นในช่วงเวลาหนึ่งๆ มีจำกัด จึงต้องทำการศึกษาและหาวิธีเพิ่มขึ้นต่อไป

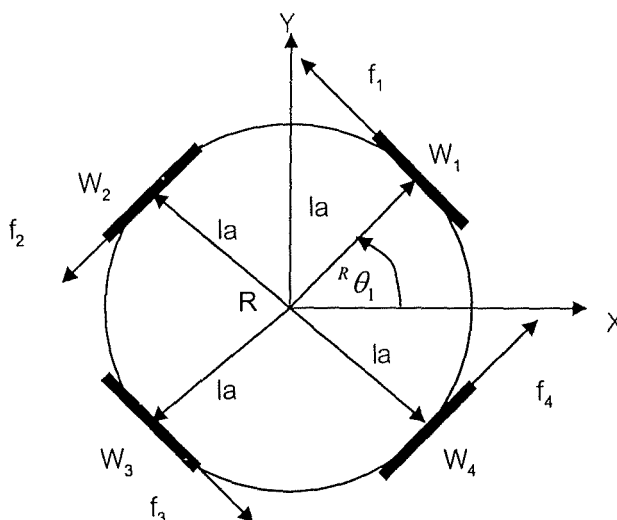
เอกสารอ้างอิง

- [1] มงคล เทียนวิบูลย์. การควบคุมหุ่นยนต์เคลื่อนที่ด้วยล้อที่สามารถเคลื่อนที่ได้อย่างอิสระ. วิทยานิพนธ์ปริญญาโทบริหารบัณฑิต ภาควิชาวิศวกรรมเครื่องกล คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย. 2542
- [2] Bonnie C. Baker. Anti-Aliasing, Analog Filter for Data Acquisition Systems. Microchip Technology Inc. Application Note. AN699.
- [3] Engineering Innovator Website. <http://www.eng.chula.ac.th/~eic>
- [4] Gary Welch, Gary Bishop. Introduction to Kalman filter. SIGGRAPH 2001 Course 8.
- [5] J. Borenstein ,H. R. Everett ,L. Feng. Where am I?, Sensors and Methods for Mobile Robot Positioning. University of Michigan. April 1996.
- [6] Official RoboCup Website. <http://www.robocup.org/>
- [7] Patrick F. Muir and Charles P. Neuman. Kinematic Modeling of Wheeled Mobile Robots. Ph.D Dissertation. Department of Electrical and Computer Engineering, The Robotics Institute, Carnegie Mellon University, 1988.

ภาคผนวก

การวิเคราะห์หาค่าสมการ Forward / Inverse Jacobian Matrix

จากการวางตัวของล้อ สามารถกำหนด Coordinate และ ทิศทางของเวกเตอร์ของความเร็วของแต่ละล้อได้ โดยอาศัยวิธีการของ Muir ดังนี้



กำหนดให้

$${}^R\theta_1 = 37^\circ, {}^R\theta_2 = 143^\circ, {}^R\theta_3 = 225^\circ, {}^R\theta_4 = 315^\circ$$

$$\text{Wheel Base Length } (la) = 0.0888 \text{ m}$$

$$\text{Wheel Radius } (R) = 0.027 \text{ m}$$

$$\text{Wheel Roller Radius } (r) = 0.012 \text{ m}$$

จากภาพจะได้เวกเตอร์แสดงทิศทางแต่ละล้อ $f_n = \begin{bmatrix} -\sin(\theta_n) \\ \cos(\theta_n) \\ 0 \end{bmatrix}$, $n = 1, 2, 3, 4$

ดังนั้น

$$f_1 = \begin{bmatrix} -\sin(37^\circ) \\ \cos(37^\circ) \\ 0 \end{bmatrix} = \begin{bmatrix} -0.601815 \\ 0.798636 \\ 0 \end{bmatrix}, f_2 = \begin{bmatrix} -\sin(143^\circ) \\ \cos(143^\circ) \\ 0 \end{bmatrix} = \begin{bmatrix} -0.601815 \\ -0.798636 \\ 0 \end{bmatrix}$$

$$f_3 = \begin{bmatrix} -\sin(225^\circ) \\ \cos(225^\circ) \\ 0 \end{bmatrix} = \begin{bmatrix} 0.707107 \\ -0.707107 \\ 0 \end{bmatrix}, f_4 = \begin{bmatrix} -\sin(315^\circ) \\ \cos(315^\circ) \\ 0 \end{bmatrix} = \begin{bmatrix} 0.707107 \\ 0.707107 \\ 0 \end{bmatrix}$$

จะได้ Jacobian ของแต่ละล้อ โดย $J_n = \begin{bmatrix} R \cdot f_n^x & r \cdot f_n^y & -la \cdot f_n^x \\ R \cdot f_n^y & -r \cdot f_n^x & -la \cdot f_n^y \\ 0 & 0 & 1 \end{bmatrix}$ ซึ่งสามารถแบ่งได้เป็น

Jacobian ของล้อในแกนที่มี Actuator หรือ Sensor อยู่ ($f_{n,s}$) นั่นคือในแนวแกนของมอเตอร์นั่นเอง และ ในแกนที่ไม่มี Actuator หรือ Sensor อยู่ ($f_{n,n}$)

แทนค่าคงที่ลงไปจะได้

$$\begin{aligned}
 J_1 &= \begin{bmatrix} -0.0162 & 0.7986 & 0.0535 \\ 0.0216 & 0.6018 & -0.0709 \\ 0 & 0 & 1 \end{bmatrix}, & J_{1,s} &= \begin{bmatrix} -0.0162 \\ 0.0216 \\ 0 \end{bmatrix}, & J_{1,n} &= \begin{bmatrix} 0.7986 & 0.0535 \\ 0.6018 & -0.0709 \\ 0 & 1 \end{bmatrix} \\
 J_2 &= \begin{bmatrix} -0.0162 & -0.7986 & 0.0535 \\ -0.0216 & 0.6018 & 0.0709 \\ 0 & 0 & 1 \end{bmatrix}, & J_{2,s} &= \begin{bmatrix} -0.0162 \\ -0.0216 \\ 0 \end{bmatrix}, & J_{2,n} &= \begin{bmatrix} -0.7986 & 0.0535 \\ 0.6018 & 0.0709 \\ 0 & 1 \end{bmatrix} \\
 J_3 &= \begin{bmatrix} 0.0191 & -0.7071 & -0.0628 \\ -0.0191 & -0.7071 & 0.0628 \\ 0 & 0 & 1 \end{bmatrix}, & J_{3,s} &= \begin{bmatrix} 0.0191 \\ -0.0191 \\ 0 \end{bmatrix}, & J_{3,n} &= \begin{bmatrix} -0.7071 & -0.0628 \\ -0.7071 & 0.0628 \\ 0 & 1 \end{bmatrix} \\
 J_4 &= \begin{bmatrix} 0.0191 & 0.7071 & -0.0628 \\ 0.0191 & -0.7071 & -0.0628 \\ 0 & 0 & 1 \end{bmatrix}, & J_{4,s} &= \begin{bmatrix} 0.0191 \\ 0.0191 \\ 0 \end{bmatrix}, & J_{4,n} &= \begin{bmatrix} 0.7071 & -0.0628 \\ -0.7071 & -0.0628 \\ 0 & 1 \end{bmatrix}
 \end{aligned}$$

ทำการกำหนด Delta matrix function $\Delta(U)$ โดยกำหนดให้

$$\Delta(U) = \begin{cases} -I & \text{for } U = \text{null} \\ U(U^T U)^{-1} U^T - I & \text{Otherwise} \end{cases}$$

ดังนั้นสามารถหา Delta matrix function $\Delta(J_{n,s})$ และ $\Delta(J_{n,n})$ ได้

$$\begin{aligned}
 \Delta(J_{1,s}) &= \begin{bmatrix} -0.6378 & -0.4806 & 0 \\ -0.4806 & -0.3622 & 0 \\ 0 & 0 & -1 \end{bmatrix}, & \Delta(J_{1,n}) &= \begin{bmatrix} -0.3593 & 0.4769 & 0.0530 \\ 0.4769 & -0.6328 & -0.0704 \\ 0.0530 & -0.0704 & -0.0078 \end{bmatrix} \\
 \Delta(J_{2,s}) &= \begin{bmatrix} -0.6378 & 0.4806 & 0 \\ 0.4806 & -0.3622 & 0 \\ 0 & 0 & -1 \end{bmatrix}, & \Delta(J_{2,n}) &= \begin{bmatrix} -0.3593 & -0.4769 & 0.0530 \\ -0.4769 & -0.6328 & 0.0704 \\ 0.0530 & 0.0704 & -0.0078 \end{bmatrix} \\
 \Delta(J_{3,s}) &= \begin{bmatrix} -0.5000 & -0.5000 & 0 \\ -0.5000 & -0.5000 & 0 \\ 0 & 0 & -1 \end{bmatrix}, & \Delta(J_{3,n}) &= \begin{bmatrix} -0.4961 & 0.4961 & -0.0623 \\ 0.4961 & -0.4961 & 0.0623 \\ -0.0623 & 0.0623 & -0.0078 \end{bmatrix} \\
 \Delta(J_{4,s}) &= \begin{bmatrix} -0.5000 & 0.5000 & 0 \\ 0.5000 & -0.5000 & 0 \\ 0 & 0 & -1 \end{bmatrix}, & \Delta(J_{4,n}) &= \begin{bmatrix} -0.4961 & -0.4961 & -0.0623 \\ -0.4961 & -0.4961 & -0.0623 \\ -0.0623 & -0.0623 & -0.0078 \end{bmatrix}
 \end{aligned}$$

ดังนั้นจึงสามารถหา Inverse Jacobian หรือเรียกว่า Actuated Inverse Solution ได้โดย
จากความสัมพันธ์

$$\dot{q} = J_a \dot{p} \quad \text{โดย } \dot{q} = \begin{bmatrix} \omega_{w_1,x} \\ \omega_{w_2,x} \\ \omega_{w_3,x} \\ \omega_{w_4,x} \end{bmatrix} \quad \text{และ} \quad \dot{p} = \begin{bmatrix} \bar{B} v_{Bx} \\ \bar{B} v_{By} \\ \bar{B} v_{Bz} \end{bmatrix}$$

$$\dot{q} = \begin{bmatrix} (J_{1,s}^T \Delta(J_{1,n}) J_{1,s})^{-1} J_{1,s}^T \Delta(J_{1,n}) \\ (J_{2,s}^T \Delta(J_{2,n}) J_{2,s})^{-1} J_{2,s}^T \Delta(J_{2,n}) \\ (J_{3,s}^T \Delta(J_{3,n}) J_{3,s})^{-1} J_{3,s}^T \Delta(J_{3,n}) \\ (J_{4,s}^T \Delta(J_{4,n}) J_{4,s})^{-1} J_{4,s}^T \Delta(J_{4,n}) \end{bmatrix} \dot{p}$$

จากข้างต้น ทำให้ได้ Inverse Jacobian Matrix ดังนี้

$$J_a = \begin{bmatrix} -22.2894 & 29.5791 & 3.2895 \\ -22.2894 & -29.5791 & 3.2895 \\ 26.1891 & -26.1891 & 3.2895 \\ 26.1891 & 26.1891 & 3.2895 \end{bmatrix}$$

ซึ่งจะพบว่ายู่ในรูป

$$\begin{bmatrix} \omega_{w_1,x} \\ \omega_{w_2,x} \\ \omega_{w_3,x} \\ \omega_{w_4,x} \end{bmatrix} = \frac{1}{R} \begin{bmatrix} f_{x_1} & f_{y_1} & l_a \\ f_{x_2} & f_{y_2} & l_a \\ f_{x_3} & f_{y_3} & l_a \\ f_{x_4} & f_{y_4} & l_a \end{bmatrix} \begin{bmatrix} \bar{B} v_{Bx} \\ \bar{B} v_{By} \\ \bar{B} v_{Bz} \end{bmatrix}$$

และจะสามารถหา Forward Jacobian หรือ Sensed Forward Solution ได้จาก

$$\dot{p} = J_s \dot{q} \quad \text{โดย } \dot{q} = \begin{bmatrix} \omega_{w_1,x} \\ \omega_{w_2,x} \\ \omega_{w_3,x} \\ \omega_{w_4,x} \end{bmatrix} \quad \text{และ} \quad \dot{p} = \begin{bmatrix} \bar{B} v_{Bx} \\ \bar{B} v_{By} \\ \bar{B} v_{Bz} \end{bmatrix}$$

ซึ่ง

$$J_s = [\Delta(J_{1,n}) + \Delta(J_{2,n}) + \Delta(J_{3,n}) + \Delta(J_{4,n})]^{-1} \cdot [\Delta(J_{1,n}) J_{1,s} \quad \Delta(J_{2,n}) J_{2,s} \quad \Delta(J_{3,n}) J_{3,s} \quad \Delta(J_{4,n}) J_{4,s}]$$

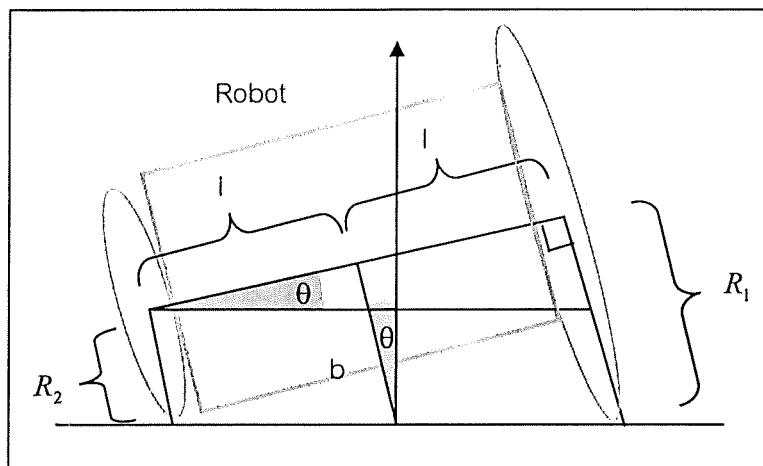
ดังนั้น จึงได้ Forward Jacobian Matrix ดังนี้

$$J_s = \begin{bmatrix} -0.0103138 & -0.0103138 & 0.0103138 & 0.0103138 \\ 0.0094756 & -0.0094756 & -0.0083896 & 0.0083896 \\ 0.0821134 & 0.0821134 & 0.0698862 & 0.0698862 \end{bmatrix}$$

การวิเคราะห์ผลของการเอียงตัวของหุ่นยนต์เนื่องมาจากล้อต่อค่าที่วัดได้

เนื่องจากหุ่นยนต์ที่ทำการทดลองใช้ล้อแบบ Omni-Directional ทำให้เกิดการสั่นสะเทือน ในขณะที่วิ่งของหุ่นยนต์เนื่องจากรัศมีของล้อไม่คงที่ จึงอาจส่งผลให้เซ็นเซอร์ที่ติดอยู่บนตัวหุ่นเกิดการเอียงตัวขึ้นมา ทำให้ค่าที่วัดได้คลาดเคลื่อนได้

ในที่นี้จะทำการวิเคราะห์ในรูปของเซ็นเซอร์ที่ติดอยู่กึ่งกลางระหว่างล้อสองล้อ เป็นระยะ l โดยล้อทั้งสองมีรัศมีเป็น R_1, R_2 ตามลำดับ ดังรูป



จากภาพข้างต้น เราจะสามารถหามุม θ และระยะ b ได้เท่ากับ

$$\theta = \tan^{-1}\left(\frac{R_1 - R_2}{2 \cdot l}\right)$$

$$b = \frac{R_1 + R_2}{2}$$

จากการวัดจะพบว่าล้อแต่ละอันนั้น มีรัศมีล้ออยู่ในช่วง 26.0 - 27.0 mm ดังนั้นเมื่อลองพิจารณาในกรณีที่ล้อทั้งสองมีรัศมีเป็น 27.0 และ 26.0 ตามลำดับ และมีระยะ $l = 80\text{mm}$ ดังนั้น

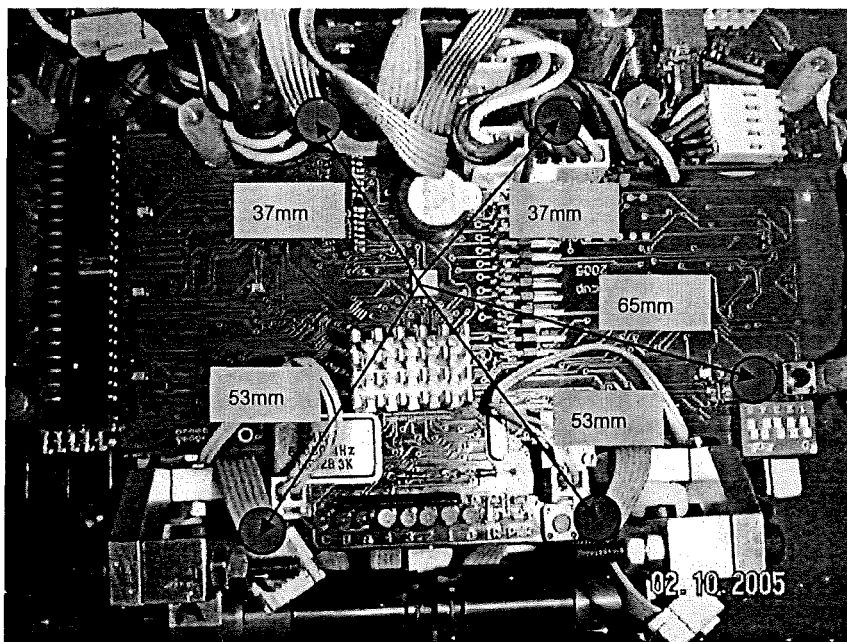
$$\theta = \tan^{-1}\left(\frac{27 - 26}{2 \cdot 80}\right)$$

$$\theta = 0.3581^\circ$$

ดังนั้นจะพบว่าค่าที่วัดได้จากเซ็นเซอร์จะวัดได้เท่ากับ $\cos(0.3581^\circ) = 0.99998$ หรือเท่ากับ 99.998% ของที่เกิดขึ้นจริงซึ่งถือว่ามีความน้อยมากเมื่อเทียบกับสัญญาณรบกวนที่เกิดขึ้น

และในความเป็นจริงหุ่นยนต์นี้ถูกออกแบบให้วิ่งอยู่บนสนามที่มีพื้นผิวเป็นพรม ทำให้ล้อสามารถยุบตัวลงในพรมได้ ดังนั้น การสั่นสะเทือน ตลอดจนการเอียงที่เกิดขึ้นจึงลดน้อยลงตามลำดับ

และเมื่อพิจารณาถึงการบิดตัวของแผ่นวงจรพิมพ์เมื่อติดตั้งลงบนตัวหุ่นยนต์โดยพิจารณาถึงจุดยึดแผ่นวงจรพิมพ์ ดังภาพ



ดังนั้นจะพบว่าจุดยึดต่างๆ อยู่ห่างจากตัวเซ็นเซอร์ไม่มาก การโก่งตัวของแผ่นวงจรพิมพ์จึงไม่น่าที่จะมีผลมากนักต่อการวัดของเซ็นเซอร์